Higher Order Envelope Random Variate Generators

Michael Evans and Tim Swartz
University of Toronto and Simon Fraser University

Abstract

Recent developments, Gilks and Wild (1992), Hoermann (1995), Evans and Swartz (1997a), have lead to algorithms for generating from distributions that have black-box characteristics in the sense that these place minimal requirements on a distribution for implementation. All of these developments were based on the construction of linear envelopes for a density or a simple transformation of the density. In this paper we generalize the approach to polynomial envelopes and show that this leads to some distinct advantages. Also we address several difficult generating problems.

1 Introduction

Suppose that we have a distribution on R with density proportional to f and that we wish to generate a sample from this distribution. From a naive point of view this is a simple problem, for if we can evaluate the distribution function

$$F(x) = \frac{\int_{-\infty}^{x} f(z)\,dz}{\int_{-\infty}^{\infty} f(z)\,dz}$$

and its inverse F^{-1}, we can then generate p from a uniform distribution on $(0,1)$ and return $X = F^{-1}(p)$. This is called the *inversion algorithm.* While inversion works well with certain special distributions it is unsatisfactory in general because F and F^{-1} are typically difficult to evaluate and the resulting algorithm is very inefficient. It is this inefficiency that leads to the consideration of alternative algorithms such as the rejection and the ratio of uniforms algorithms; see Devroye (1986) for a description of these and other algorithms and for an in-depth discussion of many of the issues surrounding random variable generation.

In Gilks and Wild (1992) an adaptive rejection algorithm was introduced for log-concave densities; i.e. when $\ln f$ is concave on its domain. Basically a piecewise linear upper envelope u and a piecewise linear lower envelope l are constructed for $\ln f$. We then have that $\exp(l(x)) \leq f(x) \leq \exp(u(x))$ and it is easy to generate from the density g proportional to $\exp(u(x))$ via inversion. This gives a rejection algorithm for f based on generating $X \sim g$, generating $p \sim U(0,1)$ and then accepting X whenever $f(X) \geq p \exp(u(X))$. If X is not accepted then we repeat this process until a value is accepted. An added efficiency step is to first see if $\exp(l(X)) \geq p \exp(u(X))$ as then we can accept X without having to evaluate f which in some cases can be much more costly. This is called *squeezing*. This paper also introduces the notion of adaptive construction of the envelopes as new envelopes are constructed, that better approximate f, each time there is a rejection.

In Hoermann (1995) this algorithm is generalized to a wider class of transformations T and the notion of T-concavity is introduced; i.e. f is *T-concave* whenever $T \circ f$ is concave on its domain. A still wider class of transformations is introduced in Evans and Swartz (1997a) and it is shown that it is not necessary for a density to be T-concave in the sense of the above definition. As described in that paper the essential ingredients for successful implementation of this methodology are transformations T_L and T_R so that the density is inevitably T_L-concave (or T_L-convex) in the left tail and inevitably T_R-concave (or T_R-concave) in the right tail and further that the inflection points be available of the possibly transformed density on the remaining part of the domain; i.e. between the points where we determine the tails to begin. Many of the standard distributions can be handled by these techniques in the sense that with little work highly efficient algorithms are produced. In each of these papers, attention is restricted to linear envelopes for the transformed density.

In Evans and Swartz (1997b) it was noted that the technique for constructing upper and lower envelopes for f could be applied to any function; i.e. not just densities, and since the envelopes are easy to integrate exactly, this yielded approximations to integrals of f with exact error bounds. These integral approximations are relatively inefficient, however, if high accuracy is required as their rate of convergence under compounding is quadratic. It is shown in that paper, however, that it is possible to easily construct piecewise polynomial upper and lower envelopes for f and that these lead to integral approximations with rates of convergence equal to any desired order. The higher order *envelope rules* developed there are shown to be practically useful for approximating integrals.

A natural question then is whether or not the piecewise polynomial envelopes can be used in the random variate generation problem and, if so, are they useful? That is the topic considered in this paper. With some qualifications both of these questions are answered affirmatively. In sections 2 and 3 we provide a detailed presentation of the algorithm which includes extensions and refinements over what has been considered previously. In particular in section 3 we demonstrate a general method for generating useful transformations to handle tails. In section 4 we present examples demonstrating the utility of the methodology and in section 5 we summarize.

Historical precedents for the envelope approach to generating from distributions can be found in Devroye (1986), Marsaglia and Tsang (1989) and Zaman (1991). Also see Hoermann and Derflinger (1996) for an application of this approach to generating from discrete distributions.

2 The Algorithm for the Center

We first consider densities with compact support (x_l, x_r) and subsequently discuss how to generate from the left and right tails. As indicated in section 1 treatment of the tails is basically a question of choosing an appropriate transformation $T : [0, \infty) \to R$ so that $T \circ f$ is concave or convex. Typically we will take x_l and x_r far out in the respective tails so that we don't generate often from the tails. We assume throughout this section that, for any functions referred to, all relevant derivatives exist.

We proceed now to the construction of the polynomial envelopes for f. Suppose that the $n{-}th$ derivative $f^{(n)}$ has no inflection points in (x_l, x_r); i.e. $f^{(n)}$ is either concave or convex in the interval. If $f^{(n)}$ has inflection points in (x_l, x_r) then we subdivide (x_l, x_r) into subintervals, with end-points given by the inflection points, and apply the envelope construction techniques within each subinterval. Therefore a necessary ingredient of our methodology is that the inflection points can be relatively easily obtained. Actually, as our examples particularly demonstrate, the need to obtain inflection points can often be entirely avoided and such situations often correspond to the simplest and most practically useful implementations of the method.

The basic idea behind the construction of the envelopes is dependent on the following simple fact derived from the Fundamental Theorem of Calculus.

Lemma 1. If $g'(x) \leq h'(x)$ for every $x \in (x_l, x_r)$ then $g(x) \leq h(x) - h(x_l) + g(x_l)$ for every $x \in (x_l, x_r)$.

Now suppose that $f^{(n)}$ is concave in (x_l, x_r). We then have the following result which gives upper and lower polynomial envelopes for f on (x_l, x_r).

Lemma 2. If $f^{(n)}$ is concave on (x_l, x_r) then for every $x \in (x_l, x_r)$,

$$l(x) = \sum_{k=0}^{n} \frac{f^{(k)}(x_l)}{k!}(x - x_l)^k + \frac{f^{(n)}(x_r) - f^{(n)}(x_l)}{x_r - x_l}\frac{(x - x_l)^{n+1}}{(n+1)!}$$

$$\leq f(x) \leq u(x) = \sum_{k=0}^{n+1} \frac{f^{(k)}(x_l)}{k!}(x - x_l)^k. \tag{2.1}$$

Proof: Because $f^{(n)}$ is concave on (x_l, x_r) we have that the chord

$$l^{(n)}(x) = f^{(n)}(x_l) + \frac{f^{(n)}(x_r) - f^{(n)}(x_l)}{x_r - x_l}(x - x_l)$$

and the tangent

$$u^{(n)}(x) = f^{(n)}(x_l) + f^{(n+1)}(x_l)(x - x_l)$$

satisfy $l^{(n)}(x) \leq f^{(n)}(x) \leq u^{(n)}(x)$ on (x_l, x_r). Then repeatedly applying Lemma 1 to both sides of this inequality we obtain the result.

If $f^{(n)}$ is convex on (x_l, x_r) then the expressions for $l(x)$ and $u(x)$ in (2.1) are reversed. Notice that because f is nonnegative we can improve the lower envelope as a bound on f by replacing l by $\max(0, l)$. We will assume that this has been done whenever this is convenient.

As f is nonnegative this implies that $u(x)$ is nonnegative and is thus proportional to a density function on (x_l, x_r). Further the distribution function is given by

$$U(x) = \frac{\int_{x_l}^{x} u(z)\,dz}{\int_{x_l}^{x_r} u(z)\,dz}$$

and this is a polynomial whose coefficients are easily and exactly calculated. A direct method of generating from U is to generate $p \sim U(0,1)$ and then solve $U(X) = p$ for X. Note that because U is necessarily strictly increasing in (x_l, x_r) there is a unique such X and this is the unique root of the polynomial $U(x) - p$ of degree $n + 2$, lying in this interval. Therefore this root can be calculated exactly using polynomial root-finding algorithms. In fact when $n = 0$ or $n = 1$ we can use the well-known formulas for the roots of a quadratic or cubic respectively. Recall that the case $n = 0$ corresponds to constructing linear envelopes to the density as discussed in Evans and Swartz (1997a). Actually, as we illustrate in section 4, the polynomial root-finding algorithms, while guaranteed to find all the roots of a polynomial, are relatively inefficient for this purpose, because they compute all the roots. It is much better is to use a general root-finder like the secant method.

Once we have X from the distribution given by U then we apply the rejection step; i.e. generate an additional, independent $p \sim U(0,1)$ and accept X when $f(X) \geq pu(X)$. Of course we can precede the comparison with a squeezing step; namely accepting X if $l(X) \geq pu(X)$. The rejection step leads to an adaptive algorithm. For if $f(X) < pu(X)$ we then put $x_1 = X$ and construct new envelopes on (x_l, x_1) and (x_1, x_r) which are combined to give new upper and lower envelopes l and u for f on (x_l, x_r). We also compute $p_1 = \int_{x_l}^{x_1} u(z)\,dz / \int_{x_l}^{x_r} u(z)\,dz$ and $p_2 = \int_{x_1}^{x_r} u(z)\,dz / \int_{x_l}^{x_r} u(z)\,dz$. A new candidate X is then generated from the density proportional to u by first generating $i \in \{1,2\}$, using the discrete distribution $\{p_1, p_2\}$, and then generating X from u conditioned to the first or second subinterval in the partition $((x_l, x_1), (x_1, x_r))$ according to the value of i. We continue modifying the envelopes; and thus further subdividing (x_l, x_r), until we get an acceptance. We note that the distribution with density proportional to the upper envelope function is a discrete mixture of continuous components with non-overlapping supports. To determine the mixture component we use the aliasing method, see Devroye (1986), which only requires the generation of two uniforms and a comparison. The expensive part of this step is the set-up and this must be done each time we adapt. Accordingly it makes sense to stop adapting when the upper and lower envelopes are providing accurate approximations to f. As discussed in Evans and Swartz (1997a) a good stopping rule to determine this is to stop adapting when the ratio

$$\frac{\int_{x_l}^{x_r} l(z)\,dz}{\int_{x_l}^{x_r} u(z)\,dz}$$

is sufficiently close to 1. Notice that both integrals in this ratio can be easily and exactly evaluated.

When $f^{(n)}$ has $k > 0$ inflection points, then the domain is split into $k+1$ regions. The algorithm proceeds as above by constructing piecewise linear envelopes on each region and by defining a mixture distribution over all regions.

The above algorithm leaves open the question of what is a suitable choice of n? A natural criterion to assess this is efficiency of computation; namely which choice of n leads to the fewest rejection steps or, perhaps more importantly, the fastest mean computation time. However, as we will see in our discussion of the examples in section 4, the choice $n = 0$ frequently leads to a perfectly satisfactory algorithm. The real virtue of the higher order polynomial approach is that there are often simpler ways to compute the polynomial envelopes than directly applying the methods just described to f. For example, suppose that f can be factored as $f(x) = g(x)h(x)$ where $g \geq 0, h \geq 0$ and we have polynomial, perhaps of degree 1, envelopes $l_g \leq g \leq u_g$ and $l_h \leq h \leq u_h$ for these functions. We then have higher order polynomial envelopes $l_g l_h \leq f \leq u_g u_h$ for f. Application of techniques similar to this can often allow us to entirely avoid the computation of derivatives of f and also the need to calculate inflection points of such derivatives. Several such techniques are presented in section 4.

3 The Algorithm for the Tails

We now discuss how to generate from the tails $(-\infty, x_l)$ and (x_r, ∞) and how we go about choosing the points x_l and x_r. We restrict our discussion to the right tail but note that the treatment is similar for the left tail. Also we note that the transformation used may differ in the two tails; e.g. see the discussion of the F distribution in Evans and Swartz (1997a).

Suppose then that we have a transformation T and constants α and β satisfying

(i) $T : (0, \sup\{f\}) \to R$ is smooth and monotone
(ii) T' and T^{-1} are easy to compute
(iii) the anti-derivative of $T^{-1}(\alpha + \beta x)$ is easy to compute for $x \in (x_r, \infty)$ and is integrable on (x_r, ∞)
(iv) it is easy to generate from the distribution with density proportional to $T^{-1}(\alpha + \beta x)$ on (x_r, ∞).

For example, if $T(f) = \ln(f)$ then all of these conditions are satisfied provided that $\beta < 0$. Also if we define T_p for $p \in (-1, 0)$ by $T_p(f) = f^p$ then T_p is smooth and decreasing. Further $T_p^{-1}(x) = x^{1/p}$ for $x > 0$ and, provided that α and β are chosen so that $\alpha + \beta x \geq 0$ on (x_r, ∞), then

$$\int T_p^{-1}(\alpha + \beta x)\, dx = \frac{1}{\beta} \frac{p}{p+1} (\alpha + \beta x)^{\frac{p+1}{p}}.$$

Therefore $T_p^{-1}(\alpha + \beta x)$ is proportional to a density on (x_r, ∞) and since the inverse cdf is easily obtained, it is straightforward to generate from this distribution via inversion. We discuss the conditions (i)-(iv) further below.

Now suppose that T satisfies these conditions and that x_r can be chosen so that f is T-concave on (x_r, ∞) when T is increasing or T-convex on (x_r, ∞) when T is decreasing. Then for $x \in (x_r, \infty)$ we have that

$$T(f(x)) \leq T(f(x_r)) + T'(f(x_r)) f'(x_r)(x - x_r)$$

when f is T-concave and

$$T(f(x)) \geq T(f(x_r)) + T'(f(x_r))f'(x_r)(x - x_r)$$

when f is T-convex. Then taking the T-inverse of both of these inequalities we obtain

$$f(x) \leq T^{-1}\left\{T(f(x_r)) + T'(f(x_r))f'(x_r)(x - x_r)\right\}$$

for $x \in (x_r, \infty)$. Note that $T^{-1}\{T(f(x_r)) + T'(f(x_r))f'(x_r)(x - x_r)\}$ serves as an upper envelope for the right-tail. This in turn leads to the tail bound

$$\int_{x_r}^{\infty} f(x)\,dx \leq \int_{x_r}^{\infty} T^{-1}\left\{T(f(x_r)) + T'(f(x_r))f'(x_r)(x - x_r)\right\}\,dx \quad (3.1)$$

and by condition (iii) the right-hand side of (3.1) can be evaluated in closed form. As noted in Evans and Swartz (1997b) when $T = \ln$ and f is the standard normal density then (3.1) is the Mills ratio inequality and so we can think of (3.1) as a generalization of this and we refer to it by this name hereafter.

Further, from condition (iv), we can easily generate from the density proportional to $T^{-1}(\alpha + \beta x)$ where $\alpha = T(f(x_r)) - T'(f(x_r))f'(x_r)x_r$ and $\beta = T'(f(x_r))f'(x_r)$. The density $T^{-1}(\alpha + \beta x)$ serves as the upper envelope in the rejection algorithm whenever we have to generate from the tails.

The question remains concerning the choice of x_r. In general, we want to choose values so that we spend most of our time using the more efficient center part of the generator. It should be noted, however, that envelopes of fixed degree in the center may not be as good if the tails are chosen too extreme. When f is normalized then (3.1) can be used to choose x_r by ensuring that the right-hand side is suitably small. When f is not normalized, and there are applications where this occurs, then we also need to construct a lower envelope for the center. The right-hand side of (3.1) divided by the integral of the lower envelope for the center gives an upper bound on the tail probability

$$\frac{\int_{x_r}^{\infty} f(x)\,dx}{\int_{-\infty}^{\infty} f(x)\,dx}$$

and can be used to select x_r.

The conditions that we have placed on T are a little different than those stated in Evans and Swartz (1997a). This is because in that paper we always took $n = 0$ and also considered transformations for the center of the distribution. Further we required T' to be homogeneous. In this paper we have avoided transformations in the center since the piecewise envelopes do not lead to simple variate generation when $n \geq 1$. For example, if we take f to be the normal density with $n = 1$ and apply the logarithm transformation in the center, then we would not be able to integrate the upper envelope in closed form as we do with the polynomial form, and we would have to make calls to a normal distribution function routine. Overall, we think that the transformation is necessary to handle the tails, but less often in the center, as the methods of section 2 are satisfactory there. Evans and Swartz (1997a) provide some examples where transforming the center is useful; e.g. the Student(λ) distribution is $T_{-1/(\lambda+1)}$-convex throughout its entire domain and hence we have a very simple adaptive generator for $n = 0$.

A general method for constructing transformations T, useful for handling tails, can be provided. For example, suppose that g is a density from which we can easily generate variates and suppose that g' and g^{-1} can be easily computed. Then define $T^{-1}(f) = g(f)$. For example, letting T equal the logarithm transformation corresponds to the Exponential(1) distribution and the power transformation T_p corresponds to the density proportional to $x^{1/p}$ on (a, ∞) for some $a > 0$. Note that we must have $p \in (-1, 0)$ whenever the support of f contains an infinite interval but other choices are useful too. For example, if f is the Beta$(1/2, 1/2)$ density then f is T_{-2}-convex and note that we definitely need to handle the tails of f in constructing a generator for this distribution even though its support is compact.

New transformations and generating algorithms are obtained from these considerations as well. For suppose that f is bounded and we are willing to generate from the half-normal distribution for the tails. As this is in effect equivalent to generating from a normal and ignoring the sign, there are excellent algorithms available for this. The transformation $T(f) = \sqrt{-2\ln(f/c)}$ where $c > \sup(f)$, corresponds to this density. For example, the density on $(0, \infty)$ proportional to $\exp\left(-x^{1/2}\right)$ is T-concave with respect to this transformation with $c = 1$ and note that it is not log-concave. As a second example, we can use the half-logistic density, which can easily be generated from by inversion, and this leads to the transformation

$$T(f) = 2\cosh^{-1}\left(\sqrt{c/f}\right) = 2\ln\left(\sqrt{c/f} + \sqrt{c/f - 1}\right)$$

where $c > \sup(f)$. Clearly there are many other densities that could be used to generate transformations. In section 3 we only make use of the logarithm transformation but see Evans and Swartz (1997a,b) for other meaningful contexts where this is not adequate.

4 Examples

4.1 The normal distribution

In the first example, we generate from a possibly truncated standard normal distribution. There exist various excellent normal generators and ours is not intended as a replacement although we remark that it does have the virtue of handling truncated normals as well. Rather we use this as a demonstration of applying the methodology in a standard context. We note further that many of the algorithms used to generate from the normal require specialized knowledge of properties of the distribution and were developed after considerable study. This is not the case with the methods that we employ which are essentially black-box. In fact while the developments of section 2 seem to require knowledge of derivatives and inflection points we show that in this example even these requirements can be avoided.

We denote the standard normal density by $\phi(x) = (2\pi)^{-1/2} \exp\left(-x^2/2\right)$. Then $\phi^{(n)}(x) = (-1)^n h_n(x)\phi(x)$ where $h_n(x)$ is the Hermite polynomial of degree n associated with ϕ. These polynomials are easy to compute as they satisfy the recursion relation $h_{n+1}(x) = xh_n(x) - nh_{n-1}(x)$ with $h_0(x) = 1$, $h_1(x) = x$. Also the inflection points of $\phi^{(n)}(x)$ are given by the roots of h_{n+2} and these are readily available in many numerical analysis packages. We can then proceed as in sections 2 and 3 to construct a generator for the $N(0,1)$ distribution. Note that when $T = \ln$ and $x > 0$, (3.1) becomes

$$\int_x^\infty \phi(z)\, dz \le \frac{1}{\sqrt{2\pi}} \frac{e^{-\frac{x^2}{2}}}{x} \tag{4.1}$$

and this is used to determine (x_l, x_r).

We also consider a method for constructing envelopes for ϕ that does not require the evaluation of derivatives or inflection points. This is based on constructing envelopes for the function e^{-x} using the methods of section 2 and then substituting $x^2/2$ for x to obtain envelopes for $\sqrt{2\pi}\phi(x)$. Note that $\left(e^{-x}\right)^{(n)} = (-1)^n e^{-x}$ and so the $n-th$ derivative is concave or convex on an interval (a,b) depending on whether n is odd or even respectively; in particular we do not need to compute inflection points. This leads to the following degree $n+1$ polynomial envelopes for e^{-x} on (a,b); namely upper envelope

$$u_n^{(a,b)}(x) = \begin{cases} e^{-a}\sum_{k=0}^{n+1} \frac{(-1)^k}{k!}(x-a)^k & n \ odd \\ e^{-a}\sum_{k=0}^{n} \frac{(-1)^k}{k!}(x-a)^k + \frac{(-1)^n}{(n+1)!}\frac{e^{-b}-e^{-a}}{b-a}(x-a)^{n+1} & n \ even \end{cases}$$

and lower envelope

$$l_n^{(a,b)}(x) = \begin{cases} e^{-a}\sum_{k=0}^{n}\frac{(-1)^k}{k!}(x-a)^k + \frac{(-1)^n}{(n+1)!}\frac{e^{-b}-e^{-a}}{b-a}(x-a)^{n+1} & n \text{ } odd \\ e^{-a}\sum_{k=0}^{n+1}\frac{(-1)^k}{k!}(x-a)^k & n \text{ } even. \end{cases}$$

Now suppose that we want polynomial envelopes to ϕ on the interval (c,d) where $c \geq 0$ or $d \leq 0$. Then it is immediate that $l_n(x) = l_n^{(a,b)}(x^2/2)$ and $u_n(x) = u_n^{(a,b)}(x^2/2)$ give polynomial envelopes of degree $2n+2$ for $\sqrt{2\pi}\phi(x)$ on (c,d) where $(a,b) = \left(c^2/2, d^2/2\right)$ when $c \geq 0$ and $(a,b) = \left(d^2/2, c^2/2\right)$ when $d \leq 0$. The integrals of these envelopes are also polynomials. Therefore taking linear envelopes for the exponential function results in quadratic envelopes for ϕ and piecewise cubic distribution functions. Although the function $(e^{-x})^{(n)}$ has no change in concavity, we break the real line into $(-\infty,0)$ and $[0,\infty)$ to conveniently handle the cases $c \geq 0$ and $d \leq 0$.

We implemented both of these algorithms with various choices for the degree of the polynomial envelopes. To assess the efficiency of the algorithms we computed the CPU time to return 10^5 values. We refer to the algorithms based on computing a derivative of ϕ and its inflection points collectively as the Derivative algorithm and that based on using the envelopes to e^{-x} as the Exponential algorithm. The value n refers to the degree of the derivative used for the center, m refers to the initial number of equal length subintervals within each concavity region and the acceptance rate refers to the proportion of generated values from the upper envelope that are accepted. This acceptance rate increases with better approximating upper envelopes. Also we considered two different scenarios. In the first one we asked for 10^5 values from the generator in a single call, so that the adaptive steps continued as we generated, and in the second one we called the generator 10^5 times. For all of the algorithms we took $(x_l, x_r) = (-5, 5)$.

We observe from Table 1 that there is considerable overhead in the setup time of our generator as it takes significantly longer in multiple calls than in a single call. The best algorithm is the Derivative algorithm based on $n = 0$ but the Exponential algorithm based on $n = 0$ compares favorably and note that it uses quadratic envelopes for ϕ. A virtue of the Exponential algorithm is that it is easier to implement. Typically, as we increase n or m the envelopes better approximate ϕ, as is reflected by the acceptance rates, but this is offset by an increase in computation time.

The fastest IMSL algorithm for the standard normal; namely *drnnoa,* took about 1.3 second to generate 10^5 standard normal variates based on a single call. Therefore it would appear that none of the algorithms we have considered are competitive with IMSL. Still we remark that our algorithms

Algorithm	n	m	Acceptance Rate	CPU (secs)
Derivative (single call)	0	1	0.999	10
Derivative (multiple calls)	0	1	0.556	50
Derivative (multiple calls)	1	1	0.556	68
Derivative (multiple calls)	2	1	0.759	75
Derivative (multiple calls)	0	3	0.902	57
Exponential (single call)	0	1	0.999	14
Exponential (multiple calls)	0	1	0.464	60

Table 1: Results from generating 10^5 standard normal variates where n is the degree of the derivative and m is the number of equi-spaced points within each concavity interval.

have been implemented with no attempt to optimize in contrast to *drnnoa* and the computation times are not prohibitive. In Evans and Swartz (1997a) we constructed a normal generator using these methods with $n = 0$, applying the log transformation to the entire density, subdividing $(x_l, x_r) = (-4, 4)$ into 60 subintervals and turning off the adaptation. In that case the IMSL algorithm was only twice as fast. The major savings in this minimal design is that exact inversion of an exponential function is faster than inversion of a quadratic using a modified secant method. Therefore it is clear that the proposed methods can yield useful algorithms even for well-studied problem such as the normal distribution.

4.2 Rational-normal and rational-beta distributions

We consider distributions on $D \subseteq R$ with densities of the form

$$\begin{aligned} f(x) &= cr(x)\phi\left(\frac{x-\mu}{\sigma}\right) \\ &= c\frac{p(x)}{q(x)}\phi\left(\frac{x-\mu}{\sigma}\right) \\ &= c\frac{\prod_{i=1}^{m_1}(x-\lambda_i)(x-\lambda_i^*)}{\prod_{i=1}^{m_2}(x-\delta_i)(x-\delta_i^*)}\phi\left(\frac{x-\mu}{\sigma}\right) \end{aligned} \tag{4.2}$$

where ϕ denotes the standard normal density function, $\mu \in R, \sigma \in (0, \infty)$, $\lambda_1 \ldots, \lambda_{m_1} \in C, \delta_1, \ldots, \delta_{m_2} \in C \backslash D$, with C denoting the complex numbers,

λ^* denoting the complex conjugate of λ, and

$$c^{-1} = \int_D r(x)\phi\left(\frac{x-\mu}{\sigma}\right) dx.$$

Notice that q is never 0 so that c^{-1} is finite. Recall that we do not need to calculate c to construct the generator and it does not matter whether or not D is a subinterval or all of R. In Evans and Swartz (1997a) generators for polynomial-normal distributions were constructed using linear envelopes; i.e. $m_2 = 0$ and $n = 0$.

Here we consider two approaches to constructing algorithms for the rational-normals. The first approach, which is a direct implementation of the algorithm discussed in the paper, requires the evaluation of the derivatives of $r(x)\phi(x)$. This is rather cumbersome but we illustrate that this can be done. From Leibniz's formula we have that

$$\left(r(x)\phi\left(\frac{x-\mu}{\sigma}\right)\right)^{(k)} = \sum_{i=0}^{k} \binom{k}{i} r^{(i)}(x)\frac{1}{\sigma^{k-i}}\phi^{(k-i)}\left(\frac{x-\mu}{\sigma}\right)$$

$$r^{(i)}(x) = \sum_{j=0}^{i} \binom{i}{j} p^{(j)}(x)\left(\frac{1}{q(x)}\right)^{(i-j)}.$$

Putting $t(x) = 1/x$, we have, from Faà di Bruno's formula (see Knuth (1973)), that

$$\begin{aligned}\left(\frac{1}{q(x)}\right)^{(i-j)} &= (t(q(x)))^{(i-j)} \\ &= \sum_{l=0}^{i-j} t^{(l)}(q(x)) \sum_{l_1+l_2\cdots+l_{i-j}=l,\, l_1+2l_2\cdots+(i-j)l_{i-j}=i-j} \\ &\quad \left\{ \begin{array}{c} \frac{(i-j)!}{[l_1!\cdots l_{i-j}!]\left[(1!)^{l_1}\cdots((i-j)!)^{l_{i-j}}\right]}\times \\ \left(q^{(1)}(x)\right)^{l_1}\cdots\left(q^{(i-j)}(x)\right)^{l_{i-j}} \end{array} \right\}.\end{aligned}$$

Of course $t^{(l)}(x) = (-1)^l l!/x^{l+1}$ and the derivatives of p and q are easily evaluated in closed form, particularly when they are written out in powers of x. Therefore the derivatives of the unnormalized f can be easily evaluated using a symbolic calculation package such as Maple or via a numerical routine. Further

$$(q(x))^{k+1}\frac{\left(r(x)\phi\left(\frac{x-\mu}{\sigma}\right)\right)^{(k)}}{\phi\left(\frac{x-\mu}{\sigma}\right)}$$

is a polynomial with the same real roots in D as $\left(r(x)\phi\left(\frac{x-\mu}{\sigma}\right)\right)^{(k)}$ and so these roots can be easily evaluated using a symbolic calculation package or numerical routine.

To handle the tails of f we need an appropriate transformation T. In the case when we have polynomial-normal distributions then we can take $T=\ln$; i.e. these distributions are log-concave in the tails , as demonstrated in Evans and Swartz (1997a). This fact generalizes to the rational-normals.

Lemma 3. The density f given by (4.2) is log-concave in the tails.
Proof: We have that

$$[\ln(f(x))]^{(2)}=\frac{-1}{\sigma^2}-\sum_{i=1}^{m_1}\left[\frac{1}{(x-\lambda_i)^2}+\frac{1}{(x-\lambda_i^*)^2}\right]+\sum_{i=1}^{m_2}\left[\frac{1}{(x-\delta_i)^2}+\frac{1}{(x-\delta_i^*)^2}\right]$$

and note that the sums in this expression go to 0 as $x\to\infty$ or $x\to-\infty$.

We also consider a simpler method for constructing a generator which does not involve having to compute complicated derivatives. As with the normal in example 1 we construct upper and lower envelopes for e^{-x} on the interval (a,b) and then note that $r(x)$ times these envelopes provides envelopes for $f(x)$. Further these envelopes take the form of rational functions and as such they can be integrated in closed form. The density proportional to the upper envelope can be generated using inversion via a root search. We used Maple to obtain exact expressions for the integrals of the upper and lower envelopes but this step could also be automated.

We illustrate the discussion here via an example with $D=R,\mu=0,\sigma=1,m_1=2,m_2=1,\ \lambda_1=2.0+.1i,\lambda_2=-2+.1i$ and $\delta_1=i$. Suppose then that we want to base our generator on, at most, the second derivative of f; i.e. $n=2$. For this we need to compute all the derivatives of f up to order 5 to determine the concavity structure of $f^{(2)}$ and for the construction of the envelopes. We used Maple to compute the concavity structure of the derivatives but otherwise used numerical routines to evaluate the derivatives and envelopes via the formulas given above. We must also handle the tails via a transformation. Observe that f is log-concave in the tails by Lemma

Algorithm	n	m	Acceptance Rate	CPU (secs)
Derivative (multiple calls)	0	1	0.610	21
Derivative (multiple calls)	1	1	0.614	36
Derivative (multiple calls)	2	1	0.738	119
Exponential (multiple calls)	0	1	0.637	9

Table 2: Results from generating 10^4 rational-normal variates where n is the degree of the derivative and m is the number of equi-spaced points within each concavity interval.

3. So we must determine x_l, x_r so that $g = \ln f$ is concave in $(-\infty, x_L)$ and (x_R, ∞). Notice that $g^{(2)}$ is a rational function so that its roots can be determined exactly. Using Maple we obtained $x_l = -2.67$ and $x_r = 2.67$ although we chose more extreme values $(x_l, x_r) = (-5, 5)$ for the implementation.

In Table 2 we record the results of a simulation based on generating 10^4 values from this distribution. The terminology is the same as that used in Table 1. As many statistical simulations involve generating variates from families of distributions with continuously changing parameters (e.g. Gibbs sampling), we consider multiple calls to our subroutines. As in the normal example, we observe that increasing the value of n improves the envelopes but at the expense of computation time. Reductions in computation time can be made to the Derivative algorithms by writing code for specific values of n that avoid the generality of the Faà di Bruno formula. Although not shown, tinkering with m can also lead to marginal improvements in the generators. However, a main point to be recorded from this example is that black-box generators can been easily constructed for the non-trivial class of rational-normal distributions. It is also worth noting that the best approach is based on the Exponential algorithm. This is truly a higher order algorithm as the envelopes for $n = 0$ involve the product of a rational function and a quadratic.

The same methodology will also work for other distributions where a basic density has been modified by multiplying it by a nonnegative rational function; i.e. density functions of the form

$$f(x) = c\frac{\prod_{i=1}^{m_1}(x-\lambda_i)(x-\lambda_i^*)}{\prod_{i=1}^{m_2}(x-\delta_i)(x-\delta_i^*)}g(x)$$

where g is a nonnegative function appropriately differentiable. For example, ARMA processes have spectral densities of the form, see Brockwell and

Davis (1991),

$$\frac{\sigma^2}{2\pi}\frac{\left|\psi(e^{-i\lambda})\right|^2}{\left|\omega(e^{-i\lambda})\right|^2}$$

where ψ and ω are polynomials and $\lambda \in [-\pi, \pi]$. Making the transformation $\lambda \to x = \cos\lambda$ gives a density of the form a ratio of two nonnegative polynomials in x times $1/\sqrt{1-x^2}$ which is proportional to a rational-Beta (1/2, 1/2) density. It is readily seen that generators for this class of densities can be constructed using the same methods as detailed above for the rational-normal densities. In fact this observation applies to any rational-beta density. It is worth noting here, however, that these densities will not always be log-concave in the tails and we will need to employ the power transformations to handle the tails. It is clear however that there is always such a power transformation to ensure T_p-concavity or T_p-convexity.

4.3 Bivariate normal distributions truncated to rectangles

Applications are quite common where a random vector X is felt to be bivariate normally distributed; i.e. $X \sim N_2(\mu, \Sigma)$, but conditions impose linear constraints on X. Here we consider the problem when these constraints take the form of a rectangle $\prod_{i=1}^{2}(a_i, b_i)$; i.e. X is known to be in this set. A very naive approach to this problem would be to simply generate from the bivariate normal and reject all values that lie outside this rectangle. This will be hopelessly inefficient, however, whenever the rectangle has small probability content. The only other method that we know of is to use Gibbs sampling; i.e. specify $x_{11} \in (a_1, b_1)$ to start, then generate $X_2 = x_{21}$ given $X_1 = x_{11}$ and that $X_2 \in (a_2, b_2)$, then generate $X_1 = x_{12}$ given that $X_2 = x_{21}$ and $X_1 \in (a_1, b_1)$, etc. It can then be shown that as the number of iterations increases, (X_1, X_2) converges in distribution to the correct distribution. Each generation of an X_i is relatively straightforward for the envelope methods as we are just generating from a truncated univariate normal. This algorithm has some disadvantages, however, as it is not exact and it is difficult to tell in a given context how many iterations are necessary until the process has "converged". Moreover when X_1 and X_2 are highly correlated the convergence can be very slow. We construct an exact generator here.

Since we can always translate (X_1, X_2) with no change in the difficulty of generation, we assume hereafter that $\mu = 0$. Also we can always rescale along one of the axes so we assume for convenience that $\sigma_{11} = 1$. The following steps describe a general method for generating from this distribution:

(i) generate $X_1 = x_1$ given that $X_1 \in (a_1, b_1)$
(ii) generate $X_2 = x_2$ given that $X_1 = x_1$ and $X_2 \in (a_2, b_2)$.
Therefore we need to be able to carry out both of these steps.

The joint density of X constrained to the rectangle is

$$g(x_1, x_2) \propto \phi(x_1)\, \phi\left(\frac{x_2 - \mu(x_1)}{\sigma(x_1)}\right)$$

where ϕ denotes the $N(0,1)$ density function, $\mu(x_1) = \sigma_{12} x_1$ and $\sigma^2(x_1) = \sigma_{22} - \sigma_{12}^2$. Note that $\sigma^2(x_1)$ does not depend on x_1. The conditional density of X_2 constrained on (a_2, b_2) satisfies

$$g_2(x_2 | x_1) \propto \phi\left(\frac{x_2 - \mu(x_1)}{\sigma(x_1)}\right).$$

This is simply a truncated normal distribution and we can easily apply the envelope methods to this. The essential difficulty in the problem arises with the marginal density of X_1 as this is given by

$$\begin{aligned} g_1(x_1) &\propto \phi(x_1) \int_{a_2}^{b_2} \phi\left(\frac{x_2 - \mu(x_1)}{\sigma(x_1)}\right) dx_2 \\ &\propto \phi(x_1) \left[\Phi\left(\frac{b_2 - \mu(x_1)}{\sigma(x_1)}\right) - \Phi\left(\frac{a_2 - \mu(x_1)}{\sigma(x_1)}\right)\right] \\ &= \phi(x_1)\, h(x_1; \sigma_{22}, \sigma_{12}, a_2, b_2) \end{aligned}$$

where the density is constrained on (a_1, b_1) and where Φ denotes the standard normal cdf.

It is clearly difficult to find the zeros for $g_1^{(n)}$ and thus construct envelopes for g_1. Notice, however, that it will be easier to do this for $\phi(x_1)$ and $h(x_1; \sigma_{22}, \sigma_{12}, a_2, b_2)$ separately and that the product of piecewise polynomial envelopes for these functions gives piecewise polynomial envelopes for g_1. Piecewise polynomial envelopes for $\phi(x_1)$ are constructed as in Example 1. Now for $n \geq 1$,

$$\begin{aligned} &h^{(n)}(x_1; \sigma_{22}, \sigma_{12}, a_2, b_2) \\ &= (-1)^n \left(\frac{\sigma_{12}}{\sqrt{\sigma_{22} - \sigma_{12}^2}}\right)^n \left[\begin{array}{c} h_{n-1}\left(\frac{b_2 - \mu(x_1)}{\sigma(x_1)}\right) \phi\left(\frac{b_2 - \mu(x_1)}{\sigma(x_1)}\right) - \\ h_{n-1}\left(\frac{a_2 - \mu(x_1)}{\sigma(x_1)}\right) \phi\left(\frac{a_2 - \mu(x_1)}{\sigma(x_1)}\right) \end{array}\right]. \end{aligned}$$

When $a_2 = -\infty$ or $b_2 = \infty$ then the zeros of this function are easily obtained from the roots of Hermite polynomials which are well-known. The problem is more difficult when neither of these situations holds but note that when we have upper and lower envelopes for functions r and s; namely $l_r \leq r \leq u_r$ and $l_s \leq s \leq u_s$ then $l_r - u_s \leq r - s \leq u_r - l_s$. Further if $r \geq s$ then $u_r - l_s \geq 0$ which is necessary if the upper envelope is to be used in the formation of a density function for the rejection step. Observe that this condition holds when we put $r(x_1) = \Phi\left(\frac{b_2 - \mu(x_1)}{\sigma(x_1)}\right)$ and $s(x_1) = \Phi\left(\frac{a_2 - \mu(x_1)}{\sigma(x_1)}\right)$ and the roots of derivatives of these functions are again translated and rescaled roots of Hermite polynomials.

If a_1 and b_1 are finite then the above algorithm constructs a generator for the distribution. Still we might like to obtain upper bounds on the probability contents of tails of g_1 and it is readily seen that crude bounds can be obtained from those for ϕ. In general, however, it would appear that we have to develop an algorithm to generate from the tails of densities that take the form

$$\phi(x)\left[\Phi(c - dx) - \Phi(b - dx)\right]$$

where $b \leq c$. We note, however, that when $a_1 = -\infty$ and $b_1 = \infty$ the problem is avoided because we can instead first generate X_2 from a truncated normal and then generate X_1 from its conditional distribution given X_2 and this is a full normal. If $a_1 = -\infty$ and $b_1 < \infty$ then we can first generate X_2 from a distribution with density proportional to

$$l(x) = \phi(x)\Phi(c - dx)$$

and then generate X_1 from its conditional given X_2 and this is a truncated normal. If $a_1 < -\infty$ and $b_1 = \infty$ then we can first generate X_2 from a distribution with density proportional to

$$r(x) = \phi(x)\left[1 - \Phi(c - dx)\right]$$

and then generate X_1 from its conditional given X_2 and this is a truncated normal.

So we must provide an algorithm for generating from the left tail of a density proportional to $l(x)$ and from the right tail of a density proportional to $r(x)$. To accomplish this we have the following result.

Lemma 4. The functions $l(x)$ and $r(x)$ are log-concave.
Proof: Put $y = c - dx$ and $m(y) = \phi(y)/\Phi(y)$. Then

$$\left[\ln(l(x))\right]^{(2)} = -1 - d^2 m(y)\left[y + m(y)\right].$$

The log-concavity for all $y \geq 0$ follows immediately. The Mill's ratio inequality (4.1), adapted for the left-tail, gives $y + m(y) \geq 0$ for $y < 0$ and thus we have log-concavity everywhere. Also, putting $M(y) = \phi(y)/(1 - \Phi(y))$ we have that

$$[\ln(r(x))]^{(2)} = -1 - d^2 M(y)\,[-y + M(y)]\,.$$

The log-concavity for all $y \leq 0$ follows immediately. The Mill's ratio inequality (4.1) gives $M(y) \geq y$ for all positive y and therefore we also have log-concavity for all $y > 0$.

To implement our algorithm we need to actually find values x_l and x_r such that l and r are log-concave in $(x_l, x_r)^c$. By Lemma 4 we see that we can take these to be any values. A sensible approach is to determine these values based on the Mill's ratio inequality for ϕ.

These considerations lead immediately to an adaptive rejection generator. We illustrate the algorithm in a particular problem; namely $\sigma_{11} = 1$, $\sigma_{22} = 4$, $\sigma_{12} = -1$, $\prod_{i=1}^{2} (a_i, b_i) = (2, 3) \times (2, 3)$ and we take $n = 0$. Here, the generation of 1000 variates using the naive approach based on the IMSL routine drnnor took 119 seconds of CPU time whereas our approach took only 1 second! Of course, the comparison will be less (more) extreme as the rectangle is closer to (further from) the bulk of the probability mass of the bivariate normal. We note that the methods discussed in the earlier papers referenced in the introduction could not successfully handle this generating problem and so this problem clearly demonstrates the value of the methods we have discussed here.

This approach can be generalized to deal with higher dimensional multivariate normals truncated to rectangles and in fact to convex polytopes. Given the additional complexity entailed, however, we defer discussion of this to another paper.

5 Conclusions

We have developed a class of techniques for constructing adaptive rejection samplers for a broad class of distributions. At a maximum these "black-box" algorithms require an appropriate transformation to handle the tails and require expressions for the derivatives, together with the roots of some of the derivatives. Quite often, however, the implementation can be simplified by using various techniques to directly compute the polynomial envelopes without evaluating derivatives. This latter aspect is probably where the higher order algorithms will find their greatest practical value. While the case $n = 0$ usually provides a perfectly satisfactory algorithm, implementation of higher order polynomial envelopes can often be simpler as shown in Examples 4.1 and 4.2. As shown in Example 4.3 the polynomial approach really is required to handle some generating problems. We have also expanded considerably on the development of appropriate transformations to handle the tails of distributions.

References

Brockwell, P.J. and Davis, R.A. (1991). Time Series: Theory and Methods. Second Edition. Springer-Verlag, New York.

Devroye, L. (1986). Non-Uniform Random Variable Generation. Springer-Verlag, New York.

Evans, M. and Swartz, T. (1997a). Random variable generation using concavity properties of transformed densities. To appear in J. Comp. Graph. Stat.

Evans, M. and Swartz, T. (1997b). An algorithm for the approximation of integrals with exact error bounds. Submitted.

Gilks, W. and Wild, (1992). Adaptive rejection sampling for Gibbs sampling. Applied Statistics, 41, 337-348.

Hoermann, W. (1995). A rejection technique for sampling from T-concave distributions. ACM Trans. Math. Softw., 21, No. 2, 182-193.

Hoermann, W. and Derflinger, G. (1996). Rejection-inversion to generate variates from monotone discrete distributions. ACM Trans. On Modeling and Computer Simulation, Vol. 6, No. 3, 169-184.

Knuth, D.E. (1973). The Art of Computer Programming. Volume 1, Fundamental Algorithms. Addison-Wesley, Reading, MA.

Marsaglia, G. and Tsang, W.W. (1984). A fast easily implemented method for sampling from decreasing or symmetric unimodal density functions. SIAM J. Sci. Stat. Comp., Vol.5, No. 2, 349-359.

Zaman, A. (1991). Generating random numbers from a unimodal density by cutting corners. Unpublished manuscript.