Generating Random Variables

Christian Robert*
Université Paris Dauphine and CREST, INSEE

George Casella†
University of Florida

Keywords and Phrases: Random Number Generator, Probability Integral Transform, Accept Reject Algorithm, Metroplis-Hastings Algorithm, Gibbs Sampling, Markov Chains

1 Introduction

The methods described in this article mostly rely on the possibility of producing (with a computer) a supposedly endless flow of random variables (usually iid) for well-known distributions. Such a simulation is, in turn, based on the production of uniform random variables. There are many ways in which uniform pseudorandom numbers can be generated. For example there is the *Kiss* algorithm of Marsaglia and Zaman (1993); details on other random number generators can be found in the books of Rubinstein (1981), Ripley (1987), Fishman (1996), and Knuth (1998).

2 Generating Nonuniform Random Variables

The generation of random variables that are uniform on the interval $[0, 1]$, the Uniform $[0, 1]$ distribution, provides the basic probabilistic representation of randomness. The book by Devroye (1986) is a detailed discussion of methods for generating nonuniform variates, and the subject is one of the many covered in Knuth (1998). Formally, we can generate random variables with any distribution by means of the following.

*Université Paris Dauphine, 75785 Paris cedex 16, France, and CREST, INSEE. Email: `xian@ceremade.dauphine.fr`

†Department of Statistics and Genetics Institute, University of Florida, Gainesville, FL 32611. Email: `casella@ufl.edu`

2.1 Probability Integral Transform

For a function F on $\Re$, the *generalized inverse* of F is $F^{-}(u) = \inf \{x : F(x) \geq u\}$. If F is a cumulative distribution function (cdf), $F^{-}(U)$ has the cdf F and to generate $X \sim F$,

1. Generate U according to Uniform$[0, 1]$
2. Make the transformation $x = F^{-}(u)$.

2.2 Generating Discrete Variables

For discrete distributions, if the random variable X satisfies

$$P(X = k) = p_k, \quad k = 0, 1, 2, \ldots$$

then if $U \sim$ uniform$[0, 1]$, X can be generated by

$$\begin{aligned} k = 0: \quad & U \leq p_0 \Rightarrow X = 0, \\ k = 1, 2, \ldots,: \quad & \sum_{j=0}^{k-1} p_j \leq U < \sum_{j=0}^{k} p_j \Rightarrow X = k. \end{aligned}$$

2.3 Mixture Distributions

In a mixture representation a density f has the form

$$\begin{aligned} f(x) &= \int_{\mathcal{Y}} f(x|y)g(y)\, dy \quad \text{(continuous)} \\ f(x) &= \sum_{i \in \mathcal{Y}} p_i\, f_i(x) \quad \text{(discrete).} \end{aligned}$$

We then can simulate X as

1. $Y \sim g(y)$, $X \sim f(x|Y = y)$, or
2. $P(Y = i) = p_i$, $X \sim f_i(x)$

3 Accept-Reject Methods

Another class of methods only requires the form of the density f of interest - called the *target density*. We simulate from a density g, called the *candidate density*.

Given a target density f, we need a candidate density g and a constant M such that

$$f(x) \leq Mg(x)$$

on the support of f.

3.1 Accept-Reject Algorithm:

To produce a variable X distributed according to f:

1. Generate $Y \sim g$, $U \sim$ Uniform$[0,1]$;
2. Accept $X = Y$ if $U \leq f(Y)/Mg(Y)$;
3. Otherwise, return to 1.

Notes:

(a). The densities f and g need be known only up to a multiplicative factor.

(b). The probability of acceptance is $1/M$, when evaluated for the normalized densities.

3.2 Envelope Accept-Reject Algorithm:

If the target density f is difficult to evaluate, the *Envelope* Accept-Reject Algorithm (called the *squeeze principle* by Marsaglia 1977) may be appropriate.

If there exist a density g_m, a function g_l and a constant M such that

$$g_l(x) \leq f(x) \leq Mg_m(x) ,$$

then the algorithm

1. Generate $X \sim g_m(x)$, $U \sim$ Uniform$[0,1]$;
2. Accept X if $U \leq g_l(X)/Mg_m(X)$;
3. Otherwise, accept X if $U \leq f(X)/Mg_m(X)$
4. Otherwise, return to 1.

produces $X \sim f$. The number of evaluations of f is potentially decreased by a factor $\frac{1}{M}$. If f is *log-concave*, Gilks and Wild (1992) construct a generic accept-reject algorithm that can be quite efficient.

4 Markov Chain Methods

Every simulation method discussed thus far has produced independent random variables whose distribution is exactly the target distribution. In contrast, Markov chain methods produce a sequence of dependent random variables whose distribution converges to the target. Their advantage is their applicability in complex situations.

Recall that a sequence $X_0, X_1, \ldots, X_n, \ldots$ of random variables is a *Markov chain* if, for any t, the conditional distribution of X_t given $x_{t-1}, x_{t-2}, \ldots, x_0$ is the same as the distribution of X_t given x_{t-1}; that is,

$$P(X_{k+1} \in A | x_0, x_1, x_2, \ldots, x_k) = P(X_{k+1} \in A | x_k).$$

4.1 Metropolis - Hastings Algorithm

The Metropolis-Hastings (M-H) algorithm (Metropolis *et al.* 1953, Hastings 1970) associated with the target density f and the candidate density $q(y|x)$ produces a Markov chain $(X^{(t)})$ through

Metropolis -Hastings Given $x^{(t)}$,

1. Generate $Y_t \sim q(y|x^{(t)})$.
2. Take
$$X^{(t+1)} = \begin{cases} Y_t & \text{with prob. } \rho(x^{(t)}, Y_t) \\ x^{(t)} & \text{with prob. } 1 - \rho(x^{(t)}, Y_t) \end{cases}$$
where
$$\rho(x, y) = \min \left\{ \frac{f(y)}{f(x)} \frac{q(x|y)}{q(y|x)}, 1 \right\},$$
3. Then $X^{(t)}$ converges in distribution to $X \sim f$.

Notes:

(a). For $q(\cdot|x) = q(\cdot)$ we have the *independent* M-H algorithm, and for $q(x|y) = q(y|x)$ we have a symmetric M-H algorithm, where ρ does not depend on q. Also, $q(x|y) = q(y - x)$, symmetric around zero, is a *random walk* M-H algorithm.

(b). Like the Accept-Reject method, the Metropolis - Hastings algorithm only requires knowing f and q up to normalizing constants.

4.2 The Gibbs Sampler

For $p > 1$, write the random variable $\mathbf{X} \in \mathcal{X}$ as $\mathbf{X} = (X_1, \ldots, X_p) \sim f$, where the X_i's are either uni- or multidimensional, with conditional distributions

$$X_i | x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_p \sim f_i(x_i | x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_p)$$

for $i = 1, 2, \ldots, p$. The associated *Gibbs sampler* is given by the following transition from $X^{(t)}$ to $X^{(t+1)}$:

The Gibbs sampler Given $\mathbf{x}^{(t)} = (x_1^{(t)}, \ldots, x_p^{(t)})$, generate

1. $X_1^{(t+1)} \sim f_1(x_1 | x_2^{(t)}, \ldots, x_p^{(t)})$;
2. $X_2^{(t+1)} \sim f_2(x_2 | x_1^{(t+1)}, x_3^{(t)}, \ldots, x_p^{(t)})$,

$\vdots$

p. $X_p^{(t+1)} \sim f_p(x_p | x_1^{(t+1)}, \ldots, x_{p-1}^{(t+1)})$,

then $X^{(t)}$ converges in distribution to $X \sim f$.

Notes:

(a). The densities $f_1, \ldots, f_p$ are called the *full univariate conditionals*.

(b). Even in a high dimensional problem, all of the simulations can be univariate.

(c). The Gibbs sampler is, formally, a special case of the M-H algorithm (see Robert and Casella 2004, Section 10.2.2) but with acceptance rate equal to 1.

4.3 The Slice Sampler

A particular version of the Gibbs sampler, called the *slice sampler* (Besag and Green 1993, Damien *et al.* 1999), can sometimes be useful. Write $f(x) = \prod_{i=1}^{k} f_i(x)$ where the f_i's are positive functions, not necessarily densities

Slice sampler Simulate

`1.` $\omega_1^{(t+1)} \sim \text{Uniform}_{[0,f_1(x^{(t)})]}$;

$\ldots$

`k.` $\omega_k^{(t+1)} \sim \text{Uniform}_{[0,f_k(x^{(t)})]}$;

`k+1.` $x^{(t+1)} \sim \text{Uniform}_{A^{(t+1)}}$, `with`

$$A^{(t+1)} = \{y;\ f_i(y) \geq \omega_i^{(t+1)},\ i = 1, \ldots, k\}.$$

then $X^{(t)}$ converges in distribution to $X \sim f$.

5 Application

In this section we give some guidelines for simulating different distributions. See Robert and Casella (2004, 2010) for more detailed explanations, and Devroye (1986) for more algorithms. In what follows, U is Uniform$(0, 1)$ unless otherwise specified.

Arcsine $f(x) = \frac{1}{\pi\sqrt{x(1-x)}}$, $0 \leq x \leq 1$, $\sin^2(\pi U/2) \sim f$.

Beta(r, s) $f(x) = \frac{\Gamma(r+s)}{\Gamma(r)\Gamma(s)} x^{r-1}(1-x)^{s-1}, 0 \leq x \leq 1, r > 0, s > 0, \frac{X_1}{X_1+X_2} \sim f$, where $X_1 \sim$ Gamma $(r, 1)$ and $X_2 \sim$ Gamma $(s, 1)$, independent.

Cauchy(μ, σ) $f(x|\mu, \sigma) = \frac{1}{\sigma\pi} \frac{1}{1+\frac{(x-\mu)^2}{\sigma^2}}, -\infty < x < \infty, \sigma \tan\left(\frac{\pi}{2}(2U-1)\right) + \mu \sim$ Cauchy (μ, σ).

If $U \sim$ uniform$[-\pi/2, \pi/2]$, then $\tan(U) \sim$ Cauchy$(0, 1)$. Also, $X/Y \sim$ Cauchy $(0, 1)$, where $X, Y, \sim N(0, 1)$, independent.

Chi squared(p) $f(x|p) = \frac{1}{\Gamma(p/2)2^{p/2}} x^{(p/2)-1} e^{-x/2}, 0 \leq x < \infty, p = 1, 2, \ldots$.

$$-2\sum_{j=1}^{\nu} \log(U_j) \sim \chi^2_{2\nu},\ \text{or}\ \chi^2_{2\nu} + Z^2 \sim \chi^2_{2\nu+1},$$

where Z is an independent Normal$(0, 1)$ random variable.

Double exponential(μ, σ) $f(x|\mu, \sigma) = \frac{1}{2\beta}e^{-|x-\mu|/\beta}$, $-\infty < x < \infty$, $-\infty < \mu < \infty$, $\beta > 0$. If $Y \sim$ Exponential(β), then attaching a random sign (+ if $U > .5$, − otherwise) gives $X \sim$ Double exponential$(0, 1)$, and $\sigma X + \mu \sim$ Double exponential(μ, σ). This is also known as the *Laplace* distribution.

Exponential(β) $f(x|\beta) = \frac{1}{\beta}e^{-x/\beta}$, $0 \leq x < \infty$, $\beta > 0$, $-\beta \log U \sim$ Exponential(β).

Extreme Value(α, γ) $f(x|\alpha, \gamma) = e^{-\frac{x-\alpha}{\gamma} - e^{-\frac{x-\alpha}{\gamma}}}$, $\alpha \leq x < \infty$, $\gamma > 0$. If $X \sim$ Exponential(1), $\alpha - \gamma \log(X) \sim$ Extreme Value(α, γ). This is also known as the *Gumbel* distribution.

F Distribution $f(x|\nu_1, \nu_2) = \frac{\Gamma\left(\frac{\nu_1+\nu_2}{2}\right)}{\Gamma(\frac{\nu_1}{2})\Gamma(\frac{\nu_2}{2})} \left(\frac{\nu_1}{\nu_2}\right)^{\nu_1/2} \frac{x^{(\nu_1-2)/2}}{\left(1+\left(\frac{\nu_1}{\nu_2}\right)x\right)^{(\nu_1+\nu_2)/2}}$, $0 \leq x < \infty$. If $X_1 \sim \chi^2_{\nu_1}$ and $X_2 \sim \chi^2_{\nu_2}$, independent, $\frac{X_1/\nu_1}{X_2/\nu_2} \sim f(x|\nu_1, \nu_2)$.

Gamma(α, β) $f(x|\alpha, \beta) = \frac{1}{\Gamma(\alpha)\beta^\alpha}x^{\alpha-1}e^{-x/\beta}$, $0 \leq x < \infty$, α, $\beta > 0$. Then $-\beta \sum_{j=1}^{a} \log(U_j) \sim$ Gamma(a, β), a an integer.

If α is not an integer, indirect methods can be used. For example, to generate a Gamma(α, β) use Algorithm 3.1 or 4.1 with candidate distribution Gamma(a, b), with $a = [\alpha]$ and $b = \beta\alpha/a$, where $[\alpha]$ is the greatest integer less than α. For the Accept-Reject algorithm the bound on the normalized f/g is $M = \frac{\Gamma(a)}{\Gamma(\alpha)} \frac{\alpha^\alpha}{a^a} e^{-(\alpha-a)}$. There are many other efficient algorithms.

Note: If $X \sim$ Gamma$(\alpha, 1)$ then $\beta X \sim$ Gamma(α, β). Some special cases are Exponential (1) = gamma$(1, 1)$, and Chi squared (p)= gamma $(p/2, 2)$. Also, $1/X$ has the *inverted (or inverse) gamma distribution.*

Logistic(μ, β) $f(x|\mu, \beta) = \frac{1}{\beta} \frac{e^{-(x-\mu)/\beta}}{\left[1+e^{-(x-\mu)/\beta}\right]^2}$, $-\infty < x < \infty$, $-\infty < \mu < \infty$, $\beta > 0$.

$$-\beta \log\left(\frac{1-U}{U}\right) + \mu \sim \text{Logistic}(\mu, \beta)$$

Lognormal(μ, σ^2) $\quad f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \frac{e^{-(\log x - \mu)^2/(2\sigma^2)}}{x}$, $0 \le x < \infty$, $-\infty < \mu < \infty$. If $X \sim \text{Normal}(\mu, \sigma^2)$.

$$e^X \sim \text{ Lognormal}(\mu, \sigma^2)$$

Noncentral chi squared (λ, p)**,** $\lambda \ge 0$ $\quad f_p(x|\lambda) = \sum_{k=0}^{\infty} \frac{x^{p/2+k-1} e^{-x/2}}{\Gamma(p/2+k) 2^{p/2+k}} \frac{\lambda^k e^{-\lambda}}{k!}$, $0 < x < \infty$.

$$K \sim \text{ Poisson } (\lambda/2), \quad X \sim \chi^2_{p+2K} \Rightarrow X \sim f_p(x|\lambda)$$

where p is the degrees of freedom and λ is the noncentrality parameter. A more efficient algorithm is

$$Z \sim \chi^2_{p-1} \text{ and } Y \sim \text{N } (\sqrt{\lambda}, 1) \Rightarrow Z + Y^2 \sim f_p(x|\lambda).$$

Normal(μ, σ^2) $\quad f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/(2\sigma^2)}$, $-\infty < x < \infty$, $-\infty < \mu < \infty$, $\sigma > 0$. The *Box-Muller algorithm* simulates two normals from two uniforms:

$$X_1 = \sqrt{-2\log(U_1)}\ \cos(2\pi U_2) \text{ and } X_2 = \sqrt{-2\log(U_1)}\ \sin(2\pi U_2)\ ,$$

then $X_1, X_2 \sim \text{ Normal}(\mu, \sigma^2)$.

There are many other ways to generate normal random variables.

(a). *Accept Reject using Cauchy* When $f(x) = (1/\sqrt{2\pi}) \exp(-x^2/2)$ and $g(x) = (1/\pi)1/(1+x^2)$, densities of the normal and Cauchy distributions, respectively, then $f(x)/g(x) \le \sqrt{2\pi/e} = 1.520$.

(b). *Accept Reject using double exponential* When $f(x) = \frac{1}{\sqrt{2\pi}} \exp(-x^2/2)$ and $g(x) = (1/2)\exp(-|x|)$, $f(x)/g(x) \le \sqrt{2e/\pi} = 1.315$.

(c). *Slice Sampler*

$$W|x \sim \text{ uniform } [0, \exp(-x^2/2)]\ , \qquad X|w \sim \text{ uniform } [-\sqrt{-2\log(w)}, \sqrt{-2\log(w)}]\ ,$$

yields $X \sim N(0, 1)$.

Pareto(α, β)**,** $\alpha > 0$, $\quad \beta > 0$ $\quad f(x|\alpha, \beta) = \frac{\beta\alpha^\beta}{x^{\beta+1}}$, $\alpha < x < \infty$,

$$\frac{\alpha}{(1-U)^{1/\beta}} \sim \text{ Pareto } (\alpha, \beta)$$

Student's t_ν $\quad f(x|\nu) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)} \frac{1}{\sqrt{\nu\pi}} \frac{1}{\left(1+\left(\frac{x^2}{\nu}\right)\right)^{(\nu+1)/2}}$, $-\infty < x < \infty$, $\nu = 1, 2, \ldots$.

$$Y \sim \chi^2_\nu \text{ and } X|y \sim \text{ N } (0, \nu/y) \Rightarrow X \sim t_\nu.$$

Also, if $X_1 \sim \text{N}(0,1)$ and $X_2 \sim \chi^2_\nu$, then $X_1/\sqrt{X_2/\nu} \sim t_\nu$.

Uniform(a, b) $\quad f(x|a, b) = \frac{1}{b-a}$, $a \leq x \leq b$,

$$(b-a)U + a \sim \text{ Uniform}(a, b).$$

Weibull(γ, β) $\quad f(x|\gamma, \beta) = \frac{\gamma}{\beta} x^{\gamma-1} e^{-x^\gamma/\beta}$, $0 \leq x < \infty$, $\gamma > 0$, $\beta > 0$.

$$X \sim \text{Exponential}(\beta) \Rightarrow X^{1/\gamma} \sim \text{Weibull}(\gamma, \beta).$$

6 References

1. Besag, J. and Green, P.J. (1993) Spatial statistics and Bayesian computation (with discussion). *J. Roy. Statist. Soc. B* **55**, 25–38.

2. Damien, P., Wakefield, J. and Walker, S. (1999) Gibbs sampling for Bayesian non-conjugate and hierarchical models by using auxiliary variables. *J. Roy. Statist. Soc. B* **61**(2), 331–344.

3. Devroye, L. (1986) *Non-Uniform Random Variate Generation.* New York: Springer-Verlag.

4. Fishman, G.S. (1996) *Monte Carlo.* New York: Springer-Verlag

5. Gilks, W.R. and Wild, P. (1992) Adaptive rejection sampling for Gibbs sampling. *Appl. Statist.* **41**, 337–348.

6. Hastings, W. (1970). Monte Carlo Sampling Methods Using Markov Chains and their Application. *Biometrika* **57** 97-109.

7. Knuth, D. E. (1998). *The Art of Computer Programming, Volume 2/ Seminumerical Algorithms, Third Edition.* Addison Wesley, Reading.

8. Marsaglia, G. (1977) The squeeze method for generating gamma variables. *Computers and Mathematics with Applications*, **3**, 321–325.

9. Marsaglia, G. and Zaman, A. (1991). *A new class of random number generators.* Annals Applied Probability **1**, pp. 462–480.

10. Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E. (1953) Equations of state calculations by fast computing machines. *J. Chem. Phys.* **21**, 1087–1092.

11. Ripley, B.D. (1987) *Stochastic Simulation.* New York: John Wiley

12. Robert, C. P. and Casella, G. (2004). *Monte Carlo Statistical Methods, Second Edition.* New York: Springer-Verlag.

13. Robert, C. P. and Casella, G. (2010). *An Introduction to Monte Carlo Methods with* `R`. New York: Springer-Verlag.

14. Rubinstein, R.Y. (1981) *Simulation and the Monte Carlo Method.* New York: John Wiley