Random Variable Generation Using Concavity Properties of Transformed Densities

M. Evans* and T. Swartz*
U. of Toronto and Simon Fraser U.

Abstract

Algorithms are developed for constructing random variable generators for families of densities. The generators depend on the concavity structure of a transformation of the density The resulting algorithms are rejection algorithms and the methods of this paper are concerned with constructing good rejection algorithms for general densities.

Keywords: random variate generator, T-concavity and T-convexity, points of inflection, adaptive rejection sampling.

1 Introduction

Good algorithms for generating from univariate distributions are a necessary part of many applications where approximations to integrals or expectations are required. For a wide class of commonly used distributions there exist excellent algorithms and for many non-standard distributions there are classes of tools that can be applied to construct good algorithms; see for example, Devroye (1986). But this is not always the case. In many situations an algorithm can be constructed by sheer brute force inversion; i.e. tabulate the distribution function at many points, but this is inelegant and rarely results in a satisfactory solution. By this we mean that the time taken to generate many independent realizations can be considerable. Further, often we want an algorithm that can generate from a family of distributions and the specific distribution in the family cannot be prespecified; e.g. it may

*Partially supported by a grant from the Natural Sciences and Engineering Research Council of Canada.

depend on data that varies from application to application, or the distribution may be changing dynamically as the simulation progresses, as in Gibbs sampling. In such contexts the brute force algorithm is not feasible.

In spite of the extensive amount of development in this area the authors have still encountered numerous situations where there is no obvious algorithm available beyond brute force inversion. We describe some of these in section 5. The purpose of this paper is to describe a general black-box algorithm that is capable of handling a wide variety of situations. In essence the user is required to input a minimal amount of information about densities belonging to a family of distributions and then the algorithm constructs an efficient generator. This is not a universal black-box as certain information is required to be available, or at least easily computed, and sometimes this is not the case. Basically, the information required is at most the first three derivatives, and the roots of the first and second derivatives, of some simple transformations of the density. This information is readily available for many univariate densities. A strong point of the algorithm described here is that excellent algorithms for specific distributions can be easily constructed and this does not demand deep insight into the properties of the distributions or great amounts of development time. The computer does all the work.

The algorithm we develop can be thought of as a generalization in several ways of the adaptive rejection algorithm developed in Gilks and Wild (1992). We will refer to this algorithm hereafter as the Gilks-Wild algorithm. The Gilks-Wild algorithm is a black-box algorithm for distributions with log-concave densities. For example, the $N(0,1)$, Gamma(α) for $\alpha > 1$ and Beta(a, b) for $a, b \geq 1$ all have log-concave densities. Further log-concavity is maintained under location-scale transformations and truncation. On the other hand the Student(λ) and the F(a, b) densities are not log-concave. While there are simple good algorithms for generating from the full Student or F distributions this is not the case for truncations. The algorithm we describe here leads to new, good algorithms for the full distributions and also easily handles truncations.

In Gilks, Best and Tan (1993) a Markov chain algorithm is developed that combines the Gilks-Wild algorithm with the Metropolis algorithm to give an approximate generator for a general univariate density. In addition to only being approximate this algorithm also suffers from the existence of correlation between realizations. The algorithms developed here are exact and generate independent realizations. The Gilks-Wild algorithm is an adaptive rejection algorithm and this is particularly suitable in a number of applications of Gibbs sampling when the full conditional densities are log-concave. The algorithms developed here are adaptive and do not require the log-concave restriction on densities.

In section 2 we discuss T-concavity and specific examples of transformations T. In section 3 we indicate how these concepts are used to construct generators. In section 4 we consider the design of good generators and in section 5 we present examples. Conclusions are given in section 6.

Other authors have developed black-box approaches to constructing generators for distributions. For example, Marsaglia and Tsang (1984), Devroye (1986), Zaman (1991) and Hörmann (1995) all contain developments relevant to this problem. In particular Hörmann (1995) first introduced the idea of T-concavity, on which much of the development in this paper is based. We have extended this development in a number of significant ways. We show that a much wider class of transformations is useful and introduce T-convexity. Densities are not required to be unimodal or bounded. Further the entire density is not required to be T-concave or T-convex provided that the inflection points of the transformed density are available. Finally adaptation is introduced together with a useful stopping rule. In section 5 we provide examples of generators that could not have been developed using previous results.

2 T-Concavity

The Gilks-Wild algorithm is based on the log-concavity of a density f; i.e. the function $\ln \circ f$ is concave. In fact there is no reason to restrict just to the logarithm transformation and there is no reason to restrict to concavity. With appropriate restrictions on transformations $T : (0, \infty) \to R$, similar algorithms can be constructed.

Accordingly we say $f : D \to R$ is *T-concave* , where D is a convex subset of R, if $T \circ f$ is concave. If $T \circ f$ is smooth then f is T-concave if and only if $(T \circ f)'' = (T'' \circ f)(f')^2 + (T' \circ f)f'' \leq 0$. Further we say that f is T-*convex* if $-(T \circ f\ )$ is concave. We call a convex subset C of D a *domain of T-concavity* of f if $T \circ f$ is concave or convex there. We will restrict our discussion hereafter to functions f with a finite partition $\{D_1, \ldots, D_m\}$ of D by domains of T-concavity. For such an f there is a coarsest such partition, which we call the *T-partition* of f, and note that this can be constructed by finding the inflection points of $T \circ f$. The T-partition together with the concavity of $T \circ f$ on each partition element will be referred to as the *T-concavity structure* of f. It is clear that T-concavity or T-convexity is preserved under location-scale transformations and truncations.

We recall some elementary facts about inflection points for smooth functions g defined on an open interval. First x is a point of inflection for g if and only if $g''(x) = 0$ and $g'''(x) \neq 0$. Further, if x is a point of inflection of g and $g'''(x) > 0$ then g changes from concave to convex as we proceed from left to right through x. Similarly g changes from convex to concave if $g'''(x) < 0$. The inflection points of $T \circ f$, perhaps for several T transformations, is typically the information needed to construct a generator using the methods of this paper. For many distributions and transformations this information is readily available.

For the transformations T considered here the T-concavity structure of a function does not change under positive multiples of f. A sufficient condition to ensure this is that T' be homogeneous of degree $\mu \in R$. For T' homogeneous of degree μ implies that T'' is homogeneous of degree $\mu - 1$ and then the above expression for $(T \circ f)''$ shows that the sign of this quantity does not change under positive multiples of f. In such a case our algorithm does not require that the density be normalized. This is practically significant, as often determining a norming constant can be a substantial computation. Further, it allows for great convenience in our development as we will ignore norming constants. As such, when referring to a density f, hereafter, we will only require that it be nonnegative and integrable.

As might be imagined an arbitrary T does not suffice for the construction of good, or even feasible, generators. For convenience we list what seem to be necessary characteristics. The necessity of these will become apparent when we present the algorithm.

1. $T : (0, \infty) \to R$ is smooth, monotone and T' is homogeneous of some degree,
2. T and its derivatives and T^{-1} are easy to compute,
3. the anti-derivative of $T^{-1}\ (\alpha + \beta x)$ is easy to compute for $x \in D$ and is integrable on D; i.e. $T^{-1}\ (\alpha + \beta x)$ is a density on D
4. it is easy to generate from the distribution with density $T^{-1}\ (\alpha + \beta x)$ via inversion.

We now present some examples of transformations that satisfy items 1-4.

2.1 Logarithm transformation

If we take $T = \ln$ then T is smooth and increasing, T' is homogeneous of degree -1, $T^{-1}(x) = \exp(x)$ and $\int T^{-1}\ (\alpha + \beta x)\, dx = \frac{1}{\beta} \exp(\alpha + \beta x)$.

Therefore T^{-1} $(\alpha + \beta x)$ is a density on (a, b) whenever $a, b \in R$, a density on $(-\infty, b)$ when $\beta > 0$ and is a density on (a, ∞) when $\beta < 0$. Further the inverse cdf of any of these distributions is easily obtained using the log function so that generating using inversion is easy.

2.2 Power transformations

We define T_p for $p \neq 0$ by $T_p(f) = f^p$. Then T_p is smooth, increasing when $p > 0$, decreasing when $p < 0$ and T' is homogeneous of degree $p - 1$. 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 D, then

$$\int T_p^{-1}(\alpha + \beta x)\, dx = \begin{cases} \frac{1}{\beta} \frac{p}{p+1} (\alpha + \beta x)^{\frac{p+1}{p}} & if \quad p \neq 0, -1 \\ \frac{1}{\beta} \ln(\alpha + \beta x) & if \quad p = -1 \end{cases} .$$

¿From this we see that, provided that $\alpha + \beta x \geq 0$ on the interval in question, $T_p^{-1}(\alpha + \beta x)$ is a density on (a, b) for every $p \neq 0$ and is a density on $(-\infty, b)$ or (a, ∞) whenever $p \in (-1, 0)$. In all of these cases the inverse cdf is easily obtained and so it easy to generate from these distributions via inversion. As we will see the requirement that $\alpha + \beta x \geq 0$ on an interval, is typically easy to satisfy as part of the algorithm.

We note that another family of transformations given by $T_p^*(f) = (f^p - 1)/p$ for $p \neq 0$ and $T_0^*(f) = \ln(f)$ includes the log and power transformations in a continuous family. There seems to be no apparent advantage to this family, however, and it ignores the fundamental difference in the restriction placed on $\alpha + \beta x$ between the log and power transformations. Also while $T_p \circ f$ and $T_p^* \circ f$ have the same concavity structure they lead to different generators.

3 The Algorithm

We restrict ourselves initially to the situation where $D = [a, b]$ is a bounded interval and suppose that we have chosen T, satisfying 1-4, so that f is T-concave or T-convex on D and $T \circ f$ is smooth. We note, however, that in general we can use different T transformations on different parts of the support of f. Further we suppose that we have chosen points $a \leq x_1 < \ldots < x_m \leq b$. In section 4 we will discuss how to choose these points.

Now let $t_i(x) = (T \circ f)(x_i) + (T \circ f)'(x_i)(x - x_i)$ be the equation of the tangent line to $T \circ f$ at x_i. Let $z_i \in (x_i, x_{i+1})$ be a point satisfying $t_i(z_i) = t_{i+1}(z_i)$ for $i = 1, \ldots, m-1$ and put $z_0 = a$, $z_m = b$. Note that the T-concavity or T-convexity ensures that z_i exists and if $(T \circ f)'(x_i) \neq (T \circ f)'(x_{i+1})$ then it is unique. If $(T \circ f)'(x_i) = (T \circ f)'(x_{i+1})$ then $T \circ f = t_i$ on (x_i, x_{i+1}). In this case we will see that there is no benefit to having both x_i and x_{i+1} in the partition and so we can delete one. Henceforth we will assume that this has been done and then

$$z_i = \frac{[(T \circ f)'(x_i)x_i - (T \circ f)'(x_{i+1})x_{i+1}] - [(T \circ f)(x_i) - (T \circ f)(x_{i+1})]}{[(T \circ f)'(x_i) - (T \circ f)'(x_{i+1})]}.$$

Further let $c_i(x) = (T \circ f)(z_i) + [(T \circ f)(z_i) - (T \circ f)(z_{i-1})](x - z_i)/(z_i - z_{i-1})$ be the equation of the secant from $(z_{i-1}, (T \circ f)(z_{i-1}))$ to $(z_i, (T \circ f)(z_i))$ for $i = 1, \ldots, m$.

Now define the *upper envelope function* by

$$u(x) = \begin{cases} t_i(x) & \text{if } z_{i-1} \leq x \leq z_i, \quad \begin{array}{c} T \circ f \text{ concave,} \\ T \text{ increasing} \end{array} \quad \text{or} \quad \begin{array}{c} T \circ f \text{ convex ,} \\ T \text{ decreasing} \end{array} \\ c_i(x) & \text{if } z_{i-1} \leq x \leq z_i, \quad \begin{array}{c} T \circ f \text{ concave,} \\ T \text{ decreasing} \end{array} \quad \text{or} \quad \begin{array}{c} T \circ f \text{ convex ,} \\ T \text{ increasing} \end{array} \end{cases}$$

and the *lower envelope function by*

$$l(x) = \begin{cases} c_i(x) & \text{if } z_{i-1} \leq x \leq z_i, \quad \begin{array}{c} T \circ f \text{ concave,} \\ T \text{ increasing} \end{array} \quad \text{or} \quad \begin{array}{c} T \circ f \text{ convex ,} \\ T \text{ decreasing} \end{array} \\ t_i(x) & \text{if } z_{i-1} \leq x \leq z_i, \quad \begin{array}{c} T \circ f \text{ concave,} \\ T \text{ decreasing} \end{array} \quad \text{or} \quad \begin{array}{c} T \circ f \text{ convex,} \\ T \text{ increasing} \end{array} \end{cases}.$$

We then have that $T^{-1}(l(x)) \leq f(x) \leq T^{-1}(u(x))$ for every $x \in (a, b)$ and on (z_{i-1}, z_i), $T^{-1}(u(x)) = T^{-1}(\alpha + \beta x)$ for some α, β. Define the mixture density $g(x) = T^{-1}(u(x)) / \int_a^b T^{-1}(u(z))\, dz = \sum_{i=1}^m p_i g_i(x)$ where $p_i = d_i/(d_1 + \cdots + d_m)$, $d_i = \int_{z_{i-1}}^{z_i} T^{-1}(u(x))\, dx$ and $g_i(x) = T^{-1}(u(x))/d_i$ on $[z_{i-1}, z_i]$ and is equal to 0 otherwise. We can generate from g since it is easy to calculate the p_i and easy to generate from component g_i using inversion. We use the aliasing algorithm; see Devroye (1986), to generate from the discrete distribution $(p_{1,\ldots,}p_m)$. Thus generating from g only requires the generation of 2 uniforms. Then the rejection sampling algorithm for f proceeds by (i) generating $X \sim g$, (ii) generating $V \sim U(0,1)$, (iii) if $f(X) \geq VT^{-1}(u(X))$ then return X else go to (i). In contexts where the computation of $f(X)$ is expensive we can add a squeezer step, between (ii) and (iii), by first testing $T^{-1}(l(X)) \leq VT^{-1}(u(X))$ and returning X if this holds, otherwise carrying out step (iii). In the adaptive version of this algorithm the point X is added to $\{x_1, \ldots, x_m\}$ and a new l and u computed whenever $f(X) < VT^{-1}(u(X))$.

We recall here the requirement that $\alpha + \beta x \geq 0$ whenever T is a power transformation. Consideration of the above shows that this restriction will automatically be satisfied piecewise by u and l whenever we require that $x \in \{x_1, \cdots, x_m\}$ if $(T \circ f)'(x) = 0$.

There are several assumptions associated with the above development. First we assumed that there exists a T such that f is T-concave or T-convex on D. This is clearly not necessarily the case but this problem is easily dealt with when $D = [a, b]$ by using the T-partition of f and constructing u and l piecewise on each element of the partition. More serious are the assumptions of bounded support and of no singularities at the end-points. As we will see in the examples, both of these problems can be dealt with in very general families of densities by making a judicious choice of a T transformation for a tail interval or an interval with a singularity as an end-point. For example, it turns out that when a tail is not log-concave then there is often a power transformation T_p such that the tail is T_p-convex and $p \in (-1, 0)$ so that T_p is decreasing. In particular, infinite intervals are easily handled, in the sense that we can construct a rejection sampler g on the whole interval, whenever f is T-concave on the interval with T increasing or whenever f is T-convex on the interval with T decreasing. In these cases u is defined exactly as in the case of a bounded interval while l must be modified so that the squeezer $T^{-1}(l(x))$ takes the value 0 on infinite intervals; e.g. in the log-concave case $l(x)$ takes the value $-\infty$ on such intervals.

4 Selecting the Points

Given a specific density f it is natural to ask which transformation T should be used, say from amongst those described in section 2. It turns out, however, that a single T is sometimes not sufficient as we will require different transformations for the tails and the central region; e.g. see section 5. In the situation where a single transformation suffices then we would like to choose that T and the points $\{x_1, \ldots, x_m\}$ which maximizes the probability of acceptance; namely

$$P(f(X) \geq VT^{-1}(u(X))) = \frac{\int f(x)\,dx}{\int T^{-1}(u(x))\,dx}.$$

This is not a tractable problem, however, even in very simple contexts. One thing we can say, based on the developments in section 3, is that choosing T so that $T \circ f$ is approximately linear seems appropriate. Accordingly, we will suppose that T has been chosen for a particular interval and consider the choice of the points $\{x_1, \ldots, x_m\}$ in this interval.

While optimal selection of the points may be a reasonable approach for distributions that are used very frequently, in general the following seems like an effective way to proceed as it demands minimal input and design of the algorithm. We start with some initial set $\{x_1, \ldots, x_m\}$ containing at least all the criticial and inflection points of $T \circ f$ and typically it pays to include more than these. For example, if the largest x value is a critical point and the distribution has an infinite right tail then we must include one more point in the right tail else g will not be integrable. A similar consideration arises if the smallest x value is a critical point and the distribution has an infinite left tail. We then let the algorithm run adaptively; i.e. every time an X generated from the rejection sampler is rejected, we add X to $\{x_1, \ldots, x_m\}$ and update the envelopes. We can either let the adaptation run indefinitely or stop after the probability of acceptance is sufficiently close to 1. Given that integrating f may be difficult, it makes sense to instead use the closeness of

$$\alpha_* = \frac{\int T^{-1}(l(x))\,dx}{\int T^{-1}(u(x))\,dx}$$

to 1 as our stopping rule as

$$\frac{\int T^{-1}(l(x))\,dx}{\int T^{-1}(u(x))\,dx} \leq \frac{\int f(x)\,dx}{\int T^{-1}(u(x))\,dx} \leq 1$$

and we expect α_* to be close to 1 for a good generator as well.

We note that the time needed to generate from a particular g is independent of the number of components in the mixture. So provided that storage is not an issue, and it rarely is, we can continue the adaptation for many adaptive steps. The only computational cost is the need to update the envelopes. Adaptation is only carried out, however, when there is a rejection and, otherwise, we are generating values from f.. This only requires that the algorithm retain a memory of its structure between calls if we are generating many values. With a typical application of Gibbs sampling a single value from the distribution is all that is required. Even in this extreme small sample setting the adaptive approach often provides a very efficient generator as has been observed many times with the Gilks-Wild algorithm.

For illustrative purposes we consider applying this algorithm to the $N(0,1)$ distribution. Of course there are many excellent algorithms for this distribution but many of these require a considerable amount of design work in contrast to the approach taken here. Note that the $N(0,1)$ density is log-concave so that the tails are easily handled and we can use the Gilks-Wild algorithm. Therefore we put $T = \ln$ and start with $\{x_1, \ldots, x_{61}\}$ obtained by dividing each of the intervals [-4,-1], [-1,0], [0,1] and [1,4] into 15 subintervals and then let the x_i equal the endpoints. We obtained $\alpha_* =$.9974 as an initial lower bound on the probability of acceptance. Other than including 0 as an initial point, as it is a critical point of $\ln \circ f$, the x_i were chosen in what seemed a reasonable way to span the support of the distribution. We compared the performance of this algorithm with the IMSL algorithm drnnoa, which is based on developments in Kinderman and Ramage (1976). The accompanying documentation cites this as the fastest normal generator available in that package of routines. Based on 10^6 calls to our routine and to drnnoa we found that the Gilks-Wild algorithm was 51% slower. Also the final value of α_* was .9998. Given that the total generation time for our routine was 132 seconds on a Sparc workstation this difference can be viewed as not very substantial. Note that this includes the computation time needed to construct the initial envelopes and we did not turn off the adaptation when α_* achieved some desired value. We stress, however, that we are not advocating our approach for distributions that already have good generators and present this only as an example of how effective the approach can be with relatively little effort. One distinct advantage of the approach taken here, however, is that it is easy to modify the algorithm for truncations by simply truncating the envelopes.

5 Examples

In this section we consider a number of examples where the techniques of this paper have been found to be useful as good generators for these distributions are not known. Many of the algebraic computations in this section leading to exact expressions were carried out using Maple and this avoided a considerable amount of tedious and error-prone algebra. The basic idea in these examples is to use the T-concavity structure of the density to subdivide the support of the density into a finite set of subintervals. On each of these intervals the methods of section 3 can then be applied. The choice of T is not entirely clear but we generally follow the principle of choosing a transformation that handles the tails and then using this T for the entire distribution. In some cases this is not possible as the left and right tail must be treated differently and then we use two transformations. While we describe how to construct algorithms for the entire distribution we recall that the same algorithm also applies to truncations of the distributions.

5.1 Makeham's distribution

The (normalized) density of this distribution is given by

$$f(x) = (a + bc^x)\exp(-ax - \frac{b}{\ln(c)}(c^x - 1))$$

where $b > 0, c > 1, a > -b, x \geq 0$. This distribution has applications in actuarial science. Scollnik (1996) discusses generating from this distribution and notes that it is not always log-concave and so recommends the Markov chain algorithm described in Gilks, Best and Tan (1993). Our methods lead to an exact generator that gives independent realizations.

We have that

$$\begin{aligned} f'(x) &= -\exp\left(-\frac{ax\ln c + bc^x - b}{\ln c}\right)(b^2(c^x)^2 - b(\ln c)c^x + 2abc^x + a^2) \\ f''(x) &= \exp\left(-\frac{ax\ln c + bc^x - b}{\ln c}\right)h(c^x) \end{aligned}$$

with

$$h(y) = b^3y^3 + 3b^2(a - \ln c)y^2 + b\left(3a^2 - 3(\ln c)a + \ln^2 c\right)y + a^3$$

and

$$f'''(x) = -\exp\left(-\frac{ax\ln c + bc^x - b}{\ln c}\right)k(c^x)$$

with

$$\begin{aligned} k(y) &= b^4y^4 + 2b^3(-3\ln c + 2a)y^3 + b^2\left(6a^2 + 7\ln^2 c - 12(\ln c)a\right)y^2 + \\ &\quad b(2a - \ln c)\left(2a^2 - 2(\ln c)a + \ln^2 c\right)y + a^4. \end{aligned}$$

Note that $h(y) \geq 0$ when y is large and this implies that the tail is convex. Roots of $h(y)$ can be solved for symbolically but these expressions are too complicated for coding so it is better to find these numerically. Those roots that are greater than 1 give inflection points for $f(x)$ via the transformation $x = \ln(y)/\ln(c)$. The sign of $k(y)$ at these points determines the concavity structure of f .

We must determine a method for handling the tail. For this put $g(x) = \ln(f(x))$ and then

$$g'(x) = -\left(b^2(c^x)^2 - b(\ln c)c^x + 2abc^x + a^2\right)(a + bc^x)^{-1}$$

$$
\begin{aligned}
g''(x) &= -b\,(\ln c)\left(b^2\,(c^x)^3 + 2ab\,(c^x)^2 - (\ln c)\,ac^x + a^2c^x\right)(a+bc^x)^{-2} \\
g'''(x) &= -b\left(\ln^2 c\right)\left(\begin{array}{c} b^3\,(c^x)^4 + 3b^2a\,(c^x)^3 + b\,(\ln c)\,a\,(c^x)^2 + \\ 3a^2b\,(c^x)^2 + a^3c^x - (\ln c)\,a^2c^x \end{array}\right)(a+bc^x)^{-3}.
\end{aligned}
$$

Then making the transformation to $y = c^x$ the third factor of $g''(x)$ becomes $l(y) = y\left(b^2y^2 + 2aby - (\ln c)\,a + a^2\right)$. The roots of $l(y)$ are 0 and $\frac{1}{b}\left(-a \pm \sqrt{a\ln c}\right)$and the roots greater than 1 give the inflection points of the log-density. Observe that $l(y) \geq 0$ for large y and this implies $g''(x) \leq 0$ for large x so that the tail of f is inevitably log-concave. If we let $m(y)$ denote the third factor of $g'''(x)$, after transforming to y, then the sign of $m(y)$ at these inflection points determines the concavity structure of the log-density.

So to construct a generating algorithm we could determine the concavity structure of f and then determine the log-concavity structure for the tail. We notice, however, that to determine the concavity structure of f is unnecessary as we can use $\ln \circ f$ for the construction of the entire generator. In fact this is simpler because calculating the concavity structure of f requires finding the roots of a cubic polynomial while for $\ln \circ f$ we need only calculate the roots of a quadratic. This is straightforward and gives a black-box algorithm for Makeham's distribution for all parameter values. Notice that whenever $a < 0$ (i.e. the non-zero roots are imaginary) or if the largest root of $l(y)$ is less than 1, then the density is log-concave.

¿From the expression for g' we see that $g'(x) = 0$ for $x > 0$ if and only if the polynomial $b^2y^2 + (-b\ln c + 2ab)\,y + a^2$ has a root greater than 1. The roots of this quadratic are given by $\frac{1}{2b}\left(\ln c - 2a \pm \sqrt{\ln c\,(\ln c - 4a)}\right)$. We then use the transformation $x = \ln(y)/\ln(c)$ on these values and on the values $\frac{1}{b}\left(-a \pm \sqrt{a\ln c}\right)$ to determine the starting set of points for the adaptive algorithm; namely those x values that are greater than 0. The sign of $g'''(x)$ at each of the inflection points determines the change of concavity.

As a specific example consider generating from this distribution with $a = .01, b = .01$ and $c = e$. Then the log-density has a single local maximum at 4.585, no local minima and a single inflection point at 2.197; i.e. in particular the density is not log-concave. To construct the initial envelopes we divided the intervals [0,2.197], [2.197,4.585], [4.585,2(4.585)] into 15 subintervals each and used the end-points for the initial x_i values. The initial value for α_* was .9888 and after generating 10^4 values it was .9979. It took 103 seconds of CPU time to generate 10^6 values. We considered a number of other cases. It consistently proved to be useful to place a value in the right tail of the density and twice the largest critical or inflection point generally proved satisfactory. This is why the point 2(4.585) was used above. For some choices of the parameters there were no critical or inflection points and in such cases we used an approximation to the median of the distribution and twice this value to give the initial intervals that were subdivided as above.

5.2 Polynomial-normal distributions

Let $\phi(x) = \exp(-\frac{1}{2}x^2)$ and $p(x) = \prod_{i=1}^{m}(x - \lambda_i)(x - \bar{\lambda}_i)$, where the λ_i are possibly complex, be a general nonnegative polynomial. Then $f(x) = p(x)\phi(x)$ can be treated as a density on R . Many calculations can be carried out exactly for this class of densities; see Evans and Swartz (1995) for an extensive discussion of this family. As discussed there it is extremely difficult to obtain an efficient generator for this family. We apply the methods of this paper to build a general generator.

We have that the derivatives take the form

$$\begin{aligned}
f'(x) &= (p'(x) - xp(x))\,\phi(x) \\
f''(x) &= \left(p''(x) - p(x) - 2xp'(x) + x^2p(x)\right)\phi(x) \\
&= q(x)\phi(x) \\
f'''(x) &= \left(p'''(x) - 3p'(x) - 2xp''(x) + 2xp(x) + x^2p'(x) - xq(x)\right)\phi(x) \\
&= r(x)\phi(x).
\end{aligned}$$

To determine the points of inflection we must calculate the real roots of $q(x)$ and use $r(x)$ to determine the concavity structure of the density. Note that computing the derivatives of p is simple once we have computed the coefficients of p and these can easily be computed recursively by multiplying one quadratic factor at a time into the product.

It is clear that the tails of f are convex. Putting $g(x) = \ln(f(x)) = \ln(p(x)) - \frac{1}{2}x^2$ we have

$$\begin{aligned}
g'(x) &= (p'(x) - xp(x))\,p^{-1}(x) \\
g''(x) &= \left[(p''(x) - p(x) - xp'(x))p(x) - (p'(x) - xp(x))p'(x)\right]p^{-2}(x) \\
&= t(x)p^{-2}(x) \\
g'''(x) &= \left[t'(x)p^2(x) - 2t(x)p(x)p'(x)\right]p^{-4}(x) \\
&= s(x)p^{-4}(x)
\end{aligned}$$

where

$$t'(x) = (p'''(x) - 2p'(x) - xp''(x))p(x) + (p''(x) - p(x) - xp'(x))p'(x) -$$

$$(p''(x) - p(x) - xp'(x))p'(x) - (p'(x) - xp(x))p'(x)$$

and these are all easily evaluated. For large x we have that $g''(x) \leq 0$ and thus the tails are always inevitably log-concave. So once we have evaluated the roots of $p'(x) - xp(x)$ and $t(x)$ we construct the generator for a polynomial-normal exactly as described for Makeham's distribution.

As an example we consider $p(x) = (x - \lambda_1)(x - \bar{\lambda}_1)(x - \lambda_2)(x - \bar{\lambda}_2)$ where $\lambda_1 = 1 + .5i$, $\lambda_2 = -3 + .5i$. Figure 1 is a plot of $f(x)$ and Figure 2 is a plot of $\ln(f(x))$. We see immediately from Figure 2 that the density is not log-concave and that apparently $\ln(f(x))$ has 3 critical points and 4 points of inflection. This is indeed the case and these points were determined via a simple calculation using Maple. Evaluating $g'''(x)$ at each of the points of inflection confirmed the changes of concavity observed in the plot. Since the largest of these 7 values was a relative maximum we added one further point in the right tail. These 8 values give 8 compact subintervals and each of these were further subdivided into 2 subintervals and the endpoints provided the initial x_i values for the adaptive algorithm. The initial value of α_* was .7714 and after 10^4 generations this became .9954. It required 106 seconds of CPU time to generate 10^6 values.

5.3 Truncated Student(λ) distribution

There are many algorithms for generating from a Student distribution. We note, however, that this statement does not extend to truncated versions of these distributions. We consider generating from the entire Student distribution and then use the truncated envelopes to generate from a truncated Student. The Student(λ) density is $f(x) = \left(\lambda + x^2\right)^{-\frac{1}{2}\lambda - \frac{1}{2}}$ for $\lambda > 0$ and its derivatives are

$$\begin{aligned}
f'(x) &= -(\lambda + 1)\left(\lambda + x^2\right)^{-\frac{1}{2}\lambda - \frac{3}{2}} x \\
f''(x) &= (\lambda + 1)\left(\lambda + x^2\right)^{-\frac{1}{2}\lambda - \frac{5}{2}} \left(\lambda x^2 + 2x^2 - \lambda\right).
\end{aligned}$$

Then letting $h(x) = \lambda x^2 + 2x^2 - \lambda$ we see that the inflection points of the density occur at $x = \pm\sqrt{\lambda / (\lambda + 2)}$ and that the tails are convex. A similar computation shows that the tails are also log-convex so we need a different transformation for the tails.

To handle the tails we consider an appropriate choice of p in $h(x) =$

$(T_p \circ f)(x) = (f(x))^p = (\lambda + x^2)^{-(\frac{1}{2}\lambda - \frac{1}{2})p}$. The derivatives equal

$$
\begin{aligned}
h'(x) &= -px\,(\lambda+1)\left(\lambda + x^2\right)^{(-\frac{1}{2}\lambda-\frac{1}{2})p-1} \\
h''(x) &= p\,(\lambda+1)\left[(p\lambda+p+1)\,x^2-\lambda\right]\left(\lambda+x^2\right)^{(-\frac{1}{2}\lambda-\frac{1}{2})p-2} \\
h'''(x) &= -p\,(\lambda+1)\,(\lambda p+p+2)\,x\left[(p\lambda+p+1)\,x^2-3\lambda\right]\left(\lambda+x^2\right)^{(-\frac{1}{2}\lambda-\frac{1}{2})p-3}.
\end{aligned}
$$

Then from $h''(x)$ we see that the tails of f are inevitably T_p-convex whenever $p \in [-1/(\lambda+1), 0)$ and inevitably T_p-concave for $p \in (-\infty, -1/(\lambda+1)] \cup (0, \infty)$. Noticing that $-1 < -1/(\lambda+1)$ for all $\lambda > 0$ and therefore $T_{-1/(\lambda+1)}$ is decreasing, we see that taking $p = -1/(\lambda+1)$ gives an algorithm for the tails. Moreover, with this choice of p we have that $h''(x) > 0$ for every x and thus the Student(λ) distribution is $T_{-1/(\lambda+1)}$-convex and this makes for a simple generating algorithm. This would appear to be a new algorithm for the Student family and truncations of Students.

As a particular example we consider generating from a Student(.5) distribution truncated to (-1,2). The initial envelopes for this distribution were constructed by truncating the initial envelopes constructed for the full Student(.5) distribution. These in turn were based on dividing the intervals [-4,-1], [-1,0], [0,1], [1,4] into 15 subintervals each and using the endpoints as the initial x_i values. This gave an initial value for α_* of .9991 for the truncated distribution and .6776 for the full distribution. After generating 10^4 values from the truncated distribution we obtained $\alpha_* = .9992$. Similarly after generating 10^4 values from the full distribution we obtained $\alpha_* = .9691$ and the adaptation proved far more useful here. Generating 10^6 values from the truncated and full distributions required 111 and 118 seconds of CPU time respectively.

5.4 Truncated F(a, b) distribution

There are many good ways to generate from the full F distribution but not from truncated versions. An important class of applications of the truncated F distribution arises when stratified sampling is implemented in conjunction with multivariate Student importance sampling; e.g. see Evans and Swartz (1995). The density is given by $f(x) = (x)^{\frac{1}{2}a-1}(b+ax)^{-\frac{1}{2}a-\frac{1}{2}b}$ for $a > 0, b > 0$ and $x > 0$. Note that when $0 < a < 2$ the density has a singularity at 0 and in this case we refer to this as the left tail. A simple computation shows that the tails are convex and log-convex so we need alternative transformations for the tails. We consider then a power transformation $h(x) = (T_p \circ f)(x) = (f(x))^p$ and try to choose p conveniently. We have that

$$\begin{aligned}
h'(x) &= -\frac{1}{2}p\,(2ax + abx - ab + 2b)\, x^{\frac{p}{2}a-p-1}\,(b+ax)^{-\frac{p}{2}a-\frac{p}{2}b-1} \\
h''(x) &= \frac{1}{4}pk(x)x^{\frac{p}{2}a-p-2}\,(b+ax)^{-\frac{p}{2}a-\frac{p}{2}b-2} \\
h'''(x) &= -\frac{1}{8}pl(x)x^{\frac{p}{2}a-p-3}\,(b+ax)^{-\frac{p}{2}a-\frac{p}{2}b-3}
\end{aligned}$$

where

$$\begin{aligned}
k(x) &= a^2\,(2+b)\,(bp+2p+2)\,x^2 - 2ab\,(bp+2p+2)\,(a-2)\,x + \\
&\quad b^2\,(a-2)\,(ap-2p-2)
\end{aligned}$$

and

$$\begin{aligned}
l(x) &= -\frac{1}{8}a^3\,(b+2)\,(pb+2p+4)\,(pb+2p+2)\,x^3 + \\
&\quad \frac{3}{8}a^2b\,(pb+2p+4)\,(pb+2p+2)\,(a-2)\,x^2 - \\
&\quad \frac{3}{8}b^2a\,(a-2)\,(pa-2-2p)\,(pb+2p+4)\,x + \\
&\quad \frac{1}{8}b^3\,(a-2)\,(pa-2p-2)\,(pa-2p-4)\,.
\end{aligned}$$

The leading coefficient of $pk(x)$ determines the inevitable concavity of the right-tail and the sign of this quantity is determined by the sign of $(b+2)p^2 + 2p$. Thus the right-tail of f is inevitably T_p-concave when $p \in [-\frac{2}{b+2}\,, 0)$ and inevitably T_p-convex when $p \in (-\infty, -\frac{2}{b+2}] \cup (0, \infty)$. Observe, however, that $-1 < -\frac{2}{b+2}$ and if $p \in (\,-1, -\frac{2}{b+2}]$ then T_p is decreasing and we have a simple algorithm for the right-tail. Moreover when $p = -\frac{2}{b+2}$ and $a \geq 2$ it is immediate that $h''(x) \geq 0$ everywhere and so the $F(a,b)$ distribution is $T_{-2/(b+2)}$-convex and there is a simple algorithm for the entire distribution.

When $0 < a < 2$ the concavity of the left-tail; i.e. when x is close to 0, is inevitably determined by the sign of the constant term in $pk\,(x)$ and the sign of this is determined by the sign of $(a-2)\,p\,(ap-2p-2)$. Therefore in this situation the left-tail is T_p-concave for $p \in [\,\frac{2}{a-2}\,, 0)$ and T_p-convex for $p \in (-\infty, \frac{2}{a-2}] \cup (0, \infty)$. When $p = \frac{2}{a-2}$ then T_p is decreasing and, since the left-tail interval is bounded, this gives a valid generator even though $\frac{2}{a-2} < -1$. Further if $p = \frac{2}{a-2}$ then $pk(x) \propto 4a\,(b+a)\,x\,((2a+ba)\,x - 2ba + 4b)$ and this implies that $h''(x) > 0$ for every $x > 0$. Therefore to construct a generator we use the transformation $T_{2/(a-2)}$ on $(0, c)$ for some $c > 0$. For the interval (c, ∞) we use T_p where $p \in (-1, -\frac{2}{b+2})$ as the density is T_p-convex on this interval and T_p is decreasing. Note that we cannot use $p = -\frac{2}{b+2}$. For a and p in the indicated ranges it is clear that $a - 2 < 0$ and $bp + 2p + 2 < 0$. Thus the largest root of $pk(x)$ is given by

$$\frac{b}{a}\frac{a-2}{b+2}\left(1 - \sqrt{\frac{2\,(a+b)}{(a-2)\,(bp+2p+2)}}\right).$$

Then provided p and c are chosen so that this root is less than or equal to c we can use T_p in (c, ∞). Note that this largest root is a monotone increasing function of $p \in (-1, -\frac{2}{b+2})$ and goes to ∞ as $p \to -\frac{2}{b+2}$ and goes to

$$c_l = \frac{b}{a}\frac{a-2}{2+b}\left(1 - \sqrt{\left(-\frac{2(a+b)}{(a-2)\,b}\right)}\right)$$

as $p \to -1$. Simple manipulations show that $c_l > 0$ for all $a \in (0, 2)$, $b > 0$. So if $c > c_l$ we can find p so that the largest root of $pk(x)$ equals c. This p is given by

$$p(c) = -2\frac{(a^2b + 2a^2)\,c^2 + (-2a^2b + 4ba)\,c - b^2a + 2b^2}{((2a+ba)\,c - ba + 2b)^2}$$

and note that $p(c) \to -\frac{2}{b+2}$ as $c \to \infty$ and $p(c_l\,) = -1$. Thus $c = dc_l$ for any $d > 1$ is appropriate.

6 Conclusions

We have presented a general black-box algorithm for the construction of a random variable generator based on the concavity properties of simple transformations of densities. This generalizes the Gilks-Wild algorithm in several ways. Provided there is minimal information available about the density then the construction of excellent rejection algorithms is easy and automatic. A number of examples have demonstrated the utility of this approach in contexts where finding good generators has proven difficult. Further these examples demonstrate that it is possible to use these methods for families of distributions with little added complexity.

It is easy to see that random variable generation from many of the most common distributions encountered in practice can be handled by these techniques. For example, the Beta(a, b) distribution is not log-concave whenever a or b is less than 1. But it is clear from our development of a generator for the F distribution that these cases can be handled in exactly the same way. Similar considerations apply to the Gamma(a) distribution whenever $a < 1$.

The authors have developed general software to allow these methods to be applied to virtually any distribution for which the information detailed in the paper is available. The software for the examples discussed in this paper is available at:

`http://www.math.sfu.ca/mast/people/faculty/tim/`
`software/software.html`

and this can be suitably modified for other distributions of interest. Additional research can also be done on using other classes of transformations and on the selection of the initial points.

References

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

Evans M. and Swartz, T. (1994) Distribution theory and inference for polynomial-normal density functions. Communications in Statistics, Theory and Methods, 23(4), 1123-1148.

Evans, M. and Swartz, T. Bayesian integration using multivariate Student importance sampling. Computing Science and Statistics, 27, eds. M.M. Meyer and J.L. Rosenberger, Interface Foundation of North America, Inc., Fairfax, Station, VA 22039-7460, p. 456-461.

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

Gilks, W., Best, N.G., and Tan, K.K.C. (1995) Adaptive rejection Metropolis sampling. Applied Statistics, 41, 337-348.

Hörmann, W. (1995) A rejection technique for sampling from T-concave distributions. ACM Transactions on Mathematical Software, Vol. 21, No. 2, 182-193.

Kinderman, A.J. and Ramage, J.G. (1976). Computer generation of normal random variables. Journal of the American Statistical Association, Vol. 71, No. 356, 893-896.

Marsaglia, G. and Tsang, W.W. (1984) A fast easily implemented method for sampling from decreasing or symmetric unimodal density functions. SIAM Journal on Scientific and Statistical Computing, 5, 349-359.

Scollnik, D. (1996) Simulating random variates from Makeham's distribution. Transactions of the Society of Actuaries, Vol. XLVII, 41-69.

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