HW5 - ISYE 6644

Marcos Grillo

April 2023

1

1.1

Matlab utilizes an optimized version of the Mersenne Twister as its default. Professor Alexopoulos is very upset with this fact. The keyword for this generator is "twister". The mrg32k3a generator has a period of roughly 2^{191}.

1.2

I found a repository for both C and Java code: https://github.com/vigna/MRG32k3a

1.3

Python currently uses the deterministic Mersenne Twister as its default generator.

1.4

R also uses the Mersenne Twister by default, however, many different methods are available in the project. A user can even supply their own generator. More details can be found here: https://stat.ethz.ch/R-manual/R-devel/library/base/html/Random.html

1.5

I visited the site I promise. Excel uses the Mersenne Twister as of 2010.

2

This question is answered in its entirety in the attached ipynb.

3

I show that the beta and gamma parameters are the respective scale and location parameters in the attached ipynb. The formula for generating realizations for this distribution is as follows:

We begin with the CDF, setting it equal to a variable u:

$$u = \frac{1}{1 + exp(-(x-\gamma)/\beta)}$$

$$\frac{1}{u} = 1 + exp(-(x-\gamma)/\beta)$$

$$\frac{1}{u} - 1 = exp(-(x-\gamma)/\beta)$$

$$ln(\frac{1}{u} - 1) = -(x-\gamma)/\beta$$

$$\beta(ln(\frac{1}{u} - 1)) = -x + \gamma$$

$$\beta(ln(\frac{1}{u} - 1)) + \gamma = x$$

Thus, generating numbers from the U (0,1) distribution, we can generate realizations from this distribution.

4

4.1

$$X = min(U_1, U_2)$$

$$F(x) = P(X \leq x)$$

$$= P(min(U_1, U_2) \leq x)$$

$$= 1 - P(min(U_1, U_2) \geq x)$$

$$= 1 - P(U_1 \geq x, U_2 \geq x)$$

$$= 1 - P(U_1 \geq x) * P(U_2 \geq x)$$

$$= 1 - (1 - P(U_1 \leq x)) * (1 - P(U_2 \leq x))$$

$$= 1 - (1 - P(U_1 \leq x))^2$$

Applying inverse transform:

$$U_3 = 1 - (1 - P(U_1 \leq x))^2$$

$$U_3 - 1 = (1 - P(U_1 \leq x))^2$$

$$\sqrt{U_3 - 1} = 1 - P(U_1 \leq x)$$

$$\sqrt{U_3 - 1} = 1 - \frac{x-a}{b-a}$$

$$\sqrt{U_3 - 1} - 1 = -\frac{x-a}{b-a}$$

$$(b-a)(\sqrt{U_3 - 1} - 1) = -x + a$$

$$(b-a)(\sqrt{U_3 - 1} - 1) + a = x$$

Where a and b are the lower and upper bounds for the uniform distribution, respectively. In this case, 0 and 1, therefore:

$$\sqrt{U_3 - 1} - 1 = x$$

4.2

$$X = max(U_1, U_2)$$

$$F(x) = P(X \geq x)$$

$$= P(max(U_1, U_2) \geq x)$$

$$= 1 - P(max(U_1, U_2) \leq x)$$

$$= 1 - P(U_1 \leq x, U_2 \leq x)$$

$$= 1 - P(U_1 \leq x) * P(U_2 \leq x)$$

$$= 1 - (P(U_1 \leq x))^2$$

Applying inverse transform:

$$U_3 = (1 - P(U_1 \leq x))^2$$

$$\sqrt{U_3 - 1} = P(U_1 \leq x)$$

$$\sqrt{U_3 - 1} = P(U_1 \leq x)$$

$$\sqrt{U_3 - 1} = \frac{x-a}{b-a}$$

$$(b-a)(\sqrt{U_3 - 1}) = x - a$$

$$(b-a)(\sqrt{U_3 - 1}) + a = x$$

Where a and b are the lower and upper bounds for the uniform distribution, respectively. In this case, 0 and 1, therefore:

$$\sqrt{U_3 - 1} = x$$

4.3

$$X = max(X_1, X_2)$$
$$F(x) = P(X \geq x)$$
$$= P(max(X_1, X_2) \geq x)$$
$$= 1 - P(max(X_1, X_2) \leq x)$$
$$= 1 - P(X_1 \leq x, X_2 \leq x)$$
$$= 1 - P(X_1 \leq x) * P(X_2 \leq x)$$
$$= 1 - (P(X_1 \leq x))^2$$

Applying inverse-transform:

$$U_3 = (1 - P(U_1 \leq x))^2$$
$$\sqrt{U_3 - 1} = P(U_1 \leq x)$$
$$\sqrt{U_3 - 1} = P(U_1 \leq x)$$
$$\sqrt{U_3 - 1} = 1 - exp(-\lambda x)$$
$$\sqrt{U_3 - 1} - 1 = -exp(-\lambda x)$$
$$-\sqrt{U_3 - 1} + 1 = exp(-\lambda x)$$
$$ln(-\sqrt{U_3 - 1} + 1) = -\lambda x$$
$$-\frac{ln(-\sqrt{U_3 - 1} + 1)}{\lambda} = x$$

5

5.1 Inverse-transform:

Inverse-transform:

$$F(x) = \int_{-1}^{x} f(x)$$
$$F(x) = \int_{-1}^{x} \frac{3x^2}{2} dx$$
$$F(x) = \frac{x^3}{2}|_{-1}^{x}$$
$$F(x) = \frac{1}{2}(x^3 + 1)$$

Setting equal to U and solving for x:

$$U = \frac{1}{2}(x^3 + 1)$$
$$2U = x^3 + 1$$
$$\sqrt[3]{2U - 1} = x$$

5.2 Composition method:

For $-1 \leq x \leq 0, f_1(x) = 3x^2$.

For $0 \leq x \leq 1, f_2(x) = 3x^2$.

$f(x)$ now becomes:

$$f(x) = 0.5(f_1(x)) + 0.5(f_2(x))$$

$$f(x) = 0.5(3x^2) + 0.5(3x^2)$$

Where:
$f_1(x) = 0$ when $x < -1$, $3x^2$ when $-1 \leq x \leq 0$ and 0 when $x > 0$.
$f_2(x) = 0$ when $x < 0$, $3x^2$ when $0 \leq x \leq 1$ and 0 when $x > 1$.

When we integrate these functions to get the cdf they become:

$$F_1(x) = x^3$$

$$F_2(x) = x^3$$

Applying inverse-transform method:
For U_1, corresponding to $F_1(x)$:

$$U_1 = x^3$$

$$\sqrt[3]{U_1} = x$$

For U_2, corresponding to $F_2(x)$:

$$U_2 = x^3$$

$$\sqrt[3]{U_2} = x$$

Finally, we have the formula for generations using the composition method:

$X = \sqrt[3]{U}$ for $0 \leq U < 0.5$ and $X = \sqrt[3]{U}$ for $0.5 \leq U \leq 1$

6

Beginning with $exp(2x)$, when U is less than 0.5:

$$F(x) = \int exp(2x)dx$$

$$F(x) = \frac{exp(2x)}{2}$$

Applying inverse transform:

$$U = \frac{exp(2x)}{2}$$

$$2U = exp(2x)$$

$$ln(2U) = 2x$$

$$\frac{ln(2U)}{2} = x$$

Continuing with $exp(2x)$, in cases where U is not less than 0.5:

$$F(x) = \int exp(2x)dx$$

$$F(x) = \frac{exp(2x)}{2}$$

Applying inverse transform:

$$U = \frac{exp(2x)}{2}$$

$$2U = exp(2x)$$

$$ln(2U) = 2x$$

$$\frac{ln(2U)}{2} = x$$

7

7.1

Deriving the function:

$$\frac{d}{dx}(20x(1-x)^3) = -20(1-x)^2(-1+4x)$$

Setting to 0 and solving for x, we get 1 and 0.25 as candidates. Plugging these back into f(x):

$$f(0.25) = 20(0.25)(1-0.25)^3 = 2.109$$

$$f(1) = 20(0^3) = 0$$

Therefore, the value that maximizes this function is 0.25.

7.2

$$t(x) = 2.109$$

$$c = \int_0^1 2.109dx = 2.109$$

$$h(x) = \frac{t(x)}{c} = 1$$

$$g(x) = \frac{f(x)}{t(x)} = \frac{20x(1-x)^3}{2.109}$$

Thus, the algorithm becomes, in pseudocode:

Repeat:
  Generate U from U(0,1)
  Generate Y from h(x)
Until $U \leq g(Y)$
Return X $\leftarrow$ Y

7.3

The expected number of iterations until a realization from $f(x)$ is delivered is equal to c, so 2.109

8

$$p = \frac{1}{20}$$
$$pf(x) = x(1-x)^3$$
$$pf(v/u) = (v/u)(1-v/u)^3$$

Fetching upper and lower bounds for v and u:

$$u_* = 0$$
$$u^* = sup(\sqrt{(v/u)(1-v/u)^3}) = 0.325$$
$$v_* = inf(v/u\sqrt{(v/u)(1-v/u)^3}) = 0$$
$$v^* = sup(v/u\sqrt{(v/u)(1-v/u)^3}) = 0.125$$

The algorithm in this case is, in pseudocode:

Repeat:
  Generate U from U(0,u^*) and V from U(v_*, v^*)
  Set Z = V/U
Until $U^2 \leq pf(z)$
Return X = Z

The acceptance probability in this case is:

$$\frac{|S|}{|T|} = \frac{p/2}{u^*(v^*-v_*)} = \frac{1/40}{0.325(0.125)} = 0.615$$

The amount of Uniform(0,1) observations required to generate a single observation from $f(.)$ in this case is equal to:

$$2(\frac{1}{0.615}) = 3.252$$