1. Basic Results — Inverse Transform Method

We want to use $\mathcal{U}(0,1)$ numbers to generate observations (variates) from other distributions.

Let X be a random variable with c.d.f. $F(\cdot)$. Then

$$U = F(X) \sim \mathcal{U}(0,1).$$

Proof: Let $Y = F(X)$ and suppose that Y has c.d.f. $G(y)$. Then (for the continuous case),

$$\begin{aligned} G(y) &= \mathsf{P}(Y \le y) = \mathsf{P}(F(X) \le y) \\ &= \mathsf{P}(X \le F^{-1}(y)) = F(F^{-1}(y)) \\ &= y. \quad \diamondsuit \end{aligned}$$

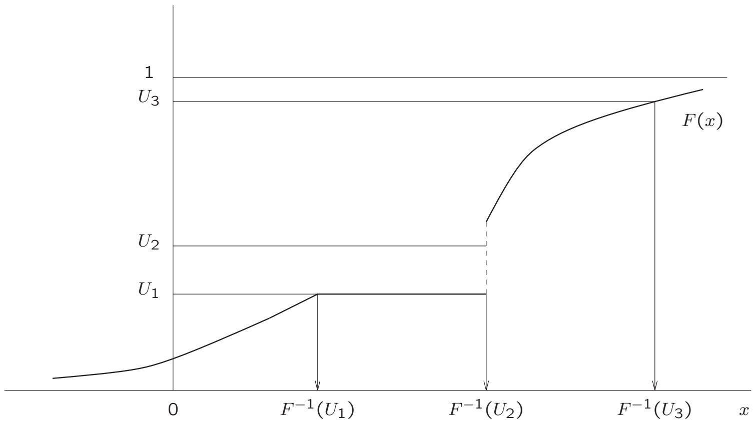
In the above, we defined the inverse c.d.f. by

$$F^{-1}(u) = \inf[x : F(x) \ge u] \quad u \in [0,1].$$

Let $U \sim \mathcal{U}(0,1)$. Then the random variable $F^{-1}(U)$ has the same distribution as X.

1. Sample U from $\mathcal{U}(0,1)$.

2. Return $X = F^{-1}(U)$.



2. Acceptance-Rejection Method

Example 14 (Baby example, which you would usually do via inverse transform, but what the heck!)

Generate a U(2/3,1) RV. Here's the A-R algorithm:

1. Generate $U \sim \mathrm{U}(0,1)$.

2. If $U \geq 2/3$, ACCEPT $X \leftarrow U$. Otherwise, REJECT and go to 1.

Motivation: The majority of c.d.f.'s cannot be inverted efficiently.

Suppose we want to simulate a continuous RV with p.d.f. $f(x)$, but that it's difficult to generate directly. Also suppose that we can easily generate a RV having p.d.f. $h(x) \equiv t(x)/c$, where $t(x)$ *majorizes* $f(x)$, i.e.,

$$t(x) \geq f(x), \quad x \in \mathbb{R},$$

and

$$c \equiv \int_{-\infty}^{\infty} t(x)\,dx \geq \int_{-\infty}^{\infty} f(x)\,dx = 1,$$

where we assume that $c < \infty$.

Then f can be written as

$$f(x) = c \times \frac{f(x)}{t(x)} \times \frac{t(x)}{c} = cg(x)h(x),$$

where

$$\int_{-\infty}^{\infty} h(x)\,dx = 1 \quad (h \text{ is a density})$$

and

$$0 \leq g(x) \leq 1.$$

Theorem 15 (von Neumann, 1951) *Let $U \sim \mathcal{U}(0,1)$, and let Y a random variable with density h. If $U \le g(Y)$, then Y has (conditional) density f.*

This suggests the following "acceptance-rejection" algorithm ...

Algorithm A-R

Repeat

Generate U from $\mathcal{U}(0,1)$

Generate Y from h

until $U \le g(Y)$

Return $X \leftarrow Y$

There are two main issues:

- The ability to quickly sample from h.

- c must be small (t must be "close" to f) since

$$\Pr[U \le g(Y)] = \frac{1}{c}$$

and the mean number of trials until "success" $[U \le g(Y)]$ is equal to c.

Example 16 (Law & Kelton) Generate a RV with p.d.f. $f(x) = 60x^3(1-x)^2$, $0 \le x \le 1$. Can't invert this analytically.

Note that the maximum occurs at $x = 0.6$, and $f(0.6) = 2.0736$.

Using the majorizing function

$$t(x) = 2.0736, \quad 0 \le x \le 1$$

(which isn't actually too efficient), we get $c = \int_0^1 f(x)\,dx = 2.0736$, and therefore

$$h(x) = 1, \quad 0 \le x \le 1 \quad \text{(i.e., a U(0,1) p.d.f.)}$$

and

$$g(x) = 60x^3(1-x)^2/2.0736.$$

E.g., if we generate $U = 0.13$ and $Y = 0.25$, then it turns out that $U \le g(Y) = \frac{60Y^3(1-Y)^2}{2.0736}$, so we take $X \leftarrow 0.25$.

Example 17 (Ross) The standard half-normal distribution with density

$$f(x) = \frac{2}{\sqrt{2\pi}} e^{-x^2/2}, \quad x \ge 0.$$

Using the majorizing function

$$t(x) = \sqrt{\frac{2e}{\pi}} e^{-x}$$

we get

$$c = \sqrt{\frac{2e}{\pi}} \int_0^\infty e^{-x}\,dx = \sqrt{\frac{2e}{\pi}} = 1.3155,$$

$$h(x) = e^{-x} \quad \text{[exponential}(\lambda = 1)\text{ density]},$$

and

$$g(x) = e^{-(x-1)^2/2}.$$

How can we generate from $N(0,1)$?

Generate U from $\mathcal{U}(0,1)$

Generate X from the half-normal distribution

Return

$$Z = \begin{cases} -X & \text{if } U \leq 1/2 \\ X & \text{if } U > 1/2. \end{cases}$$

How can we generate from $N(\mu, \sigma^2)$?

Use the transformation $\mu + \sigma Z$.

Example 18 The gamma distribution with density

$$f(x) = \frac{(x/\alpha)^{\beta-1}}{\alpha\Gamma(\beta)} e^{-(x/\alpha)^{\beta}}, \quad x > 0.$$

If the shape parameter $\beta < 1$, we use the following A-R algorithm with $c \leq 1.39$:

Algorithm GAM1

$b \leftarrow (e+\beta)/e$ (e is the base of the natural logarithm)

While (True)

Generate U from $\mathcal{U}(0,1)$; $W \leftarrow bU$

If $W < 1$

$Y \leftarrow W^{1/\beta}$; Generate V from $\mathcal{U}(0,1)$

If $V \le e^{-Y}$: Return $X = \alpha Y$

Else

$Y \leftarrow -\ln[(b-W)/\beta]$

Generate V from $\mathcal{U}(0,1)$

If $V \le Y^{\beta-1}$: Return $X = \alpha Y$

If $\beta \ge 1$, the value of c for the following A-R algorithm decreases from $4/e = 1.47$ to $\sqrt{4/\pi} = 1.13$ as β increases from 1 to ∞.

Algorithm GAM2

$a \leftarrow (2\beta-1)^{-1/2}$; $b \leftarrow \beta - \ln 4$; $c \leftarrow \beta + a^{-1}$; $d \leftarrow 1 + \ln 4.5$

While (True)

Generate U_1, U_2 from $\mathcal{U}(0,1)$

$V \leftarrow a \ln[U_1/(1-U_1)]$

$Y \leftarrow \beta e^V$; $Z \leftarrow U_1^2 U_2$

$W \leftarrow b + cV - Y$

If $W + d - 4.5Z \ge 0$: Return $X = \alpha Y$

Else

If $W \ge \ln Z$: Return $X = \alpha Y$

Example 19 The Poisson distribution with probability function

$$\Pr(X = n) = e^{-\lambda}\frac{\lambda^n}{n!}, \quad n = 0, 1, \ldots$$

Define A_i as the ith interarrival time from a Pois(λ) process. Then

$$\begin{aligned}
& X = n \\
& \quad \Leftrightarrow \text{ See exactly } n \text{ PP}(\lambda) \text{ arrivals by } t = 1 \\
& \quad \Leftrightarrow \sum_{i=1}^{n} A_i \le 1 < \sum_{i=1}^{n+1} A_i \\
& \quad \Leftrightarrow \sum_{i=1}^{n} \left[\frac{-1}{\lambda}\ell\text{n}(U_i)\right] \le 1 < \sum_{i=1}^{n+1} \left[\frac{-1}{\lambda}\ell\text{n}(U_i)\right] \\
& \quad \Leftrightarrow \frac{-1}{\lambda}\ell\text{n}\left(\prod_{i=1}^{n} U_i\right) \le 1 < \frac{-1}{\lambda}\ell\text{n}\left(\prod_{i=1}^{n+1} U_i\right) \\
& \quad \Leftrightarrow \prod_{i=1}^{n} U_i \ge e^{-\lambda} > \prod_{i=1}^{n+1} U_i. \qquad (5)
\end{aligned}$$

The following A-R algorithm samples U(0,1)'s until (5) becomes true.

Algorithm POIS1

$a \leftarrow e^{-\lambda}$; $p \leftarrow 1$; $X \leftarrow -1$

Until $p \le a$

  Generate U from $\mathcal{U}(0, 1)$

  $p \leftarrow pU$; $X \leftarrow X + 1$

Return X

Example 20 Apply Algorithm POIS1 to obtain a Pois($\lambda = 2$) variate.

Sample until $e^{-\lambda} = 0.1353 > \prod_{i=1}^{n+1} U_i$.

n	U_{n+1}	$\prod_{i=1}^{n+1} U_i$	Stop?
0	0.3911	0.3911	No
1	0.9451	0.3696	No
2	0.5033	0.1860	No
3	0.7003	0.1303	Yes

Thus, we take $X = 3$.

Remark 21 How many U's are required to generate one realization of X? Easy argument says that the expected number you'll need is $\mathsf{E}[X+1] = \lambda + 1$.

Algorithm POIS2 (For $\lambda \geq 20$)

$a \leftarrow \pi\sqrt{\lambda/3}$; $b \leftarrow a/\lambda$; $c \leftarrow 0.767 - 3.36/\lambda$; $d \leftarrow \ln c - \ln b - \lambda$

Repeat

Repeat

Generate U from $\mathcal{U}(0,1)$

$Y \leftarrow [a - \ln((1-U)/U]/b$

until $Y > -1/2$

$X \leftarrow \lfloor Y + 1/2 \rfloor$

Generate V from $\mathcal{U}(0,1)$

until $a - bY + \ln[V/(1+e^{a-bY})^2] \leq d + X\ln\lambda - \ln(X!)$

Return X

Alternatively, we can use the normal approximation

$$\frac{X-\lambda}{\sqrt{\lambda}} \approx N(0,1).$$

Algorithm POIS3 (For $\lambda \geq 20$)

$\alpha \leftarrow \sqrt{\lambda}$

Generate Z from $N(0,1)$

Return $X = \max(0, \lfloor \lambda + aZ + 0.5 \rfloor)$ (Note that this employs a "continuity correction."

3 Generating Poisson Arrivals

When the arrival rate is constant, say λ, the interarrival times are i.i.d. exponential(λ) and the arrival times are generated recursively:

$$\begin{aligned} T_0 &= 0 \\ T_i &= T_{i-1} - \frac{1}{\lambda} \ln U_i, \quad i \geq 1 \end{aligned}$$

How can we generate a fixed number n of arrivals in a time interval $[a,b]$?

Generate $U_1, \ldots, U_n$ from $\mathcal{U}(0,1)$

Sort the U_i's: $U_{(1)} < U_{(2)} < \cdots < U_{(n)}$

Set the arrival times to $T_i = a + (b-a)U_{(i)}$

4. Special-Case Techniques

4.1. Box-Müller Method

Nice way to generate standard normals.

Theorem 23 *If U_1, U_2 are i.i.d. U(0,1), then*

$$\begin{aligned} Z_1 &= \sqrt{-2\ell\text{n}(U_1)}\cos(2\pi U_2) \\ Z_2 &= \sqrt{-2\ell\text{n}(U_1)}\sin(2\pi U_2) \end{aligned}$$

are i.i.d. Nor(0,1).

Note that the trig calculations must be done in radians.

Proof Someday soon. $\diamondsuit$

Some cool corollaries from Box-Müller.

Example 24 Note that

$$Z_1^2 + Z_2^2 \sim \chi^2(1) + \chi^2(1) \sim \chi^2(2).$$

But

$$\begin{aligned} &Z_1^2 + Z_2^2 \\ &= -2\ell\text{n}(U_1)(\cos^2(2\pi U_2) + \sin^2(2\pi U_2)) \\ &= -2\ell\text{n}(U_1) \\ &\sim \text{Exp}(1/2). \end{aligned}$$

Thus, we've just proven that

$$\chi^2(1) + \chi^2(1) \sim \text{Exp}(1/2).$$

Example 25 Note that

$$Z_1/Z_2 \sim \text{Nor}(0,1)/\text{Nor}(0,1) \sim \text{Cauchy}.$$

But

$$\begin{aligned} Z_1/Z_2 &= \frac{\sqrt{-2\ell\text{n}(U_1)}\sin(2\pi U_2)}{\sqrt{-2\ell\text{n}(U_1)}\cos(2\pi U_2)} \\ &= \tan(2\pi U_2). \end{aligned}$$

Thus, we've just proven that

$$\tan(2\pi U) \sim \text{Cauchy}.$$

Similarly,

$$\cot(2\pi U) \sim \text{Cauchy}.$$

Similarly,

$$Z_1^2/Z_2^2 = \tan^2(2\pi U) \sim F(1,1).$$

(Did you know that?)

Polar Method — a little faster than Box-Müller:

1. Generate U_1, U_2 i.i.d. U(0,1).

Let $V_i = 2U_i - 1$, $i = 1,2$, and $W = V_1^2 + V_2^2$.

2. If $W > 1$, reject and go back to step 1.

Otherwise, let $Y = \sqrt{-2(\ell\text{n}W)/W}$, and accept $Z_i \leftarrow V_iY$, $i = 1,2$.