Transformation Procedure for Generating Random Variates from the Exponential Distribution

Akram M. Al-Abood[1], Adil M. Ahmed[2], Akram H. Al-Shather[1]

[1] Department of Mathematics and Computer Applications, College of Science, Al-Nahrain University

[2] Department of Mathematics, Ibn Al-Haitham College of Education, Baghdad University

Abstract

In this paper we introduce a new technique for generating random variates from the exponential distribution. The procedure is based on a proposed mapping that transform an order sample from U(0,1) to the unit Exp(1). The method is developed theoretically and assessed practically by the basic Monte Carlo simulation. Comparison is made with that of inverse transform method.

Introduction

In recent years, several procedures are suggested, tested, and used for generating random variates from Exp(λ). The procedures of Naylor[1], Fochel[2], Ahrens and Dieter[3], and Marsaglia[4] for generating random variates from Exp(λ) are based on the composition method, Acceptance rejection method and Forsythe method[5]. These procedures without the benefit of a logarithmic Transformation Neumann's method[6] for generating random variates from Exp(1) based on sampling a sequence of i.i.d r.v's from the standard triangular distribution

$$f(x)=\begin{cases}\dfrac{2x}{\lambda} & 0<x<\lambda\\[2mm] \dfrac{2(x-1)}{1-\lambda} & \lambda<x\le 1\end{cases}$$

and it is shown that generation of one exponential variates in such way requires on the average a sequence of 6 random number.

Inverse transform method[7] is the one of most common use for generating random variates from Exp(λ) which can be described by the flowing algorithm. Note that an exponential variates X has p.d.f

$$f(x)=\frac{1}{\lambda}e^{-\frac{x}{\lambda}},0<x<\infty$$

$$=0 \qquad ,e.w,\lambda>0$$

By inverse transform method

$$U=F(X)=1-e^{-\frac{x}{\lambda}}$$

So that X=-λln(1-U).

Since 1-U is distributed in the same way as U, we have

X=-λln U

II. Algorithm

1. Read λ.
2. Generating U from U(0,1).
3. Set X=-λ ln U.
4. Deliver X as a random variates generated from Exp(λ).

We note that, if the random variates X is sampled from Exp(1) then the r.v Y=λX is from Exp(λ).

Although the algorithm seem very simple, but the computation of the natural algorithm on a digital computer consist of a power series expansion for the each uniform variates generated.

2.Proposition

Let $U_1,U_2,\ldots,U_n,U_{n+1},\ldots,U_{2n-1}$ be a random sample of size 2n-1 from U(0, 1) and let $W_1,W_2,\ldots,W_{n-1}$ be the order statistic corresponding to the r.v's $U_{n+1},U_{n+2},\ldots,U_{2n-1}$. Assume $W_0=0$ and $W_n=1$, then the r.v's

$$Y_k=-(W_k-W_{k-1})\ln\prod_{i=1}^{n}U_i,k=1,2,\ldots,n$$ represent a sample of size n from Exp(1)

Proof

Let $X_k=W_k-W_{k-1}$, $k=1,2,\ldots,n-1$ and let

$$X_n=-\ln\prod_{i=1}^{n}U_i=-\sum_{i=1}^{n}\ln U_i$$

The distn of X_n can be obtained by using m.g.f technique. Viz $M_{X_n}(t)$ be the m.g.f of X_n then

$$M_{X_n}(t)=E(e^{tX_n})=E(e^{-t\sum_{i=1}^{n}\ln U_i})$$

$$E(\prod_{i=1}^{n}e^{-t\ln U_i})=E(\prod_{i=1}^{n}e^{\ln U_i^{-t}})=E(\prod_{i=1}^{n}U_i^{-t})=\prod_{i=1}^{n}E(U_i^{-t})$$

because U_i, $i=1,\ldots,n$ are independent but for any U~U(0, 1),

$$E(U^{-t})=\int_0^1 u^{-t}du=\frac{1}{1-t}$$

Therefore $M_{X_n}(t)=\prod_{i=1}^{n}\frac{1}{1-t}=\frac{1}{(1-t)^n}$ which in the m.g.f of G(n,1), that is X_n~G(n,1) with p.d.f

$$f_n(x_n)=\frac{1}{\Gamma(n)}x_n^{n-1}e^{-x_n},0<x_n<\infty$$

$$=0,e.w$$

next, let us find the distn. Of r.v's $X_k = W_k - W_{k-1}$, $k=1,2,\ldots,n-1$.

Since $U_{n+1}, U_{n+2}, \ldots, U_{2n-1}$ are independent from U(0, 1). Then the joint p.d.f is $g(u_{n+1}, u_{n+2}, \ldots, u_{2n-1}) = 1$, $0 < u_i < 1$, $i=1,2,\ldots,n-1$

$= 0$, e.w.

from order statistics theory the joint p.d.f is of $W_1, W_2, \ldots, W_{n-1}$ is

$h(w_1, w_2, \ldots, w_{n-1}) = (n-1)!$ $\quad 0 < w_1 < w_2 < \ldots < w_{n-1} < 1$

$= 0$ $\quad$,e.w.

now the functions $x_1 = w_1$, $x_2 = w_2 - w_1, \ldots, x_{n-1} = w_{n-1} - w_{n-2}$, define one-to-one transformation that maps the space $A = \{(w_1, w_2, \ldots, w_{n-1}): 0 < w_1 < w_2 < \ldots < w_{n-1} < 1\}$ on the space $B = \{(x_1, x_2, \ldots, x_{n-1}): x_i > 0, i=1,2,\ldots,n-1$ inside $\sum_{k=1}^{n-1} x_k \le 1\}$ with inverse transform

$w_1 = x_1, w_2 = x_1 + x_2, \ldots, w_{n-1} = x_1 + x_2 + \ldots + x_{n-1}$

and the jacobian of the transformation

$$J = \frac{\partial(w_1, w_2, \ldots, w_{n-1})}{\partial(x_1, x_2, \ldots, x_{n-1})} = \begin{vmatrix} 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \ldots & \vdots \\ 0 & 0 & \cdots & 1 \end{vmatrix}_{(n-1)\times(n-1)} = 1$$

Then the joint p.d.f of $X_1, X_2, \ldots, X_{n-1}$ is

$$f^*(x_1, x_2, \ldots, x_{n-1}) = h\left(x_1, x_1 + x_2, \ldots, \sum_{i=1}^{n-1} x_i\right)|J|$$

$$= (n-1)!, x_k \ge 0, k=1,2,\ldots,n-1, \quad \sum_{i=1}^{n-1} x_i \le 1$$

Since the r.v's are independent of X_n, then the joint p.d.f of $X_1, X_2, \ldots, X_n$.

$$f(x_1, x_2, \ldots, x_{n-1}, x_n) = f^*(x_1, x_2, \ldots, x_{n-1}) f_n(x_n)$$

$$= (n-1)! \frac{1}{\Gamma(n)} x_n^{n-1} e^{-x_n}$$

$$= x_n^{n-1} e^{-x_n}, 0 < x_n < \infty$$

$$= 0, \quad e.w.$$

Finally, consider the transformation $Y_k = X_k X_n$, $k=1,2,\ldots,n-1$ and $Y_n = (1 - X_1 - X_2 - \ldots - X_{n-1}) X_n = x_n$

That is

$Y_1 = X_1 X_n$

$Y_2 = X_2 X_n$

.

.

.

$Y_{n-1} = X_{n-1} X_n$

$Y_n = (1 - X_1 - X_2 - \ldots - X_{n-1}) X_n$

and

$$\sum Y_i = (X_1 + X_2 + \ldots + X_n) X_n + (1 - X_1 - X_2 - \ldots - X_n) X_n = X_n$$

Now, the function $y_k = x_k x_n$, $k=1,2,\ldots,n-1$ and

$$y_n = \left(1 - \sum_{k=1}^{n-1} x_k\right) x_n$$

define one-to-one transformation that maps the space

$$A = \left\{(x_1, x_2, \ldots, x_{n-1}, x_n) : x_i > 0, i=1,2,\ldots,n-1, \sum_{i=1}^{n-1} x_i \le 1, 0 < x_n < \infty\right\}$$

onto the space

$$B^* = \{(y_1, y_2, \ldots, y_n) : 0 < y_i < \infty, i=1,2,\ldots,n\}$$

With inverse

$$X_k = \frac{y_k}{\sum_{i=1}^{n} y_i}, k=1,2,\ldots,n-1, x_n = \sum_{i=1}^{n} y_i$$

$$J^{-1} = \frac{\partial(y_1, y_2, \ldots, y_n)}{\partial(x_1, x_2, \ldots, x_n)} = \left(\sum_{i=1}^{n} y_i\right)^{n-1}$$

Then the joint p.d.f of $Y_1, Y_2, \ldots, Y_n$ is

$$g^*(y_1, y_2, \ldots, y_n) = f\left[\frac{y_1}{\sum_{i=1}^{n} y_i}, \frac{y_2}{\sum_{i=1}^{n} y_i}, \ldots, \frac{y_k}{\sum_{i=1}^{n} y_i}, \sum_{i=1}^{n} y_i\right]|J|$$

$$= \left(\sum_{i=1}^{n} y_i\right)^{n-1} e^{-\sum_{i=1}^{n} y_i} \left(\sum_{i=1}^{n} y_i\right)^{-(n-1)}$$

$$= e^{-\sum_{i=1}^{n} y_i} \qquad 0 \le y_i \le \infty, \quad i=1,2,\ldots,n$$

Which is the joint p.d.f of n independent r.v's from Exp(1).

Algorithm PT describes the necessary steps for generating r.v's from Exp(1) by the proposed procedure

PT-Algorithm:

1. Generate $U_1, U_2, \ldots, U_n, U_{n+1}, \ldots, U_{2n-1}$ from U(0, 1).
2. Arrange $U_{n+1}, \ldots, U_{2n-1}$ in ascending order of magnitudes by using the order statistics $W_1, W_2, \ldots, W_{n-1}$.
3. Set $W_0 = 0$ and $W_n = 1$.
4. Put $Y_k = -(W_k - W_{k-1}) \ln \prod_{i=1}^{n} U_i, k=1,2,\ldots n.$
5. Deliver Y_k, $k=1,2,\ldots,n$ as a r.v generated from Exp(1).

3. Conclusion:

Comparing PT-Algorithm with PT-Algorithm

1. The advantage of PT-Algorithm, it requires only one computation of $\ln\prod_{i=1}^{n} u_i$ for generating n exponential variates simultaneously while IT-Algorithm requires n computation of lnU for each variates Y_i $(i=1,2,\ldots,n)$ separately.
2. The disadvantage of PT-Algorithm, it need 2n-1 uniform variates while IT-Algorithm requires only n uniform variates.
3. PT-Algorithm requires the arrangement of the uniform variates $U_{n+1}, U_{n+2}, \ldots, U_{2n-1}$ to be order statistics $W_1<W_2<\ldots<W_{n-1}$ and then calculation $W_i - W_{i-1}$ which also time consuming.
4. Simulating both algorithm, we find that PT-Algorithm is faster then IT-Algorithm for n=3, 4, 5, 6. The optimal n is 4.

References

1. Naylor, T.H. etal., Computer Simulation Techniques, Wiley, New York, 1966.
2. Tocher, K.D., The Art of Simulation, Van Nostrand, Princeeton, New Jersey, 1963.
3. Ahrens, J.H. and U. Dieter, Computer Methods for Sampling from the Exponential and normal distribution, Comm. Assoc. Comp. Mech.,15,1972, 873-882.
4. Marsaglia, G., Generating Exponential Random Variables, Ann. Math Stat., 32, 1961, 899-900.
5. Forsythe, G.E., Von Neumann's Comparison Method for Random Sampling and from the Normal and Other Distributions, Math. Comp., 1972, 817-826.
6. Von Neumann, J., Various Techniques used in connection with Random Digits, U.S. Nat. Bur Stand. Appl. Math., No. 12, pp 36-38, 1951.
7. Sobol, J.M., Computational Methods for Monte Carlo, Nauka, Moscow, 1973.