Computing 30, 185–188 (1983)

Computing

A Note on Gamma Variate Generators with Shape Parameter less than Unity

D. J. Best, Sydney

Received May 28, 1981

Abstract — Zusammenfassung

A Note on Gamma Variate Generators with Shape Parameter less than Unity. A modification is given for an algorithm of Ahrens and Dieter [1] which generates random Gamma variates with shape parameter less than unity. The modified algorithm is substantially faster, although hardly more complex than the original one.

AMS Subject Classifications: 62E30, 62E25.

Key words and phrases: Random numbers, simulation, gamma distribution, pseudo-random.

Eine Bemerkung über Gamma-Zufallsgeneratoren mit Formparameter kleiner als Eins. Es wird eine Modifikation des Algorithmus von Ahrens und Dieter [1] angegeben, welcher gammaverteilte Zufallsvariable mit einem Formparameter kleiner Eins erzeugt. Der modifizierte Algorithmus ist deutlich schneller, obwohl er kaum komplexer ist als der ursprüngliche.

1. Introduction

In recent years a large number of gamma variate generators have been proposed for the case when the gamma shape parameter, a say, is greater than unity. Cheng and Feast [3] discuss the best of these and give an improved generator valid for $a>0.25$. However, the most competetive generator which covers the complete range $0<a<1$ still appears to be given by algorithm GS of Ahrens and Dieter. It relies on the inequality

$$f(x)=x^{a-1}e^{-x}/\Gamma(a)\leq g(x)$$

where

$$g(x)=x^{a-1}/\Gamma(a) \text{ if } 0\leq x\leq 1;\ g(x)=e^{-x}/\Gamma(a) \text{ if } x\geq 1. \tag{1}$$

Hence,

$$h(x)=\frac{e\,a}{e+a}x^{a-1} \text{ if } 0\leq x\leq 1;\ h(x)=\frac{e\,a}{e+a}e^{-x} \text{ if } x\geq 1 \tag{2}$$

is a density for which $g(x)=\dfrac{e+a}{e\,a\,\Gamma(a)}h(x)$ holds. The acceptance-rejection technique leads to the following algorithm.

0010-485X/83/0030/0185/$ 01.00

Algorithm GS $(0<a\leq 1)$

1. Generate U. Set $b\leftarrow(e+a)/e$ and $P\leftarrow bU$. If $P\geq 1$ go to 3.
2. (Case $x\leq 1$). Set $X\leftarrow P^{1/a}$. Generate U^*. If $U^*>e^{-X}$ go back to 1. Otherwise deliver X.
3. (Case $x>1$). Set $X\leftarrow -\ln((b-P)/a)$. Generate U^*. If $U^*>X^{a-1}$ go back to 1. Otherwise deliver X.

2. A First Modification of Algorithm GS

As in algorithm GS a two part envelope $g(x)$ for the gamma density $f(x)$ is used, but the change point z is no longer fixed at unity. It has already been suggested in the discussion of the paper of Atkinson and Pearce [2] that z should be a function of a: $z=z(a)$. Details of how this revision is to be achieved are not given in that paper and do not appear to be published elsewhere. For the new proposal take

$$g(x)=x^{a-1}/\Gamma(a) \text{ if } 0\leq x\leq z;\ g(x)=z^{a-1}e^{-x}/\Gamma(a) \text{ if } x\geq z; \tag{1'}$$

majorization $g(x)\geq f(x)$ is obvious for all $0<a\leq 1$. The expected number $\alpha(z)$ needed to generate one gamma variate becomes

$$\alpha(z)=\int_0^\infty g(x)\,dx=(z^a/a+z^{a-1}e^{-z})/\Gamma(a), \text{ or}$$

$$\alpha(z)=b\,z^a/(a\,\Gamma(a)) \text{ where } b=1+e^{-z}a/z.$$

Hence, sampling takes place from the density function $h(x)=g(x)/\alpha(z)$:

$$h(x)=(x/z)^{a-1}a/b\,z \text{ if } 0\leq x\leq z;\ h(x)=e^{-x}a/b\,z \text{ if } x\geq z. \tag{2'}$$

The probability of using the first part is b^{-1}. The case $z=1$ yields the old algorithm GS, but the optimum break point z^* is a solution of the transcendental equation

$$z^*=e^{-z^*}(1-a+z^*)$$

which minimizes $\alpha(z)$. In the revised version of GS, say RGS, z^* is approximated – after some trial and error – by the easy expression

$$\hat{z}=.07+.75(1-a)^{1/2}\approx z^*$$

which is shown to be a good fit in Table 1: the differences $\alpha(\hat{z})-\alpha(z^*)$ are small, whereas the expected number of trials $\alpha(1)$ for GS are substantially larger than $\alpha(z^*)$, particularly towards the bottom of the table.

3. A Second Modification of Algorithm GS

The speed of the algorithm may be further improved by avoiding the exponentiation in steps 2 and 3 of GS with high probability. For this two inequalities are needed.

$$e^{-x}\geq(2-x)/(2+x) \quad \text{if } x\geq 0 \tag{3}$$

$$(1+x)^{-c}\geq(1+c\,x)^{-1} \quad \text{if } x\geq 0,\ 1\geq c\geq 0. \tag{4}$$

For proving (3) and (4) consider

$$q(x)=\frac{2-x}{2+x}e^x \text{ which implies } q(0)=1,\ \frac{q'(x)}{q(x)}=-\frac{x^2}{4-x^2}<0,\ x<2$$

$$r(x)=\frac{(1+x)^c}{1+cx} \text{ which implies } r(0)=1,\ \frac{r'(x)}{r(x)}=-\frac{c(1-c)x}{(1+x)(1+cx)}<0.$$

Hence $x=0$ is the only maximum for both functions.

4. The Final Algorithm RGS

Algorithm RGS $(0<a<1)$

0. Initialize $z\leftarrow .07+.75(1-a)^{1/2}$, $b\leftarrow 1+e^{-z}a/z$.
1. Generate U and set $P\leftarrow bU$. If $P>1$ go to 4.
2. Set $X\leftarrow zP^{1/a}$. Generate U^*. If $U^*\leq(2-X)/(2+X)$, deliver X.
3. If $U^*>e^{-X}$ go to 1, otherwise deliver X.
4. Set $X\leftarrow -\ln(z(b-P)/a)$, $Y\leftarrow X/z$. Generate U^*.
   If $U^*(a+Y-aY)<1$, deliver X.
5. If $U^*>Y^{a-1}$ go to 1, otherwise deliver X.

Step 0 is only performed once if the parameter a does not change during the sample generation.

Table 1 compares times – when step 0 is performed once – for algorithm GS with those of RGS on a CDC 7600 computer. Comparisons were also made on a PDP 11/34, a much slower computer, using the manufacturer's random uniform generator. The ratio of times was found to be very similar to those of Table 1. The times are based, as seems common practice, on samples of 10,000 produced by Fortran DO loop with function call. The times given in Table 1 indicate considerable time savings for the case of constant parameter a by using RGS rather than GS particularly as a approaches unity.

Table 1. *Time in μsecs to generate a gamma variate on a CDC 7600 with FTN 4.6 compiler, optimization level l, and RANF random uniform variate generator*

a	z^*	$\alpha(z^*)$	$\alpha(\hat{z})-\alpha(z^*)$	Time RGS [μsec]	Time GS [μsec]	$\alpha(1)$
ε	.8065	1.0000	.000 0000	15.7	18.1	1.0000
0.1	.7725	1.0856	.000 0091	15.7	18.5	1.0898
0.2	.7358	1.1577	.000 0059	17.6	20.6	1.1693
0.3	.6960	1.2142	.000 0008	18.4	21.6	1.2372
0.4	.6523	1.2534	.000 0009	18.5	22.3	1.2929
0.5	.6035	1.2738	.000 0064	18.1	22.3	1.3359
0.6	.5480	1.2739	.000 0104	18.4	23.4	1.3662
0.7	.4831	1.2525	.000 0045	17.1	23.1	1.3840
0.8	.4030	1.2074	.000 0055	16.7	23.2	1.3897
0.9	.2933	1.1337	.000 1844	14.6	23.3	1.3840
$1-\varepsilon$	.0000	1.0000	.002 3938	11.6	23.3	1.3679

The author wishes to acknowledge the help of the referees in improving the presentation of the results.

References

[1] Ahrens, J. H., Dieter, U.: Computer methods for sampling from gamma, beta, Poisson and binomial distributions. Computing *12*, 223–246 (1974).
[2] Atkinson, A. C., Pearce, M. C.: The computer generation of beta, gamma, and normal random variables (with discussion). J. Royal Stat. Soc. *A139*, 431–461 (1976).
[3] Cheng, R. C. H., Feast, G. M.: Gamma variate generators with increased shape parameter range. Comm. ACM *23, 389–394 (1980)*.

D. J. Best
CSIRO, DMS
P.O. Box 52
North Ryde, 2113
Australia

Verleger: Springer-Verlag, Mölkerbastei 5, A-1010 Wien. — Herausgeber: Prof. Dr. Hans J. Stetter, Institut für Numerische Mathematik der Technischen Universität Wien, Gußhausstraße 27—29, A-1040 Wien. — Redaktion: Gußhausstraße 27—29, A-1040 Wien. — Hersteller: Satz Austro-Filmsatz Richard Gerin, Zirkusgasse 13, A-1020 Wien, Druck Paul Gerin, Zirkusgasse 13, A-1021 Wien. — Verlagsort: Wien. — Herstellungsort: Wien. — Printed in Austria.