Acceptance Tail Method for Sampling from Unimodal Distributions

Efraim Shmerling

Ariel University, Israel

MCQMC2018

1-6 JULY 2018

Abstract

A universal method for generating continuous random variables (rv's) with unbounded range and infinite-valued discrete rv's is presented.The validity of the method is proved . It is shown in some detail how the method can be implemented for generating unimodal continuous rv's. Theorem that enables one to estimate the efficiency of the unimodal version of the presented method is proved.

INTRODUCTION

A new efficient method named acceptance tail (AT) for generating rv's is presented in the paper. Different versions of the AT method have been developed: a discrete version, a multivariate continuous version, a multimodal univariate continuous version and the univariate unimodal continuous version which will be called the *UMAT* method in further discussion. The development of these versions shows that the AT method like the "classical" acceptance rejection (AR) method extends to all types of rv's except multivariate rv's of high order.

In Section 2 the general AT method is presented. In Section 3 which is the most important part of the paper the *UMAT* method is described in some detail. For the case where the probability density function (pdf) is monotonically decreaing or is a symmetric unimodal function, a very efficient in terms of generation time ziggurat algorithm has been developed by Marsaglia and Tsang (see [1,2]), and became the standard generator for Gaussian rv's in many platforms.

However the ziggurat algorithm has not been implemented for distributions other than Gaussian and exponential, primarily due to complicated and time-expensive setup and the fact that is does not extend to nonmonotonic and nonsymmetric distributions. In section 3 of the paper it is shown how the presented UMAT method overcomes these drawbacks of the ziggurat algorithm while retaining its main advantage which is high computational speed.

DESCRIPTION OF THE GENERAL AT METHOD

In order to describe the general AT method we introduce the following notations. Let X designate the "target" rv . Let the term: "density region" denote in the case where X is a continuous rv the region under the graph of the pdf f(x) of X , and the union of vertical intervals which extend from $(x_i; 0)$ to $(x_i; p(x_i)), x_i \in R, i = \overline{1, \infty}$ in the case where X is a discrete rv.

Here p(x) designates the probability mass function of X. Let H designate in the case where X is a continuous rv a bounded part of R such that the integral $\int_H f(x)dx = 1 - \delta$ is very close to one , and in the discrete case a finite subset of R such that the sum

$$\sum_{i:x_i \in H} p(x_i)$$

is very close to one. Let T designate the complement of R with respect to H. Let C designate the "head" of the density region which is the part of the density region over H, and let D designate the "tail" of the density region which is the part of the density region over T.

Let V(A) designate the volume of a domain A in the case where $A \subset R^k, k \geq 2$, area of A in the case where $A \subset R^2$, and the total length of the "vertical" intervals included in A in the case where A is a union of "vertical" intervals.

The AT method is based on covering C by a set of m instrumental units L_i, $1 \leq i \leq m$ which are "vertical" intervals with common length $1/n$ in the discrete case, rectangles with common area $1/n$ in the univariate continuous case or hyperrectangles with common volume $1/n$ in the multivariate case, $n \geq m$. An algorithm for defining the instrumental units is a part of any specific version of the AT method.

Besides defining the set of instrumental units, an algorithm for generating the "tail" rv G needs to be chosen at the preliminary stage of AT. The pdf $g(x)$ of G is defined by the formula: $g(x) = f(x)/\delta,\ x \in T$ in the case where X is continuous, and the pmf $g(x)$ of G is defined by the formula : $g(x_i) = p(x_i)/\delta,\ x_i \in T$ for the case where X is discrete. Note that the efficiency of the algorithm for generating G does not affect significantly the efficiency of the AT method since the probability that the generation of G will be required in order to obtain a generated value of X is very low as shown below. In order to define the set of instrumental units respective tables need to be initialized at the setup stage of the AT method.

Provided that such tables have been initialized and the algorithm for generating G chosen, the generation of X looks as follows.

1. Select randomly an integer i in the range $1 : n$ (thus one of the covering elements L_i is chosen).
2. If $i \leq m$, generate a point $P = (P1; P2)$ uniformly distributed in L_i; otherwise go to step 4.
3. If P belongs to the density region of X, set $X = P1$.
4. Generate G and set $X = G$.

Obviously, by the presented method a point P is chosen from the density region of X and $P1$ accepted as a generated value of X. (At step 4 a point P is selected from D by the AR method, and P1 is accepted as a generated value of X). Thus in order to prove the validity of the AT method it is enough to show that for any domain A which belongs to the density region of X the probability P(A) that the selected by the AT method point P belongs to A equals V(A).

In order to prove the assertion, let's first consider the case where

$$A \subset \widehat{L_i} = L_i \bigcap C, 1 \leq i \leq m$$

for some $i, 1 \leq i \leq m$. In this case we have

$$P(A) = P(B_i) * P(A|B_i) = (1/n) * (V(A)/(1/n)) = V(A),$$

where the event B_i is selecting the integer i at step 1.

Now let's consider the case where $A \subset D$. In this case in order to show that $P(A) = V(A)$ it is enough to show that the probability $P(D)$ that step 4 will be executed in generating a sampled value of X by the AT method equals $V(D) = q$. Obviously, P(D) can be expressed as the sum $(n-m)/n + \sum_1^m P_i(D)$, where the first term is the probability that an integer larger than m will be chosen at step 1, and $P_i(D)$ is the probability that a given integer i less or equal to m will be chosen at step 1 and the point P generated at step 2 will not belong to $\widehat{L_i}$.

The probability $P_i(D)$ equals $(1/n) * ((1/n - V(\widehat{L_i}))/(1/n)) = 1/n - V(\widehat{L_i})$. It follows that the sum $\sum_1^m P_i(D)$ equals $\sum_1^m (\frac{1}{n} - V(\widehat{L_i})) = \frac{m}{n} - \sum_1^m V(\widehat{L_i}) = \frac{m}{n} - V(C) = \frac{m}{n} - (1-q)$, therefore P(A)=P(D). Q. E. D.

THE UNIVARIATE UNIMODAL VERSION OF THE AT METHOD

Let's consider the case where the pdf $f(x)$ of the "target" rv X is monotonically decreasing with mode $M = 0$. n horizontal rectangular layers L_i, $1 \leq i \leq n$ with common area $1/n$ need to be obtained according to the procedure described below, a layer i extends from 0 to x_i horizontally and from y_i to y_{i-1} vertically, $x_0 = 0, y_0 = f(0)$, $x_{1-1} \leq x_i$, $y_i < y_{i-1}$, $1 \leq i \leq n$. The obtained instrumental units define H as $H = \{x : 0 \leq x \leq x_n\}$.

At the setup stage of the *UMAT* method 2 tables: $\{x_i\}_{i=1}^n$, $\{y_i\}_{i=1}^n$ have to be generated. Let G designate the "tail" rv with pdf $g(x) = f(x)/\delta, x > x_n$. Then the generation stage of the *UMAT* method looks as follows.

1. Select randomly an integer i in the range $1 : n$.
2. Generate U_0 uniformly distributed in $(0, 1)$ and calculate $s = U_0 * x_i$.
3. If $s < x_i$, set $X = s$.
4. Generate U_1 uniformly distributed in $(0, 1)$ independently of U_0.
5. If $U_1 < (f(s) - y_i)/(y_{i-1} - y_i)$, set $X = s$;
6. Generate G and set $X = G$.

The initialization of the tables required by the *UMAT* method looks as follows. First the values of x_i , y_i $i = 1, 2, ...$ are calculated recursively utilizing the equalities:

$$(f(x_i) - y_{i-1}) \cdot x_i = \frac{1}{n},\ y_i = f(x_i),\ x_0 = 0,\ y_0 = f(0). \qquad (1)$$

The consecutive values of x_i are calculated until the point x_{i^*}, for which the condition $S(i^*) = \sum_{j=1}^{i^*}(x_j - x_{j-1}) \cdot f(x_{j-1}) > 1$ is satisfied, is obtained. Since $S(i^* - 1) < 1$ and $S(i^*) > 1$, the following inequality holds:
$x_{i^*-1} < d = (n - i^* + 1)/(n \cdot f(x_{i^*-1})) < x_{i^*}$. We set
$x_i = d, i^* \leq i \leq n$, and we set
$y_i = f(x_i)$, $1 \leq i < i^*$, $y_i = f(x_{i^*-1}/(n - i^* + 1) \cdot (n - i)$, $i^* \leq i \leq n$.

Note that ziggurat method is also based on coverage of the density region by horizontal layers, all of which except one rectangular. However, unlike the $UU\ AT$ method, the layers cover the entire density region, and the common area A of the layers is greater than $1/n$, though not significantly.Let's show graphically the difference between the layers required by the ziggurat method and the layers required by the $UU\ AT$ method. Crude version of the ziggurat method is pictured in Figure 1 (8 layers were chosen for clarity: in practice the number of layers is many times larger). The density region under the curve $y = f(x) = e^{-\frac{x^2}{2}}$, $x > 0$ is covered with 7 rectangles and a bottom string tailing off to infinity. All the sets are of area A greater than $1/16$.

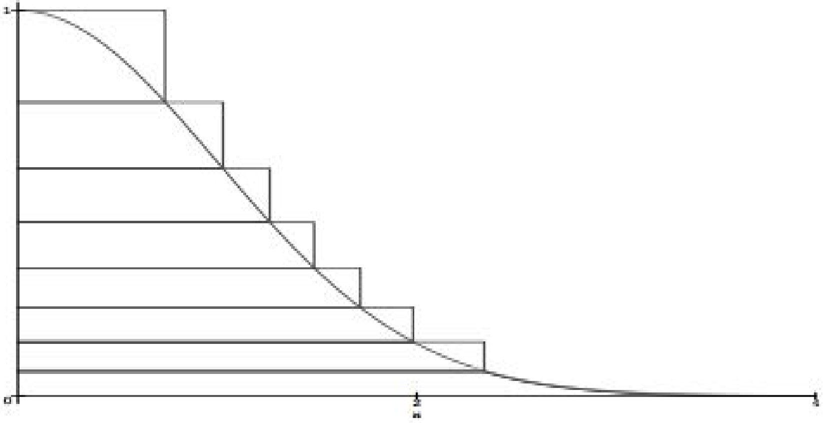

Figure 1

The AT method requires utilizing rectangles of area $1/16$ in this case. If we build 8 horizontal rectangular layers of area $1/16$ starting with the top layer L_1, the coverage of the entire density region except the bottom string will be obtained. This is shown in Figure 2.

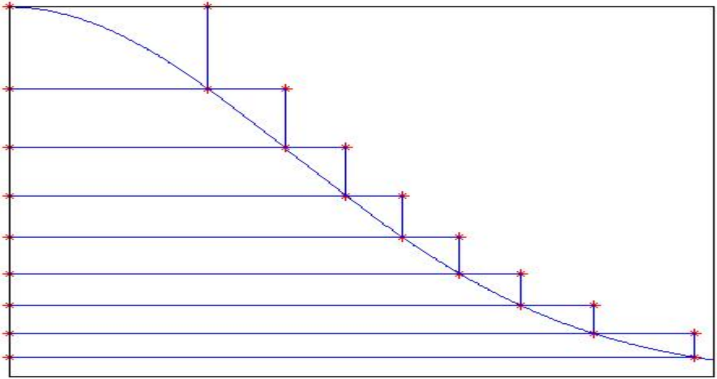

Figure 2

In order to obtain the set of 8 instrumental rectangles required by the AT method, we need to check for each generated layer L_i whether the combined area $T(i)$ of i generated layers and the area $x_i * y_i$ of the rectangle below L_i is less than one. Having generated L_6, we found that the condition is not satisfied, we seek a point d on the x-axis such that $d * x_5 = \frac{3}{16}$ and divide the rectangle under L_5 into 3 congruent rectangular layers L_6,L_7 and L_8 as shown in Figure 3.

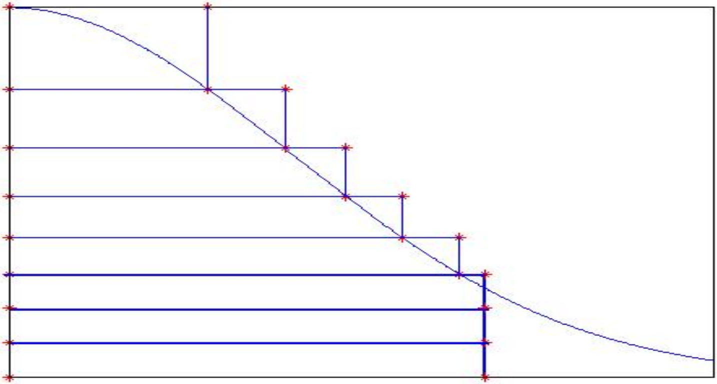

Figure 3

Note that the generation of the tables requires calculating a root of the equation (1) less than n times. Ziggurat method requires the calculation of the common area A of the layers utilizing an iteration method (such as the bisection method), and at each iteration the equation $f(x) = y_i$, where $y_i = y_{i-1} + A/x_{i-1}$ has to be solved for each i, $1 < i \leq n$ ($n - 1$ times), from which follows that the setup time required by the *UMAT* method is many times shorter than the setup time of the ziggurat.

The main advantage of the *UMAT* method is its computational speed which follows from the fact that the probability $P^*(n)$ that only the first 3 steps of the algorithm will be executed in order to obtain a generated value of the "target" rv X is close to one for sufficiently large n. In order to estimate $P^*(n)$ we introduce the following notation.

Let $E_i, 1 \leq i \leq i^*$ designate the rectangular part of L_i bounded on the left by the line $x = x_{i-1}$, and let $S(A)$ designate the area of an arbitrary region A in R^2. Then $P^*(n)$ can be calculated via the following formula: $P^*(n) = 1 - S(U_1^{i^*} E_i)$. The following theorem proves that $P^*(n)$ tends to one as n tends to infinity and enables one to estimate the rate of convergence.

Theorem 1 If the derivative of $f(x)$ is bounded : $\mid f^{'}(x) \mid \leq M, x > 0$, then for any $\delta > 0$ and any n which satisfies the inequality

$$l(n) = \frac{2 + ln(\sqrt{Mn} \cdot x_{\delta})}{n} < \delta \quad (2)$$

the following inequalities hold: $x_n > x_{\delta}$ and $P^{*}(n) > 1 - 2\delta$. Here x_{δ} designates the point on the x-axis for which the equality $\int_{x_{\delta}}^{\infty} f(x)dx = \delta$ is satisfied.

Proof

In order to prove the theorem we introduce the following notations. Let $E(x', x'')$ designate the part of $\bigcup_{i=1}^{*} E_i$ between the lines $x = x'$ and $x = x''$, where x' and x'' are some points on the x-axis. Let $\tilde{E}(x', x'')$ designate the part of $E(x', x'')$ over the graph of f(x), and let $\hat{E}(x', x'')$ designate the part of $E(x', x'')$ below the graph of $f(x)$.
The inequality

$$S(E(kx_1, (k+1)x_1)) < \frac{1}{kn} \quad (3)$$

holds for any $k \geq 1$ since $E(kx_1, (k+1)x_1)$ consists of nonintersecting rectangles, the total length of which equals x_1, and the width of each of them is less than $\frac{(y_0 - y_1)}{nk}$. (Note that $(y_0 - y_1) \cdot x_1 = \frac{1}{n}$)). Let $\eta = \lfloor \frac{x_\delta}{x_1} \rfloor$. The inequality

$$S(E(0, x_\delta)) < \frac{1}{n}(1 + \sum_{k=1}^{\eta} \frac{1}{k}) \quad (4)$$

holds in view of (3).

Utilizing the inequalities $\sum_{k=1}^{\eta} \frac{1}{k} \leq 1 + \ln \eta$, $x_1 > \frac{1}{\sqrt{nM}}$ and $\eta < \frac{x_\delta}{x_1}$, we obtain the inequality $S(E(0, x_\delta)) < I(n)$. The first assertion of the theorem ($x_{\boldsymbol{n}} > x_\delta$ if n satisfies (2)) follows. Indeed, we have $S(E(0, x_{\boldsymbol{n}})) > \bar{F}(x_{\boldsymbol{n}}) = \delta_{x_n}$. On the other hand, we have $S(E(0, x_\delta)) < \delta$ if n satisfies (2). These two inequalities imply $x_{\boldsymbol{n}} > x_\delta$ since it follows from $x_{\boldsymbol{n}} < x_\delta$ that $\delta_{x_n} > \delta$ and $S(E(0, x_{\boldsymbol{n}})) = S(E(x_\delta))$. The second assertion of the theorem follows directly from the inequality $S(E(x_\delta, x_{\boldsymbol{n}})) < \delta$ which holds in view of $\tilde{E}(x_\delta, x_{\boldsymbol{n}}) < \delta_{x_n}$ and $\hat{E}(x_\delta, x_{\boldsymbol{n}}) < \delta - \delta_{x_n}$.
Q.E.D.

Gamma Generator

Now let us consider the AT algorithm for generating gamma rv X with shape parameter $\alpha > 1$ and mode M. The covering rectangles(layers) $L_i, \ 1 \leq i \leq n$ with common area 1/n, length of each rectangle L_i equal to $\mid x_i - M \mid$ extend from y_i to $y_i + h_i$ along the y-axis. Most covering rectangles can be divided into two rectangles: the greater one with area p_i/n is situated between the vertical lines $x = x_{i-1}$ and x=M and lies entirely within the density and the smaller one which lies partly within the density of X.

Provided that a set of covering rectangles has been defined (the values of x_i, y_i, p_i and $hi; 1 \leq i \leq n$; found), the generation of the "target" rv X consists of the following steps.

1. Choose randomly an integer $i, 1 \leq i \leq n$
2. Generate U_0 uniformly distributed in (0; 1) and set $s = M + (x_i - M) \cdot U_0$.
3. If $U_0 < p_i$, set x = s.
4. Generate U_1 independently of U_0.
5. If $U_1 < (f(s) - y_i) = h_i, setx = s$.
6. Generate G and set $x = G$.

Here G designates the "tail" rv with pdf $g(t) = f(t)/\int_{x_n}^{\infty} f(z)dz, t > x_n$.

It can be shown in the way similar to the proof of the theorem presented above that only the first 3 steps of the algorithm are executed for a typical table size which makes the AT gamma generation algorithm almost as fast as the ziggurat in terms of generation time, while the setup of the AT algorithm is very significantly less time-expensive, which is critically important because in the case of sampling from gamma distributions pre-computed stored tables can't be utilized.

A simple and fast algorithm for initializing the tables that define the covering rectangles has been designed. The algorithm looks as follows. Set $x_0 = M, y_0 = f(M)$. Calculate $xi; y_i; i = 1; 2; ...$ according to the following recursive procedure: for any $i \geq 1$ find the real root $\tilde{x}_i$ of the equation (1) $gi(x) = y_{i-1} \cdot (M - x_{i-1}) - \frac{1}{n} = 0, x < M$ set $x_i = \tilde{x}_i$ and $y_i = f(x_i)$ provided that the inequality $(M(x_{i-1}) \cdot y_{i-1} > \frac{1}{n}$ is satisfied. If the inequality is not satisfied for some $i = i^{'}$, stop the recursive procedure and set $x_i^{'} = M - \frac{1}{y_{i'-1} \cdot n}, y_{i'} = 0$. Then set x_{i+1} equal to the real root of the equation $g_{i'+1}(x) = (y_0 - f(x)) \cdot (x - M) - \frac{1}{n} = 0$ and set $y_{i'} + 1 = f(x_{i'+1})$.

Calculate $x_i, y_i, i = i' + 2, i' + 3, ...$ according to the recursive procedure: for each i find the real root$\tilde{x}_i$ of $g_i(x) = (x - M)(y_{i-1} - f(x)) - \frac{1}{n} = 0, x > M$, set $x_i = \tilde{x}_i, y_i = f(x_i)$ provided that the inequality $\tilde{x}_i \cdot f(\tilde{x}_i) > \frac{1}{n \cdot (n-i+1)}$ is satisfied . If the inequality is not satisfied for some $i = i''$, stop the recursive procedure and set $x_i'' = \frac{n-i''+1}{y_{i''-1} \cdot n}, y_i'' = y_{i''-1} \cdot \frac{n-i''}{n-i''+1}$. Set $xi = x_i''; y_i = (y_{i''-1}/n - i'' + 1) \cdot (n - i), i'' + 1 \leq i \leq n$ In order to initialize the tables we need to solve an equation of the form (1) almost n times. Since it is very important to minimize the setup time, the fastest possible numerical root-finding method for solving the equations needs to be utilized. It is well known that in the case where for some continuous monotonous function$h(x)$ the condition $h(a) \cdot h(b) < 0$ is satisfied, $h''(x)$ does not change sign in the interval [a; b] and f'(x) can be calculated in a very easy way provided that the value of f(x) has been found for any $x \in [a; b]$ the performance of the Newton's method is superior to the performance of any alternative root-finding method, for example, the secant method.

Luckily, all the conditions are satisfied for the equations that need to be solved in order to initialize the tables, therefore in order to solve an equation $g_i(x) = 0$ only several calculations of the value of f(x) will be required which makes the setup time insignificant compared to the generation time for sufficiently big samples.
The tables $\{p_i\}_1^n$ and $\{h_i\}_1^n$ are initialized in initialized in the following way:

$$p_i = (M - x_i)/(M - x_i), 1 \leq i \leq n, i \neq i' + 1, p_{i'+1} = 1$$

$$h_i = (y_{i-1} - y_i, 1 \leq i \leq n, i \neq i' + 1, h_{i'+1} = f(M) - f(y_i)$$

Software for initializing the tables has been designed and carefully tested in experimental calculations.

CONCLUDING REMARKS

1. A *RNG* for generating gamma rv's implementing the *UU AT* method has been developed. It was carefully tested in experimental calculations and showed extremely good results in terms of speed and accuracy
2. Size n of the covering set (which is also the size of the tables that need to be generated) affects the efficiency of the *AT* method. With growth of n the "tail" diminishes, therefore the probability that step 4 of the *AT* method will be executed decreases. Besides that n is best chosen as power of 2 to make random selection easy
3. In the near future we expect to develop algorithms and $RNG's$ for various widely applied rv's and classes of rv's implementing the AT method.
4. Summarizing the discussion in the paper, we assert that the presented AT method is universal, simple and fast, which enables one to expect that in the near future $RNG's$ implementing the *AT* method will be incorporated as standard generators in most widely used platforms.

References

- L. Devroye, *Non-Uniform Random Variate Generation*,(Springer-Verlag,New York,1986),Ch.9(3).

- G. Marsaglia and W.W. Tsang, ”A simple method for generating gamma variables”. ACM Transactions on Mathematical Software 26(3),363-372 (2000).

- G. Marsaglia and W.W. Tsang, ”The ziggurat method for generating random variables”, Journal of Statistical Software 5,17 (2000),www.jstatsoft.org.

THANKS FOR YOUR ATTENTION