A Universal Generator for Discrete Log-Concave Distributions

Wolfgang Hörmann

Department of Applied Statistics and Data Processing
Wirtschaftsuniversität Wien

Preprint Series

Preprint 8
October 1993

http://statistik.wu-wien.ac.at/

Version: 11.10.1993 (Accepted for publication in "Computing")

A Universal Generator for Discrete Log-Concave Distributions

Wolfgang Hörmann
Institut f. Statistik, WU-Wien, Augasse 2-6, A-1090 Wien
e-mail: whoer@statrix2.wu-wien.ac.at

Abstract: We give an algorithm that can be used to sample from any discrete log-concave distribution (e.g. the binomial and hypergeometric distributions). It is based on rejection from a discrete dominating distribution that consists of parts of the geometric distribution. The algorithm is uniformly fast for all discrete log-concave distributions and not much slower than algorithms designed for a single distribution.
AMS Subject Classification: 65C10, 68C25.
Key Words: Random number generation, log-concave distributions, rejection method, simulation.

1. Introduction

A discrete distribution on the integers is called log-concave if the probabilities p_k satisfy $p_k^2 \geq p_{k-1}p_{k+1}$ for all k. Most of the classical discrete distributions like the binomial, Poisson, negative binomial and hypergeometric distributions are of this type (the only non log-concave distribution that has a chapter of its own in [9] is the logarithmic series distribution). Among the less often used log-concave distributions contained in [9] are the Pólya-Eggenberger distribution with parameter $s \leq 0$ and the hyper-Poisson distribution with parameters $\lambda > 0$ and $\theta > 0$. For the classical distributions fast methods for generating random numbers are given in the literature. On the other hand a simple black-box algorithm for arbitrary log-concave discrete distributions with given mode is suggested in [5]. The disadvantage of that rejection algorithm lies in the fact that it has an acceptance probability of less than 0.25 for any distribution which makes the algorithm very slow. The aim of this paper is to develop an universal algorithm that is not much slower than the special algorithms for the classical distributions and has a uniformly bounded execution time over the whole class of log-concave distributions.

2. The Method

For continuous log-concave distributions rejection algorithms with piecewise exponential envelopes that touch the density in several points were suggested (see e.g. [4] and [6]). For discrete distributions an envelope with uniform centerparts and exponential tails was used in [10] and in the universal algorithm of [5]. We will apply a discrete dominating distribution consisting of several geometric parts as this leads to higher acceptance probabilities especially if the variance of the distribution is small. Each geometric part touches the desired distribution in two neighbouring points, log-concavity implies that the described envelope dominates the target distribution for all k. We have designed two algorithms: One with an arbitrary number of points of contact which is fast but has a slow set-up (the C-code is available on request), and the simplest and most important special case which will be explained in more detail in this paper. The dominating distribution consists of three parts: a uniform center touching in the mode and two geometric tails. We have $g(k) \geq p_k$ for all integers k and

$$g(k) = \min\left(p_{tlx}e^{al(k-tlx)},\ p_m,\ p_{trx}e^{ar(k-trx)}\right)$$

where tlx denotes the point of contact on the left, trx the point of contact on the right-hand side of the mode m and $al = \log(p_{(tlx+1)}) - \log(p_{tlx})$, $ar = \log(p_{trx}) - \log(p_{(trx-1)})$. Figure 1 compares – as an example – the probabilities of the binomial distribution (thick lines, $n = 100$, $p = 0.2$) with $g(k)$ (thin lines, $tlx = 14$, $trx = 26$). To generate random numbers of the discrete distribution with probabilities proportional to $g(k)$ it is first necessary to compute the largest k

of the left tail (called *bl* in the algorithm below), the smallest k of the right tail (*br*), the mass of the three parts (called *voll*, *volc* and *volr*) and to use decomposition. To generate a sample from a discrete distribution with $p_k = const \cdot e^{(-ak)}$ for $0 \le k \le c$ it is easy to verify that we can use $X = \lfloor -\log(U)/a \rfloor$ where U is uniformly distributed over the interval $(e^{-a(c+1)}, 1)$. This method is used to generate the left and right tail of the dominating distribution. As the evaluation of the probabilities is time consuming for most discrete distributions it is worth-wile to use the exponential functions connecting the points of contact of the envelope as simple squeezes (shown as dashed lines in Figure 1).
The above idea works with many choices of the points of contact on the left-hand side (called tlx) and on the right-hand side (trx). It is enough that the ascent of the logarithmic tangent al in tlx is positive and that ar in trx is negative to get a rejection algorithm with a bounded expected number of iterations to return one random deviate of a fixed distribution. But if we claim that the method can be applied to any discrete log-concave distribution in practice it is important that the expected number of iterations is uniformly bounded over all discrete log-concave distributions. In the sequel we will show that this is the case for simple choices of tlx and trx.
As a first step we will prove the following theorem for continuous distributions (which is implicitly contained in [4] section VII.2.6 by combining Theorem 2.6 (p. 299) with example 2.2 (p. 304)). A continuous log-concave distribution has the property that the logarithm of the density is concave on its support.

Theorem 1: Let $f(x)$, $x \ge 0$ be the density function of a continuous log-concave distribution with mode at 0. Then f is majorized by

$$g(x) = \min\left(f(0), f(x_t)e^{(\log f(x_t))'(x-x_t)}\right)$$

and $\int_0^\infty g(x)\,dx \le t_0 = e/(e-1)$ for $x_t = t_0/f(0)$ which is the optimal choice for x_t.
Proof: To deal with the worst case first we take the density $f_1(x) = e^{-x/t_0}$ in the interval $(0, t_0)$ and 0 elsewhere. It is obvious that $\int_0^\infty g(x)\,dx = t_0$ for $x_t \le t_0/f(0)$ and $\int_0^\infty g(x)\,dx = x_t$ for $x_t \ge t_0/f(0)$, which proves the optimality of the choice.
To complete the proof we denote the intersection between the two parts of g by $x_i = x_t + \log(f(0)/f(x_t))/(\log f(x_t))'$ and get:

$$\int_0^\infty g(x)\,dx = x_i f(0) + f(x_t)\int_{x_i}^\infty e^{(\log f(x_t))'(x-x_t)}\,dx = x_t f(0) + \frac{f(0)}{(\log f(x_t))'}\left(\log\left(\frac{f(0)}{f(x_t)}\right) - 1\right)$$

Now we substitute $x_t = t_0/f(0)$ into the last expression. As it follows from the envelope $g(x)$ in [4] p 288 Theorem 2.1 that $f(t_0/f(0)) \le f(0)/e$ the proof is complete. □

Remark: For the case that the left and the right derivative of $\log(f(x))$ in x_t are different the theorem remains correct. It makes no difference whether the derivative is replaced by the left or by the right derivatives. Therefore it is possible to apply Theorem 1 to discrete distributions:

Theorem 2: Let p_k be the probabilities of a discrete log-concave distribution with mode m. Then the p_k's are dominated by

$$g(k) = \min\left(p_{tlx}e^{al(k-tlx)},\ p_m,\ p_{trx}e^{ar(k-trx)}\right)$$

where $tlx = m - \lceil\frac{e/(e-1)}{p_m}\rceil$, $al = \log(p_{(tlx+1)}) - \log(p_{tlx})$, $trx = m + \lceil\frac{e/(e-1)}{p_m}\rceil$, $ar = \log(p_{trx}) - \log(p_{(trx-1)})$
and we have: $\sum_{k=-\infty}^{\infty} g(k) \le 2e/(e-1) + p_m$.
Proof: For $x \ge m$ we define $f(x) = p_k \exp((\log p_{k+1} - \log p_k)(x-k)), k \le x \le k+1$ for all integers $k \ge m$. Thus $f(k) = p_k$ for the integers right of the mode and $log(f(x))$ consists of straight lines connecting the integer points. It is obvious that the area below $f(x)$ is smaller than

1, that f is log-concave and that $\left(\log(f(m + \frac{e/(e-1)}{f(m)}))\right)' = ar$. Thus the dominating function of Theorem 1 can be applied to $f(x)$ and has the same values as $g(k)$ on the integers. Therefore $\sum_{k=m}^{\infty} g(k) \le e/(e-1) + p_m$.
$\sum_{k=-\infty}^{m} g(k)$ can be estimated by the same argument reflecting k at m.
$\sum_{k=-\infty}^{\infty} g(k) = (\sum_{k=-\infty}^{m} g(k)) + (\sum_{k=m}^{\infty} g(k)) - g(m) \le 2e/(e-1) + p_m$ completes the proof. □

Theorem 2 gives a simple rule to choose the points tlx and trx such that the expected number of iterations is smaller than $3.164 + p_m$ for any discrete log-concave distribution. But the minimax approach is by far not optimal for many standard distributions. In the symmetric case the minimax approach gives $e/(2(e-1))/p_m = 0.791/p_m$ as the optimal distance between the mode and tlx. For the standard normal distribution it is possible to compute the optimal distance (see [4] p. 299 Theorem 2.6) resulting in $tlx = \sqrt{2} = (1/\sqrt{\pi})/f(0) = 0.564/f(0)$. As the classical discrete distributions have the normal distribution as a limiting case we decided to take the optimal value for the normal distribution as the first choice in the algorithm below, but then the expected number of iterations, which is equal to *volcompl*, is not uniformly bounded over all log-concave distributions. Therefore the value of the minimax approach is taken in the case that *volcompl* is larger than $3.164 + p_m$. Thus Theorem 2 implies that the expected number of iterations in Algorithm DLC is below $3.164 + p_m$.

3. The Algorithm

For most discrete log-concave distributions it is much faster to calculate $\log(p_k)$ instead of p_k. Therefore we give the details of the universal algorithm for discrete log-concave distributions on the positive integers with given mode and computable $\log(p_k)$.

Algorithm DLC:

1: [*Set-up*]

1.0 Set $co \leftarrow 0.564$ and go to step 1.2.

1.1 Set $co \leftarrow 1.582$

1.2: [*Compute the points of contact (tlx, trx) and the borders of the support (left, right).*]
Set $m \leftarrow$ mode of the distribution, $logpm \leftarrow \log(p_m)$, $pm \leftarrow \exp(logpm)$, $c \leftarrow \lceil co/pm \rceil$,
$tlx \leftarrow m - c$, $trx \leftarrow m + c$,
$left \leftarrow$ smallest k with $p_k > 0$.
If the right tail of the distribution is infinite set $fright \leftarrow 0$,
else set $fright \leftarrow 1$ and $right \leftarrow$ largest k with $p_k > 0$.

1.3: [*Compute the border to the center-part (bl), the ascent of log(g_k) (al) and of the squeeze (sal) and the sum of the g_k's (voll) for the left tail.*]
If $tlx < left$ set $bl \leftarrow left - 1$, $voll \leftarrow 0$ and go to step 1.4
else set $tly \leftarrow \log(p_{tlx})$, $al \leftarrow \log(p_{tlx+1}) - tly$.
If $al \le 0$ go to step 1.1. else set $bl \leftarrow \lfloor tlx + (logpm - tly)/al - 10^{-10} \rfloor$,
$col \leftarrow \exp(-al * (bl - left + 1)) - 1$, $voll \leftarrow \exp((bl - tlx) * al + tly) * col/(\exp(-al) - 1)$,
$sal = (logpm - tly)/(m - tlx)$.

1.4: [*Analogous to 1.3 for the right tail.*]
If $fright = 1$ and $trx > right$ set $br \leftarrow right + 1$, $volr \leftarrow 0$ and go to step 1.5
else set $try \leftarrow \log(p_{trx})$, $ar \leftarrow try - \log(p_{trx-1})$.
If $ar \ge 0$ go to step 1.1. else set $br \leftarrow \lceil trx + (logpm - try)/ar + 10^{-10} \rceil$,
if $fright = 1$ set $cor \leftarrow \exp(ar * (right - br + 1)) - 1$,
else set $cor \leftarrow -1$.
Set $volr \leftarrow \exp((br - trx) * ar + try) * cor/(\exp(ar) - 1)$, $sar \leftarrow (logpm - try)/(m - trx)$.

1.5: [*Compute the sum of the p_k's for the center-part (volc) and for the whole envelope (volcompl).*]
Set $volc \leftarrow (br - bl - 1) * pm$, $volcr \leftarrow volc + volr$, $volcompl \leftarrow volcr + voll$.
If $volcompl \geq 3.164 + pm$ go to step 1.1.

2: [*Variate Generation*]

2.0: Generate a uniform random number U and set $U \leftarrow U * volcompl$.

2.1: [*Generation of uniform random number for the center part; immediate acceptance for m.*]
If $U > volc$ go to step 2.2.
Set $k \leftarrow \lfloor U * (br - bl - 1)/volc + bl + 1 \rfloor$. If $k = m$ return k, else set $lek \leftarrow logpm$ and go to step 2.4.

2.2 [*Generation of geometric variate for the right tail; compute lek=log(g_k).*]
If $U > volcr$ go to step 2.3.
Set $k \leftarrow \lfloor \log(((U - volc)/volr) * cor + 1)/ar \rfloor + br$, $lek \leftarrow try + (k - trx) * ar$ and go to step 2.4.

2.3 [*Analogous to 2.2 for left tail.*]
Set $k \leftarrow bl - \lfloor \log(((U - volcr)/voll) * col + 1)/(-al) \rfloor$, $lek \leftarrow tly + (k - tlx) * al$.

2.4 Generate a uniform random number V and set $V \leftarrow \log(V) + lek$.

2.5 [*Testing the left "squeeze"*]
If $k > m$ go to step 2.6.
If $voll > 0$ and $k >= tlx$ and $V \leq logpm - (m - k) * sal$ return k, else go to step 2.7.

2.6 [*Testing the right "squeeze"*]
If $volr > 0$ and $k <= trx$ and $V \leq logpm - (m - k) * sar$ return k.

2.7 [*Final acceptance test*]
If $V \leq \log(p_k)$ return k, else go to step 2.

Remark: Adding 10^{-10} in steps 1.3 and 1.4 guarantees that the intersections between centerpart and tails are rounded towards the tails of the distribution.

4. Comparison of algorithms

First we compare our method with the generator given in [5]. The advantage of our generator lies in the fact that the expected number of iterations is much lower, even in the worst case it is $3.164+p_m$ compared with $4+p_m$ of Devroye's algorithm. For four classical distributions (Poisson, binomial, negative binomial and hypergeometric distribution) we computed the expected number of iterations for many choices of the parameters, it was always well below 1.2, in most cases even below 1.15. Of course this implies that the expected number of uniforms required is much lower for Algorithm DLC as well. Comparative timings with the four classical distributions showed that for fixed parameters DLC is between four and six times faster, on the other hand our algorithm requires a longer setup. Therefore DLC is not much faster if the parameters of the distribution vary after every call. We are convinced that the lower amount of uniform random numbers and the higher speed in the fixed parameter situation justify the more complicated coding (59 C-statements compared with 22) for most applications especially as an universal agorithm is coded only once for a variety of possible applications.
In a second step we compared specialized uniformly fast algorithms for the four classical distributions with our universal method (the evaluation of $\log(p_k)$ was implemented using the Stirling approximation). At first the code of any of the specialized algorithms is at least as long

as Algorithm DLC because all of them utilize two different methods, one for low values of the expectation and one for high ones whereas DLC works for any parameter constellation.
For the Poisson and binomial distributions the fastest known algorithms ([2], [7] and [11], [8]) are about two times faster than DLC, the comparable simple uniformly fast algorithms ([1], [3], [13]) are about as fast as DLC in the fixed parameter situation. In the varying parameter situation DLC is not competitve, because the setup takes four to six times longer than the generation of one random number.
For the hypergeometric and negative binomial distributions the fastest known algorithms (cf. [12], [10] and [4] p 543) have about the same speed as DLC in the fixed parameter case. For the hypergeometric distribution DLC is competitive for the varying parameter situation as well. The set-up is faster than the set-up of the algorithm in [10] and about 50 percent slower than the set-up of the algorithm in [12].

5. Conclusions

We are convinced that the comparisons of section 4 show that Algorithm DLC can well be recommended as a fast but not too complicated universal algorithm for discrete log-concave distributions. The comparative timings justify the conclusion that Algorithm DLC can replace specialized algorithms for classical log-concave distributions for most applications as long as the speed in the varying parameter situation is not of major importance.

References

[1] Ahrens, J. H. and Dieter, U. (1980), Sampling from binomial and Poisson distributions: a method with bounded computation times, Computing 25, 193-208.

[2] Ahrens, J. H. and Dieter, U. (1982), Computer generation of Poisson deviates from modified normal distributions, ACM Transactions on Mathematical Software 8, 163-179.

[3] Ahrens, J. H. and Dieter, U. (1991), A convenient sampling method with bounded computation times for Poisson distributions, in: The Frontiers of Statistical Computation, Simulation & Modelling, P. R. Nelson, Ed., American Sciences Press, 137-149.

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

[5] Devroye, L. (1987), A simple generator for discrete log-concave distributions, Computing 39, 87-91.

[6] Gilks, W. R. and Wild, P. (1992), Adaptive rejection sampling for Gibbs sampling, Appl. Statist. 41, 337-348.

[7] Hörmann, W. (1993), The transformed rejection method for generating Poisson random variables, Insurance: Mathematics and Economics 12, 39-45.

[8] Hörmann, W. (1993), The generation of binomial random variates, Journal of Statistical Computation and Simulation 46, 101-110.

[9] Johnson, N. L. and Kotz, S. (1969), Discrete Distributions, John Wiley, New York.

[10] Kachitvichyanukul, V. and Schmeiser, B. W. (1985), Computer generation of hypergeometric random variates, Journal of Statistical Computation and Simulation 22, 127-145.

[11] Kachitvichyanukul, V. and Schmeiser, B. W. (1988), Binomial random variate generation, Communications of the ACM 31, 216-222.

[12] Stadlober, E. (1990), The ratio of uniforms approach for generating discrete random variates, Journal of Computational and Applied Mathematics 31, 181-189.

[13] Stadlober, E. (1991), Binomial random variate generation: a method based on ratio of uniforms, in: The Frontiers of Statistical Computation, Simulation & Modelling, P. R. Nelson Ed., American Sciences Press, 93-112.