Monte Carlo Methods

David Madigan
Columbia University

Recall the strong theorem of large numbers: let $x_1, x_2, \ldots, x_N$ be a sequence of independent random variables having a common distribution and let $E(x_i) = \mu$. Then with probability 1,

$$\frac{x_1 + x_2 + \ldots + x_N}{N} \to \mu \quad as\, N \to \infty .$$

It follows that one way to estimate $E[x] = \int x f_x(x) dx$ is to simulate $x_1, x_2, \ldots, x_N$ from $f_x(x)$ and form the arithmetic mean of the x's as the estimate.

In general can estimate

$$E[g(x)] = \int\int \cdots \int g(x_1, x_2, \ldots, x_N) f(x_1, x_2, \ldots, x_N) dx_1 dx_2 \ldots dx_N \text{ by } \frac{1}{N}\sum_i g(x^{(i)})$$

where $x^{(i)} = (x_1^{(i)}, x_2^{(i)}, \ldots, x_N^{(i)})$ are drawn independently from $f(x_1, x_2, \ldots, x_N)$.

Suppose a random variable X has a beta distribution with parameters a, b:

$$f_x(x) = \frac{1}{B(a,b)} x^{a-1}(1-x)^{b-1} \text{ if } x \in [0,1]$$

where $B(a,b) = \int_0^1 x^{a-1}(1-x)^{b-1}\,dx$ is called a Beta function.

We want to evaluate $P(X \le 0.25) = \int_0^{0.25} \frac{1}{B(a,b)} x^{a-1}(1-x)^{b-1}\,dx$. This is an incomplete beta function which has been tabulated.

Alternatively define $J(x) = \begin{cases} 1 & x \in [0, 0.25] \\ 0 & otherwise \end{cases}$

Then $P(X \le 0.25) = E[J(x)] = \int_0^1 J(x) f_x(x)\,dx$

So estimate $P(X \le 0.25)$ by simulating $x_1, x_2, \ldots, x_N$ from $f_x(x)$ and letting $P(X \le 0.25) \approx \frac{1}{N}\sum J(x_i)$.

This general procedure is called Monte Carlo Integration.

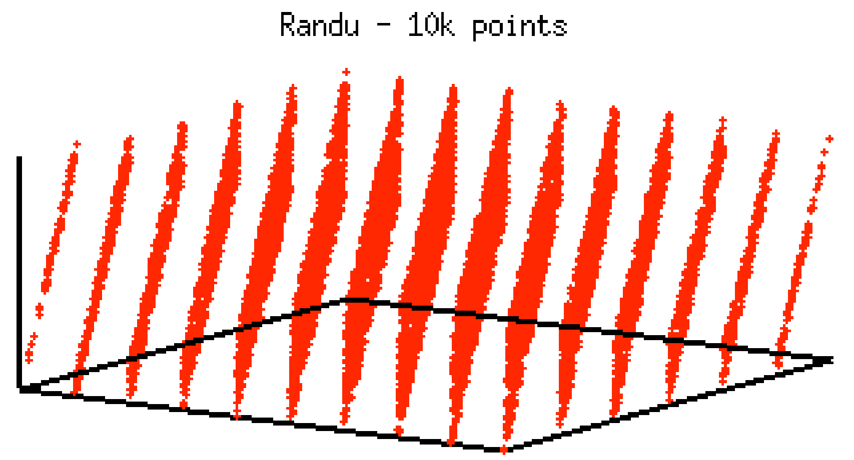
How do you draw random variables from $f_x(x)$?

- Underlying many of the methods we will discuss for simulation will be the requirement to generate random number e.g. a random number between 0 and 1.
- Random number generation and assessing random number generators is a major industry in itself. We will just assume that we have <u>some</u> method for doing it.

e.g. $X_{n+1} = (aX_n + c) \bmod m$ then $\frac{X_n}{m}$ is approximately $U(0,1)$.

Randu - a (bad) pseudo-random number generator.

$r_{i+1}=65539 * r_i \text{ modulo } 2^{31}$



cerd@colorado.edu

Methods for Simulating Continuous Random Variables

The Inverse Transformation Method (method 1)

Proposition:
Consider a random variable X with CDF $F_X(x)$. Define a new random variable $Y = F_X(x)$. Then $Y \sim unif(0,1)$.

Proof:

$$
\begin{aligned}
F_Y(y) &= P(Y \le y) \\
&= P(F_X(x) \le y) \\
&= P(X \le F_X^{-1}(y)) \quad \text{since } F_X \text{ is monotone} \\
&= F_X(F_X^{-1}(y)) = y \quad \text{so } Y \sim unif(0,1)
\end{aligned}
$$

Corollary:

Let $U \sim unif(0,1)$. Define a random variable $X = F^{-1}(U)$ where F is a CDF. Then F is the CDF of X.

Proof:

$$F_X(x) = P(X \le x) = P(F^{-1}(u) \le x) = P(u \le F(x)) = F(x)$$

.

Thus, <u>if</u> we know how to invert F_X, we can simulate X by generating a random u ~ *U(0, 1)* and letting $X = F^{-1}(u)$.

Example:

Simulating a Weibull. For a Weibull random variable *X* have:

$$F_X(x) = 1 - e^{-(x/\beta)^\alpha}$$

Then $F^{-1}(u)$ is that value of *X* such that:

$$1 - e^{-(x/\beta)^\alpha} = u$$
$$\Rightarrow e^{-(x/\beta)^\alpha} = 1 - u$$
$$\Rightarrow \left(\frac{x}{\beta}\right)^\alpha = -\log(1-u)$$
$$\Rightarrow x = \beta(-\log(1-u))^{1/\alpha}$$

X generated in this fashion will have Weibull (α, β) distribution.

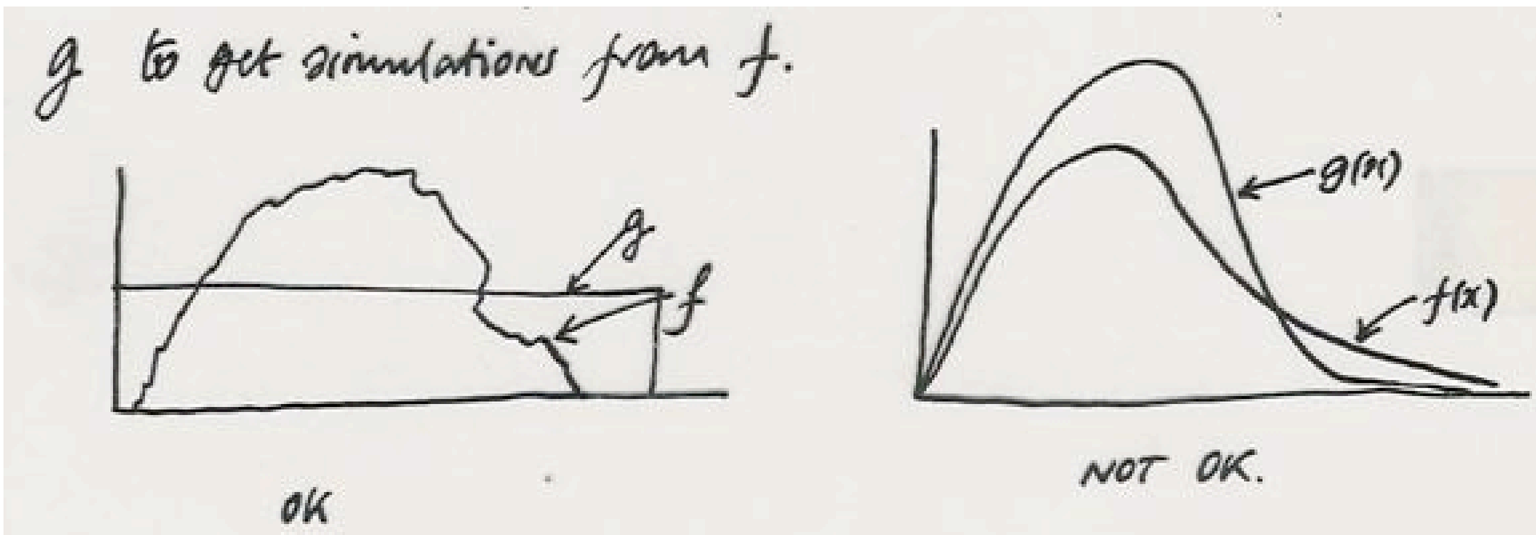
Note:

This method only works when F_X is invertible which will not often be the case.

The rejection method (method 2)

Suppose there is a density $g(x)$ which is "close" to the density f that we wish to simulate from but it is much easier to simulate from g than f (e.g. f might be gamma and g Weibull). Then provided c such that

$$\frac{f(x)}{g(x)} \leq c$$ for all x, we can use g to get simulations from f.



Note:

$g(x)$ must have support at least as big as $f(x)$.

Here is how it works:

<u>Step 1</u> Simulate Y having density g and simulate a random number U.

<u>Step 2</u> If $U \le \frac{f(Y)}{cg(Y)}$ set $X = Y$. Otherwise return to step 1.

Claim: that the value X has density function f.

Proof:

$$P(X \le x) = P(Y_N \le x) = P\left(Y < x \,\middle|\, u \le \frac{f(Y)}{cg(Y)}\right)$$

$$= \frac{P(Y < x \cap u \le \frac{f(Y)}{cg(Y)})}{P\left(u \le \frac{f(Y)}{cg(Y)}\right)} = \frac{\int_{-\infty}^{\infty} P(Y < x \cap u \le \frac{f(Y)}{cg(Y)} | Y = y) g(y) dy}{P\left(u \le \frac{f(Y)}{cg(Y)}\right)}$$

$$= \frac{\int_{-\infty}^{x} \frac{f(y)}{cg(y)} g(y) dy}{P\left(u \le \frac{f(Y)}{cg(Y)}\right)} = \frac{\int_{-\infty}^{x} f(y) dy}{cP\left(u \le \frac{f(Y)}{cg(Y)}\right)}$$

The denominator does not involve x.

But $\lim_{x \to \infty} P(X \le x) = 1 \Rightarrow cP\left(u \le \frac{f(Y)}{cg(Y)}\right) = 1 \Rightarrow P\left(u \le \frac{f(Y)}{cg(Y)}\right) = \frac{1}{c}$.

Note:

Since $P\left(u \le \frac{f(Y)}{cg(Y)}\right) = \frac{1}{c}$, the number of iterations until an acceptance will be geometric with mean c. Thus it is important to choose $g(Y)$ so that c is small.

Note:

However, a difficulty with the rejection method is that in many applications, c is hard to compute. Often end up choosing c very conservatively large thereby resulting in very high computational costs.

```
myrbeta<-function(n,a,b) {
  c <- dbeta((a-1)/((a-1)+(b-1)),a,b) # density value at the mode
  u <- runif(n);  # random numbers
  g <- runif(n);  # candidate draws
  acceptYN <- (u <= (dbeta(g,a,b)/c)); # f/cg
  return(g[acceptYN]);
}


par(mfrow=c(2,1))
plot(density(rbeta(10000,3,2)))
plot(density(myrbeta(10000,3,2)))
```

Sampling Importance Resampling (method 3) (Rubin, 1987)

Again, assume there is a density $g(x)$ which is close to the density *f* that we want to simulate from. Then to generate a sample of size n from *f*, proceed as follows:

1. draw $x_1, x_2, \ldots, x_N$ from $g(x)$
2. sample a value *x* from the set $(x_1, x_2, \ldots, x_N)$ where the probability of sampling each x_i is proportional to $w(x) = \dfrac{f(x)}{g(x)}$
3. sample a second value *x* using the same procedure, but excluding the already sampled value from the set
4. repeatedly sample without replacement $n - 2$ more times.

Adaptive Rejection Sampling

Appl. Statist. (1992)
41, *No.* 2, *pp.* 337–348

Adaptive Rejection Sampling for Gibbs Sampling

By W. R. GILKS†

Medical Research Council Biostatistics Unit, Cambridge, UK

and P. WILD

National Institute for Research and Safety in Occupational Health, Vandoeuvre, France

Non-Adaptive Upper-Lower Rejection Sampling

Want to sample from $f(x)$, $x \in D$

Suppose $g(x) = c\,f(x)$ for some (possibly unknown) constant c

Define an <u>envelope function</u> $g_u(x)$ such that $g_u(x) \geq g\,(x)\ \forall\ x \in D$

Define a <u>squeezing function</u> $g_l(x)$ such that $g_l(x) \leq g\,(x)\ \forall\ x \in D$

Sample a value x^* from $g_u(x)$, and sample a value w independently from the uniform(0, 1) distribution. If you have defined a $g_l(x)$-function, perform the following squeezing test: if

$$w \leqslant g_l(x^*)/g_u(x^*)$$

then accept x^*. Otherwise evaluate $g(x^*)$ and perform the following rejection test: if

$$w \leqslant g(x^*)/g_u(x^*)$$

then accept x^*; otherwise reject x^*. Repeat until n points have been accepted.

Note: this amounts to accepting with probability g/g_u

But: $$\frac{g}{g_u} = \frac{cf}{g_u} = \frac{f}{c' g_u} \text{ where } c' = \frac{1}{c}$$

$$g_u \geq g \Rightarrow g_u \geq cf \Rightarrow \frac{f}{g_u} \leq c'$$

So this is plain old rejection sampling but minimizes the number of evaluations of g

Adaptive Rejection Sampling

Let $h(x) = \log g(x)$ and assume that $h(x)$ is concave

(i.e., $h'(x)$ decreases monotonically with increasing x in D)

First evaluate $h(x)$ and $h'(x)$ at $x_1 \leq x_2 \leq \ldots \leq x_k \in D$

Define the envelope u_k as the piecewise linear upper hull formed from the tangents to $h(x)$ at $T_k = \{x_1, x_2, \ldots, x_k\}$.

Define the squeezing function l_k as the lower hull in the picture:

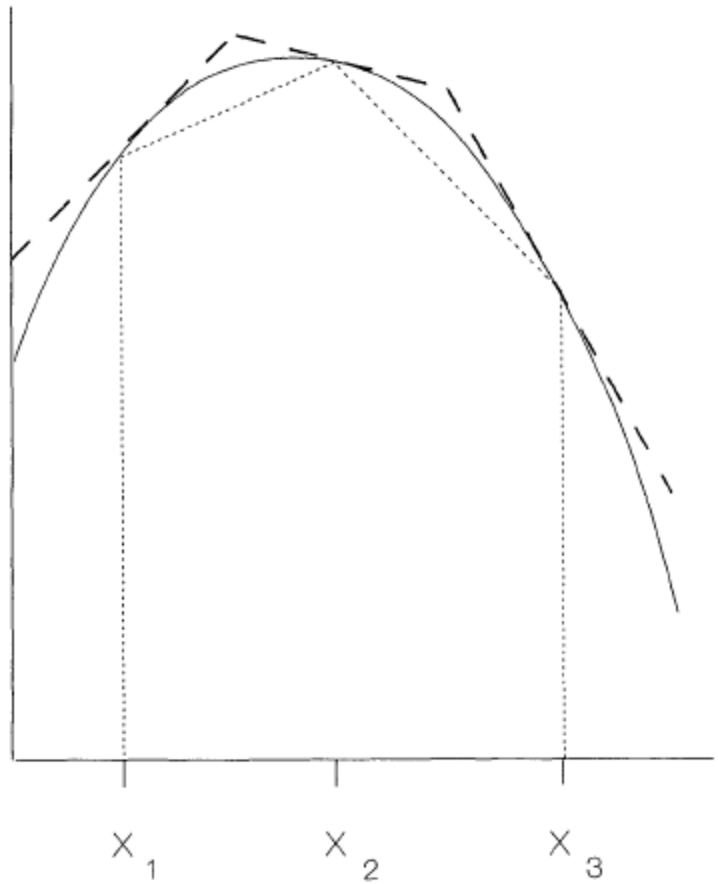
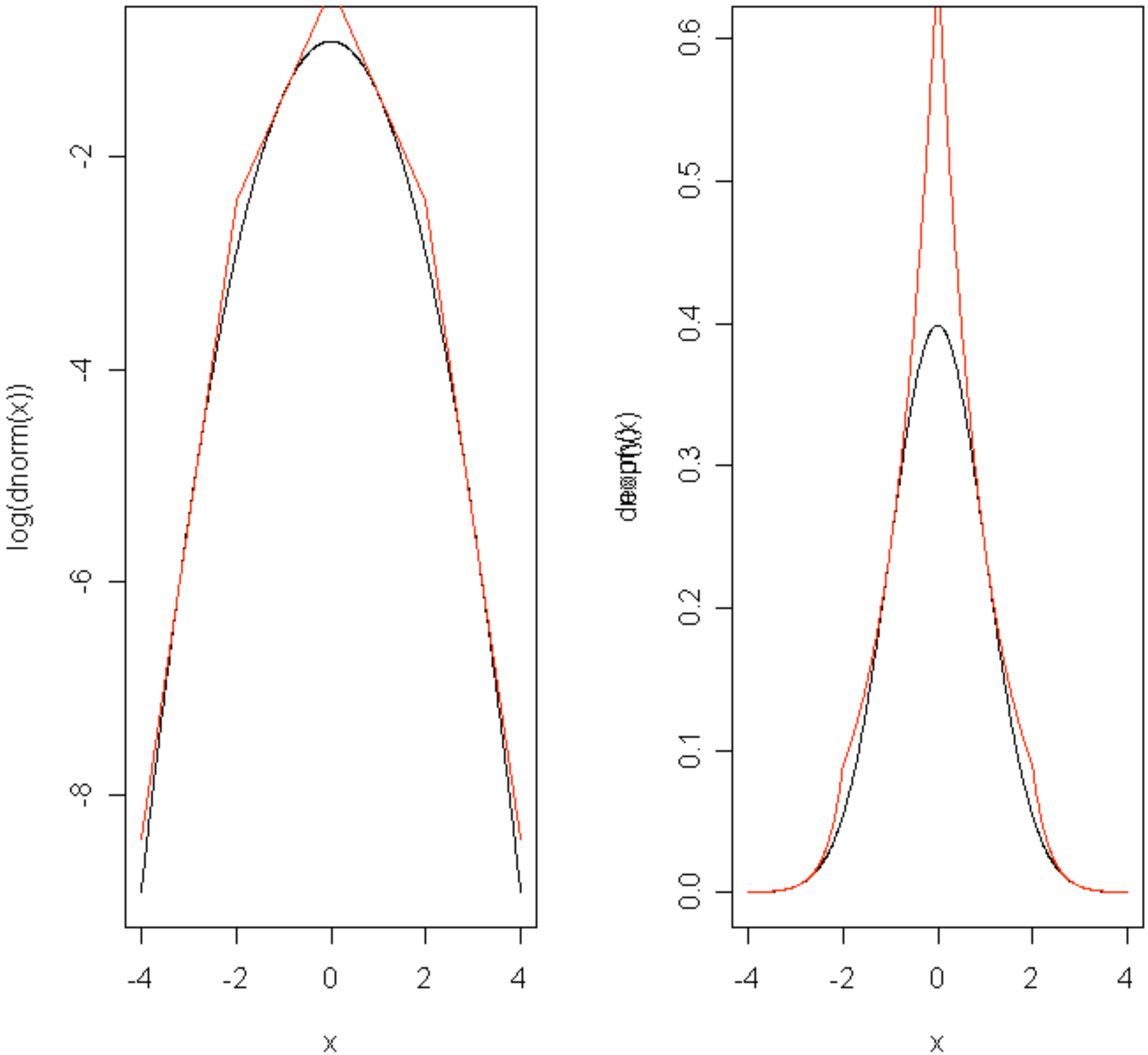
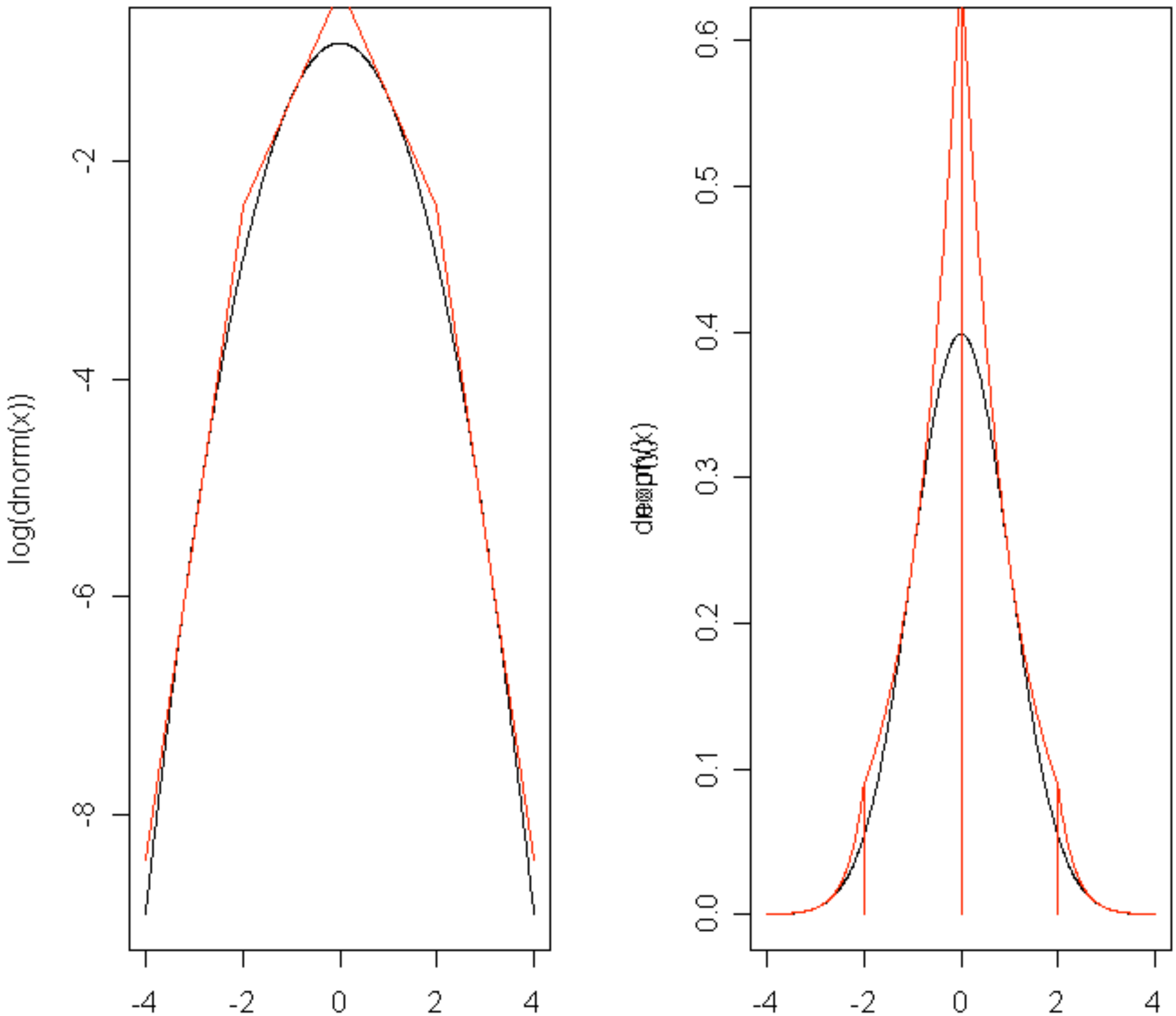


Fig. 1. A concave log-density $h(x)$, bounded on the left, showing upper and lower hulls based on three abscissae (x_1, x_2, x_3): ———, $h(x)$; — — —, upper hull; ········, lower hull


log(dnorm(x))
-2
-4
-6
-8
-4
-2
0
2
4
x
0.0
0.1
0.2
0.3
0.4
0.5
0.6
-4
-2
0
2
4
x


log(dnorm(x))
-2
-4
-6
-8
-4
-2
0
2
4
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-4
-2
0
2
4

2.2.1. *Initialization step*

Initialize the abscissae in T_k. If D is unbounded on the left then choose x_1 such that $h'(x_1) > 0$. If D is unbounded on the right then choose x_k such that $h'(x_k) < 0$. Having defined k starting abscissae, calculate the functions $u_k(x)$, $s_k(x)$ and $l_k(x)$

$$s_k(x) = \exp u_k(x) \Big/ \int_D \exp u_k(x')\, \mathrm{d}x'$$

2.2.2. *Sampling step*

Sample a value x^* from $s_k(x)$ and sample a value w independently from the uniform(0, 1) distribution. Perform the following squeezing test: if

$$w \leqslant \exp\{l_k(x^*) - u_k(x^*)\}$$

then accept x^*. Otherwise evaluate $h(x^*)$ and $h'(x^*)$ and perform the following rejection test: if

$$w \leqslant \exp\{h(x^*) - u_k(x^*)\}$$

then accept x^*; otherwise reject x^*.

2.2.3. *Updating step*

If $h(x^*)$ and $h'(x^*)$ were evaluated at the sampling step, include x^* in T_k to form T_{k+1}; relabel the elements of T_{k+1} in ascending order; construct the functions $u_{k+1}(x)$, $s_{k+1}(x)$ and $l_{k+1}(x)$ from equations (2), (3) and (4) respectively on the basis of T_{k+1}; increment k. Return to the sampling step if n points have not yet been accepted.

TABLE 1

Evaluations of $h(x)$ required to sample one point from the standard normal density, using adaptive rejection sampling, for various starting abscissae†

Starting abscissae x_1	x_2	*Mean number‡ of evaluations of $h(x)$*	*Maximum number‡ of evaluations of $h(x)$*
−0.5	0.5	3.1	7
−1.0	1.0	2.8	6
−2.0	2.0	3.3	6
−5.0	5.0	4.4	8
−10.0	10.0	5.1	8
−9.0	1.0	4.3	8
−8.0	2.0	4.4	7
−7.0	3.0	4.5	8
−6.0	4.0	4.4	8

†1000 simulations.

‡Including evaluations at the starting abscissae x_1 and x_2.

TABLE 2

Log-concavity for common probability density functions $f(x)$ †

$f(x)$	*Parameters*	*log $f(x)$ concave with respect to*	*log $f(x)$ not concave with respect to*
Normal	Mean μ, variance σ^2	x, μ, $1/\sigma$, $\log \sigma$	σ
Log-normal	Location μ, scale σ^2	$\log x$, μ, $1/\sigma$, $\log \sigma$	x, σ
Exponential	Rate λ	x, $\log x$, λ	
Gamma	Index r, rate λ	$\log x$, x (if $r \geqslant 1$), λ, r	x (if $r < 1$)
Beta	Shape a, b	logit(x), x (if a, $b \geqslant 1$), a, b	
Double exponential	Location α, scale β	x, α, $1/\beta$, $\log \beta$	β
Weibull	Shape b, scale a	$\log x$, x (if $b \geqslant 1$), a, b	
Logistic	Location α, scale β	x, α, $1/\beta$	β
Pareto	Shape θ, bound x_0	$\log x$, $1/x$ (if $\theta \geqslant 1$), x_0, $\log x_0$, θ	x
Gumbel or extreme value	Location α, scale β	x, α, $1/\beta$	β
t	Degrees of freedom k		x
F	Degrees of freedom m, n	$\log x$	x
χ^2	Degrees of freedom k	$\log x$, x (if $k \geqslant 2$), k	x (if $k < 2$)
Bernoulli	Proportion p	p, logit(p)	
Binomial	Proportion p, index r	p, logit(p)	
Poisson	Rate λ	λ	
Geometric	Proportion p	p, logit(p)	
Negative binomial	Proportion p, index r	p, logit(p)	

Technical Report No. 2005, Department of Statistics, University of Toronto

Slice Sampling

Radford M. Neal
Department of Statistics and Department of Computer Science
University of Toronto, Toronto, Ontario, Canada
http://www.cs.utoronto.ca/~radford/
radford@stat.utoronto.ca

29 August 2000

3 The idea of slice sampling

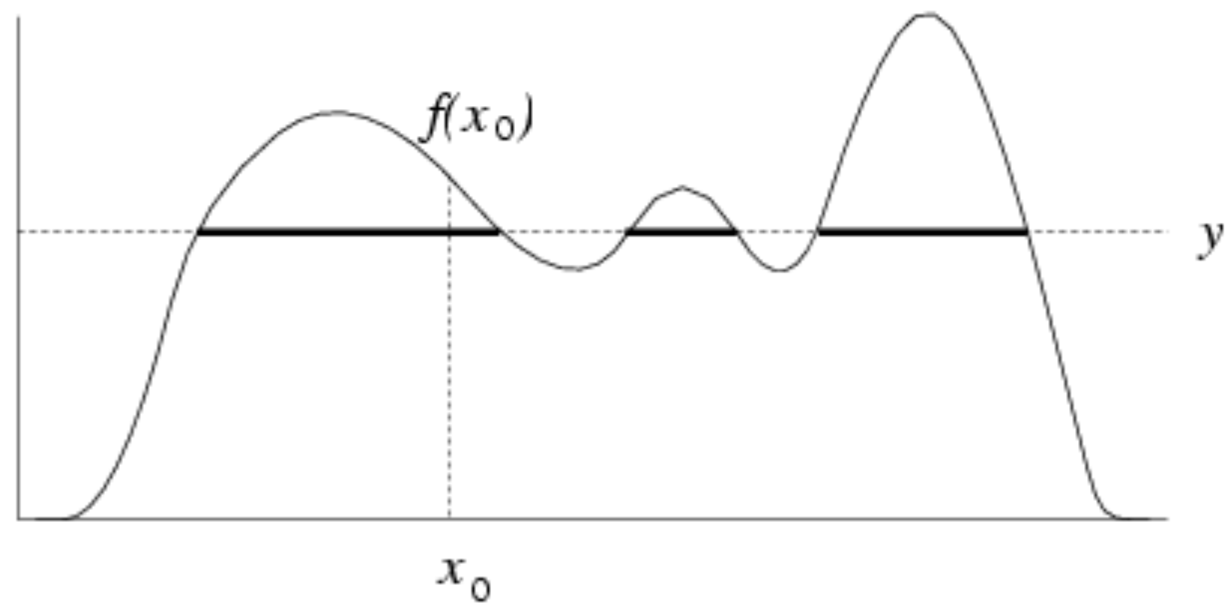
Suppose we wish to sample from a distribution for a variable, x, taking values in some subset of $\Re^n$, with density function proportional to some function $f(x)$. We can do this by sampling uniformly from the $n+1$ dimensional region that lies under the plot of $f(x)$. This idea can be formalized by introducing an auxiliary real variable, y, and defining a joint distribution over x and y that is uniform over the region $U = \{\,(x, y) \;:\; 0 < y < f(x)\,\}$ below the curve or surface defined by $f(x)$. That is, the joint density for (x, y) is

$$p(x, y) \;\;=\;\; \begin{cases} 1/Z & \text{if } 0 < y < f(x) \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $Z = \int f(x)\,dx$. The marginal density for x is then

$$p(x) \;\;=\;\; \int_0^{f(x)} (1/Z)\,dy \;\;=\;\; f(x)\,/\,Z \tag{2}$$

as desired. To sample for x, we can sample jointly for (x, y), and then ignore y.



Simple slice sampling via Gibbs…

We sample alternately from the conditional distribution for y given the current x — which is uniform over the interval $(0, f(x))$ — and from the conditional distribution for x given the current y — which is uniform over the region $S = \{x : y < f(x)\}$,

Methods for simulating Normal Random Variables

1. Sum of 12 uniforms (approximate algorithm)

$$X = \left(\sum_{i=1}^{12} u_i - 6\right), \qquad u_i \sim unif\,(0,1)$$

$$E[X] = \sum_{i=1}^{12} E[u_i] - 6 = 12 \cdot \tfrac{1}{2} - 6 = 0$$

$$V[X] = \sum_{i=1}^{12} V[u_i] = 12 \cdot \tfrac{1}{2} = 1$$

and X is approximately normal by Central Limit Theorem. This is a fairly crude approximation but may be adequate for some applications.

2. Rejection Sampling

First note that if $f_Z(z) = \frac{1}{\sqrt{2\pi}} e^{-z^2/2}$ (i.e. $Z \sim N(0, 1)$)

Then letting $x = |z|$

$$f_X(x) = \frac{2}{\sqrt{2\pi}} e^{-x^2/2}, \quad 0 < x < \infty$$

($F_X(x) = P(X \le x) = P(-x \le z \le +x) = F_Z(x) - F_Z(-x) \Rightarrow f_X(x) = f_Z(x) + f_Z(x)$)

Try candidate distribution $g_X(x) = e^{-x}$ (i.e. exp(1))

Then $\frac{f(x)}{g(x)} = \frac{2}{\sqrt{2\pi}} e^{x-\frac{x^2}{2}} = \frac{2}{\sqrt{2\pi}} e^{-\frac{1}{2}+x-\frac{x^2}{2}} \cdot e^{\frac{1}{2}} = \sqrt{\frac{2e}{\pi} e^{-\frac{1}{2}(x-1)^2}} \le \sqrt{\frac{2e}{\pi}} \cong 1.3155$

($e^{-\frac{1}{2}(x-1)^2} \le 1$ since if $e^{-\frac{1}{2}(x'-1)^2} > 1 \Rightarrow -\frac{1}{2}(x'-1)^2 > 0$ which is impossible)

So, the procedure is:

1. generate a realization from $g_X(x) = e^{-x}$ (Inverse transform)
2. generate $u \sim unif(0,1)$
3. if $u \leq e^{-\frac{1}{2}(x-1)^2}$ then accept *x*. Else go to 1.
4. generate $u \sim unif(0,1)$. If $u < 0.5$ $x \leftarrow -x$.

Note:

This is quite efficient requiring an average of 1.32 iterations per acceptance.

3. Box-Muller

Let $X, Y \sim N(0, 1)$, X, Y independent

Consider a transformation to polar coordinates:

$$R^2 = X^2 + Y^2$$

$$\theta = \tan^{-1}\left(\frac{Y}{X}\right)$$

To get the joint distribution of R^2 and θ need Jacobian of the transformation

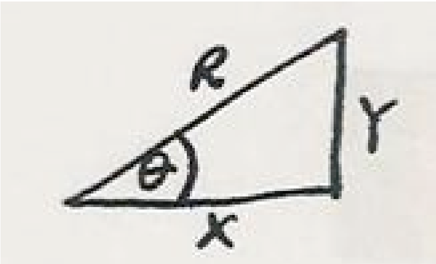
$$J = \begin{vmatrix} \frac{\partial d}{\partial x} & \frac{\partial d}{\partial y} \\ \frac{\partial \theta}{\partial x} & \frac{\partial \theta}{\partial y} \end{vmatrix} = \begin{vmatrix} 2x & 2y \\ \frac{1}{1+\frac{y^2}{x^2}}\left(-\frac{y}{x^2}\right) & \frac{1}{1+\frac{y^2}{x^2}}\left(\frac{1}{x}\right) \end{vmatrix} = 2$$

Since $f_{X,Y}(x,y) = \frac{1}{\sqrt{2\pi}} e^{-(x^2+y^2)/2}$

$$f_{R^2,\theta}(d,\theta) = \frac{1}{\sqrt{2\pi}} e^{-d/2} \cdot \frac{1}{2} = \frac{1}{2} e^{-d/2} \cdot \frac{1}{2\pi} \quad \text{for } 0 < d < \infty \text{ and } 0 < \theta < 2\pi$$

So R^2 and θ are independent. Furthermore $R^2 \sim \exp(1/2)$, $\theta \sim unif(0, 2\pi)$.

So proceed by generating R^2 and θ and then setting $X = R\cos\theta, Y = R\sin\theta$



resulting in two independent standard normal random variables.