

XIV Conferencia Latinoamericana de Informática
17avas. Jornadas Argentinas de Informática
e Investigación Operativa
Buenos Aires, Setiembre 1988.

TECHICAS EFICIENTES PARA GENERAR VALORES NORMALES Y LOGNORMALES POR EL METODO DEL COCIENTE DE UNIFORMES

R.LAVIERI, R.A.PASTORIZA

Escuela de Computación, Facultad de Ciencias,
Universidad Central de Venezuela.
Apartado 47002. Caracas, Venezuela.

RESUMEN

Se presentan algoritmos de composición basados en el cociente de uniformes para generar valores de variables normales y lognormales. Las pruebas realizadas permiten asegurar una alta eficiencia en relación a la velocidad de ejecución y almacenamiento de estos algoritmos.

INTRODUCCION

El método del cociente de uniformes fue introducido en 1977 por Kinderman y Monahan [1], y resultó particularmente eficiente para generar valores de variables aleatorias normales, aún cuando se trata de un método general para generar valores de ciertas variables aleatorias mediante el muestreo uniforme en una región R del plano en lugar de generarlos empleando la función de distribución $F_X(x)$.

La idea central de este método para generar variables aleatorias con una distribución deseada $F_X(x)$ consiste en generar un punto uniformemente en una región adecuada R del plano. El cociente de los valores de las coordenadas de un punto interior a R proporciona un valor de la variable aleatoria que se reparte según la función de distribución especificada. En la implantación casi siempre se usa la técnica del rechazo ya que en general la región de muestreo R suele ser irregular; es conocido que la eficiencia de un algoritmo de este tipo depende muy fuertemente de la aproximación que se haga mediante la región de acotación C .

Los puntos se generan uniformemente en C , suponiendo que el muestreo en esa región se puede realizar fácilmente; aquellos ensayos cuyos puntos caen fuera de la región R se rechazan, y se vuelve a probar. La eficiencia global del algoritmo depende del valor que tiene $e = [R]/[C]$, cociente de las respectivas áreas.

En general un algoritmo de este tipo se puede mejorar más aún, dividiendo la región R en subregiones disjuntas R_i cada una de las cuales, a su vez, es encerrada en su propia región de acotación C_i .

En síntesis, se estructura entonces un algoritmo de composición donde se emplean los valores de probabilidad $p_i = [R_i]/[C_i]$ para seleccionar la subregion R_i , y luego, al igual que antes, se utiliza la técnica del rechazo para generar los valores deseados.

El uso de pruebas previas que definen regiones de aceptación-rechazo automáticas, poco costosas de realizar computacionalmente, suele mejorar notablemente aún más la eficiencia global del algoritmo.

El método para generar una variable aleatoria X , con densidad $f_X(x)$ es emplear la región R definida mediante la expresión

$$R = \{ (u,v) / 0 \leq u \leq (f(v/u))^{1/2} \}$$

En estas circunstancias, si el par de variables aleatorias (u,v) está distribuido uniformemente en esa región R , el valor del cociente $X = v/u$ tiene la densidad deseada $f_X(x)$.

NORMAL

Para la generación de valores normales $N(\mu, \sigma^2)$ se ha propuesto generar valores X normales standard y efectuar la transformación

$$Z = \sigma * X + \mu$$

Así tomando la función $g(t) = e^{-t^2/2}$

la región definida por

$$u^2 \leq g(v/u)$$

queda dada por la expresión

$$(v/u)^2 \leq -4 * \ln u$$

En 1980 Robertson y Walls [2] describen el uso de ésta técnica para en la generación de valores aleatorios normal y gamma.

Estos autores encierran la región de muestreo en un trapecio compuesto a su vez por dos triángulos y un rectángulo, con zonas de aceptación-rechazo automáticas.

Posteriormente Knuth [3] propone también zonas de aceptación-rechazo automáticas, pero su método se basa en encerrar la región de muestreo mediante figuras curvas.

Aquí propondremos algunas pequeñas mejoras a estas ideas, modificaciones, en algunos aspectos, que permiten acelerar ligeramente el tiempo esperado de ejecución del algoritmo.

La primera modificación está basada en la siguiente observación :

Sea una variable aleatoria $X \sim N(0, \sigma^2)$, entonces el valor transformado siguiente se distribuye también como una normal

$$\{ (\sigma_1/\sigma) * X + \mu \} \sim N(\mu, \sigma_1^2)$$

Tomando como función la

$$g(t) = e^{-t^2/2 \cdot \sigma^2}$$

entonces la región de aceptación es

$$(v_1/u)^2 \leq -4 \cdot \sigma^2 \cdot \ln u \quad \text{con } u > 0$$

Así se propone generar valores $N(0, \sigma^2)$ con varianza $\sigma^2 = e/8$, de manera que los valores aceptados queden enteramente contenidos en la región

$$[0,1] \times [-.5, .5]$$

y utilizar posteriormente la transformación

$$z = (\sigma_1/\sigma) \cdot X + \mu$$

para generar valores $N(0, \sigma_1^2)$.

Algoritmo normal (μ, σ_1)

inicialización

$$A = (e/8)^{1/2} \quad ; \quad B = e/8 \quad ; \quad C = \sigma_1/A \quad ;$$

$$D = 5 \cdot B \quad ; \quad E = 0.5 \cdot e^{1.25} \quad ; \quad F = 0.5 \cdot e^{-.35};$$

$$G = 1.4 \cdot B \quad ; \quad H = -0.5 \cdot e \quad ; \quad P = 27/16$$

paso 1.

generar $(u,v) == \text{unif}[0,1] \times [0,1]$,
poner $X = (v - 0.5)/u$

paso 2.

poner $T = X^2$

si $T \leq D - E \cdot u$, poner $z = C \cdot X + \mu$,
devolver z y finalizar

en caso contrario, continuar

paso 3.

si $T \geq F / u + G$, ir al paso 1

en caso contrario, continuar

paso 4.

si $T \leq H * \ln u$, poner $Z = C * X + \mu$,
devolver Z y finalizar

en caso contrario, ir al paso 1

Los pasos 2 y 3 dependen muy fuertemente de la velocidad con la cual se realiza la operación de logaritmicación. Si ésta operación es relativamente lenta respecto a las operaciones aritméticas simples, se propone otra zona de rechazo automática.

La nueva zona se puede describir de manera muy simple mediante la expresión

$$(v/u)^2 \geq P * (1-u)/8, \text{ con } u > 2/3$$

cuya implantación es el paso 3.5

paso 3.5

si $\{ u > 2/3 \}$ y $\{ T \geq P * (1-u) \}$ ir al paso 1

Pruebas comparativas realizadas con equipos mainframe y microcomputadores - TABLA 1 - permiten asegurar que se obtiene una mejora del orden de un 15% en el tiempo de ejecución, utilizando FORTRAN, respecto a las otras implantaciones comparadas [2] y [3].

LOGNORMAL

Para la generación de valores lognormales $LN(0, \sigma^2)$ tomamos la función

$$g(t) = 1/t * e^{-\log^2 t / 2 * \sigma^2}$$

En este caso la región queda definida por :

$$2 * \sigma^2 * (\log u + \log v) \leq - (\log v - \log u)^2$$

Es fácil ver que la región R queda enteramente contenida en el cuadrado

$$[0, e^{\sigma^2/4}] * [0, e^{\sigma^2/4}]$$

y que es simétrica respecto a la recta $u = v$.

De todas las posibles regiones $R(\sigma)$ se seleccionó la definida por

$$\sigma = 2^{1/2}$$

ya que la misma queda enteramente por debajo de la recta

$$u + v = 2$$

lo cual permite gran simplicidad en la zona de aceptación-rechazo automática.

Sea (u_0, v_0) un par uniforme en el cuadrado

$$[0, e^{1/4}] \times [0, e^{1/4}]$$

Podemos entonces definir una zona de rechazo automática

$$R_1 = \{ (u_0, v_0) / u_0 + v_0 > 2 \}$$

y una zona de aceptación automática

$$R_2 = \{ (u_0, v_0) / (v_0, u_0) \leq e^2 \text{ y } (u_0, v_0) \geq e^{-2} \\ \text{ y } (u_0 + v_0 \leq e^{-3/2} + e^{1/2}) \}$$

Ahora bien si $u == \text{unif}[0,1]$ y $v == \text{unif}[0,1]$ entonces

$$u_0 = e^{1/2} * u$$

y

$$v_0 = e^{1/2} * v$$

por lo que $v_0/u_0 = v/u$, y podemos además describir las regiones como sigue

$$R((2)^{1/2}) = \{ (u, v) / 1 + \log(u) + \log(v) \leq \\ - 1/4 * (\log(u) - \log(v))^2 \}$$

$$R_1 = \{ (u, v) / u + v > 2 * e^{-1/2} \}$$

$$R_2 = \{ (u, v) / (v/u) \leq e^2 \text{ y }$$

$$(v/u) \geq e^{-2} \text{ y } (u + v \leq e^{-2} + 1) \}$$

Ahora bien, si $X == \text{lognormal}(0,2)$ entonces

$$e^{\mu} * X^{\sigma/\sqrt{2}} == \text{lognormal}(\mu, \sigma^2)$$

El algoritmo propuesto es el siguiente :

Algoritmo lognor(0,sigma)

inicialización

$A = 2 * e^{-1/2}$; $B = e^2$; $C = e^{-2}$;

$D = 1 + C$; $E = e^{\mu}$; $F = \text{sigma} / \text{sqrt}(2)$

paso 1.

generar $u, v \sim \text{unif}(0,1)$

paso 2.

hacer $G = u + v$

si $G > A$, ir al paso 1

en caso contrario, continuar

paso 3.

hacer $R = v/u$;

si $(R \leq B)$ y $(R \geq C)$ y $(G \leq D)$,
ir al paso 5

en caso contrario, continuar

paso 4.

hacer $Z = \ln(u)$ y $W = \ln(v)$;

si $(1 + Z + W \leq -.25 * (Z-W)^2)$,
ir al paso 5

en caso contrario, ir al paso 1

paso 5.

hacer $L = E * R^F$; devolver L y finalizar

Referencias

1. Kinderman, A. J. y Monahan, J. F. : "Computer Generation of Random Variables Using the Ratio of Uniform Deviates" ACMTOMS, vol. 3, num. 3, p 257/260, 1977.
2. Robertson, I. y Walls, L. A. : "Random Number Generators for the Normal and Gamma Distributions Using the Ratio of Uniforms Method", Harwell AERE - R 10032, 1980.
3. Knuth, D.E. : "THE ART OF COMPUTER PROGRAMMING", vol. 2, Reading, Mass., second ed., 1981.

TABLA 1

Tiempos promedio obtenidos generando 100.000 valores tras 20 corridas en un equipo Burroughs B-6700, en FORTRAN

Polar	101.3 seg.
Teorema Central del Limite	68.2 seg.
Robertson-Walls	36.9 seg.
Knuth	34.2 seg.
Lavieri-Pastoriza	29.7 seg.

Apéndice

```

***** LISTADO *****
ARCHIVO NORMAL.FOR
*****

```

REAL FUNCTION NORMAL (MU,SIGMA)

OBJETIVO :

GENERAR UN VALOR DE VARIABLE ALEATORIA QUE SE
DISTRIBUYE SEGUN UNA NORMAL N(MU,SIGMA)

PROCEDIMIENTO :

SE EMPLEA EL METODO DEL COCIENTE DE UNIFORMES,
EN BASE A UN ALGORITMO DE COMPOSICION Y LA
TECNICA DEL RECHAZO

VARIABLES DE ENTRADA :

MU : MEDIA DE LA FUNCION DE DENSIDAD NORMAL
SIGMA : DESVIACION STANDARD DE LA FUNCION DE DENSIDAD
NORMAL

VARIABLES DE SALIDA :

Z : VALOR DE LA VARIABLE ALEATORIA
NORMAL(MU,SIGMA)

SUBROUTINAS LLAMADAS :

NINGUNA

REFERENCIAS :

KNUTH,D.E.: THE ART OF COMPUTER PROGRAMMING,
VOL.2 2ND ED., ADDISON-WESLEY, READING, MASS.,
P 121/125, (1981)

LAVIERI,R.y PASTORIZA,R.A: TECNICAS EFICIENTES
PARA GENERAR VALORES NORMALES Y LOGNORMALES
POR EL METODO DE COCIENTES DE UNIFORMES.

PROGRAMADOR :

R.A.P. (1986)

U.C.V. FAC.CIENCIAS

ESCUELA DE COMPUTACION

A / (E/8)^{1/2} / , B / e/8 / , D / 5 * B / ,
E / 0.5 * E^{1.25} / , P / 27/16 / , F / 0.5 * e^{-0.35} / ,
G / 1.4 * (e/8) / , H / -e/2 / , W / 2/3 / ,
FK / 0.45 /

DATA A /.5829109/, B /.3397852/, D /1.6989261/,
1 E /1.7451714/, P /1.68750/, F /.3523440/,
2 G /.4756993/, H /-1.3591409/, W /.66666667/,
3 FK / 0.45 /

```

REAL MU
C = SIGMA/A
30 CONTINUE
U = GNA(S)
V = GNA(S)
X = (V-.5)/U
T = X * X

C
C PRUEBA PREVIA DE ACEPTACION
C
IF ( T .LE. (D- E * U) ) GO TO 200

C
C PRUEBAS PREVIAS DE RECHAZO
C
IF (( U .LT. FK ) .AND. ( T .GE. G + F/U )) GO TO 30
IF (( T .GE. W ) .AND. ( T .GE. (P * (1-U)/8) )) GO TO

30
C
C REGION CURVA PARA LA NORMAL
C
IF ( T .GT. H * ALOG(U) ) GO TO 30
200 Z = C * X + MU
NORMAL = Z
RETURN
END

C
C *****
C

```