SAMPLING FROM A DISCRETE DISTRIBUTION WHILE PRESERVING MONOTONI--ETC(U)
FEB 82  G S FISHMAN; L R MOORE
N00014-26-C-0302
SAMPLING FROM A DISCRETE DISTRIBUTION

WHILE PRESERVING MONOTONICITY

George S. Fishman and Louis R. Moore, III

Technical Report No. UNC/ORSA/TR-81/7

Curriculum in Operations Research
and Systems Analysis
University of North Carolina at Chapel Hill

This research was supported by the Office of Naval Research under contract N00014-26-C-0302. Reproduction in whole or part is permitted for any purpose of the United States government.
ABSTRACT

This paper describes a cutpoint method for sampling from an n-point discrete distribution that preserves the monotone relationship between a uniform deviate and the random variate it generates. This property is useful when developing a sampling plan to reduce variance in a Monte Carlo or simulation study. The alias sampling method generally lacks this property and requires 2n storage locations while the proposed cutpoint sampling method requires m+n storage locations, where m denotes the number of cutpoints. The expected number of comparisons with this method is derived and shown to be bounded above by (m + n - 1)/n. The paper describes an algorithm to implement the proposed method as well as two modifications for cases in which n is large and possibly infinite.
I. INTRODUCTION

Let $X$ be a discrete random variable on the integers $1,\ldots,n$ with probability mass function \{${p_i; i=1,\ldots,n}$\} and distribution function (d.f.)

$$q_0 = 0$$
$$q_i = q_{i-1} + p_i \quad i=1,\ldots,n. \quad (1.1)$$

One straightforward way to sample from \{${q_i; i=1,\ldots,n}$\} is to sample $U$ from the uniform distribution on $(0,1)$ and then determine $X$ from

$$X = \min \{i: q_i \geq U\}. \quad (1.2)$$

If the d.f. of $X$ is stored in a table beforehand, this procedure, known as the inverse transform method, requires $n$ storage spaces and $EX$ comparisons on average, which may prove costly when $n$ is moderate or large.

At present, the most efficient way to sample repeatedly from the d.f. of $X$ is the alias method of Walker (1974a, 1974b, 1977). See also Krommal and Peterson (1979). This method requires storage for two arrays, the aliases $A_1,\ldots,A_n$ and alias probabilities \{${r_i = \Pr(X=i|L=i); i=1,\ldots,n}$\} where for $i=1,\ldots,n$. 

\begin{equation}
\text{pr}(L=i) = \frac{1}{n}
\end{equation}

\begin{equation}
\text{pr}(X=i|L=i) + \text{pr}(X=A_i|L=i) = 1 
\end{equation}

\text{and}

\begin{equation}
p_i = \text{pr}(X=i) = \frac{1}{n} \sum_{j=1}^{n} \text{pr}(X=i|L=j).
\end{equation}

Prior to beginning the sampling experiment one chooses these arrays to satisfy (1.3). After sampling \( U \) one computes \( L = \lceil nU \rceil + 1 \) and selects \( X = L \) if \( r_L \geq U(\mod 1/n) \) or otherwise selects \( X = A_L \). Here \( \lceil \theta \rceil \) denotes the largest integer less than or equal to \( \theta \). Note that only one comparison is required to generate each \( X \). The arrays determined by (1.3) require \( 2n \) storage locations and are not unique. If one wishes to retain the tabled values of \( \{q_i\} \), an additional \( n \) storage locations are required.

Although the time independent nature of the alias method has clear appeal, the method has two limitations that deserve attention:

a. In general, the alias method does not preserve a monotone relationship between \( U \) and \( X \) as does the inverse transform method (1.2).

b. The allocation of \( 2n \) storage spaces may be infeasible either due to the magnitude of \( n \) or the requirements of other steps in the program in which the alias method is imbedded.

Furthermore, if one wishes to maintain the table of \( \{q_i\} \), \( 3n \) storage locations are required.
While the issue of storage requirements is self explanatory, the significance of the monotone property needs clarification. Let $Y_1$ and $Y_2$ be random variables with d.fs. $F_1$ and $F_2$ and inverse d.fs. $G_1(u) = \min(y: F_1(y) > u)$ and $G_2(u) = \min(y: F_2(y) > u)$ respectively. Then the minimal correlation between $Y_1$ and $Y_2$ occurs for $Y_1 = G_1(U)$ and $Y_2 = G_2(1-U)$. The result is due to Hoeffding (1940). See also Whitt (1976). In Monte Carlo sampling and discrete event simulation one often wants to make use of the minimal correlation property to induce a variance reduction for a given sampling cost. More generally, one often can achieve a variance reduction by appropriate use of the sequence $U_k = U + (k-1)/r \ (\text{mod} \ 1)$ for $k=1, \ldots, r$ with a sampling technique that preserves monotonicity. See Hammersley and Handscomb (1964) and Fishman and Huang (1980). The alias method may prevent one from effecting this reduction in variance.

2. THE CUTPOINT METHOD

We now describe the cutpoint method for sampling from the d.f. of $X$. The procedure preserves monotonicity, maintains the table of $\{q_i\}$ and allows the user to adjust space and time requirements to accommodate the global needs of the problem setting. The procedure again uses the inverse transform approach but with more information computed beforehand, as in the alias method. The proposed method is not new having been described in Chen and Asau (1974). However, the present paper is the first to study the tradeoff between computation time and space analytically.
For a given positive integer \( m \), define the cutpoints

\[
I_k = \min \{ i : q_i > (k-1)/m \} \quad k=1, \ldots, m \tag{2.1}
\]

\[I_{m+1} = n.\]

Let \( L = \lceil m \rceil \) so that

\[
\Pr( I_L \leq X \leq I_{L+1} ) = 1 \tag{2.2}
\]

where \( X \) is as defined in (1.2) and \( \lceil \cdot \rceil \) denotes the smallest integer greater than or equal to \( \cdot \). The maximal number of comparisons needed to determine \( X \) exactly is \( I_{L+1} - I_L + 1 \) and the expected maximal number of comparisons is

\[
C_{m,n} = (I_{m+1} - I_1 + m)/m. \tag{2.3}
\]

The storage requirements are \( m+n \) locations with the tabled d.f. of \( X \) comprising \( n \) of them.

Note that \( C_{n,n} \) is less than 2, implying that the cutpoint method requires less than one additional comparison on average to preserve monotonicity when compared to the alias method with the same allocation of storage. If the d.f. of \( X \) also is to be maintained in the alias method, then for equal storage for the cutpoint method \( C_{2n,n} \) is less than 1.5 so that at most 1/2 of an additional comparison is needed on average.

A more revealing evaluation arises if the d.f. of \( X \) is directly taken into account. Let \( J_\cdot \) denote the number of comparisons on a trial. Then \( J_\cdot \) equals \( X - I_L + 1 \), where \( X \) is determined as in (1.2), and has expectation
\[ EJ_m = 1 + EX - EI_L \]
\[ = 1 + EX - \sum_{k=1}^{m} \frac{I_k}{m} \]  

(2.4)

Table 1 illustrates the proposed method using the eight-point distribution in Fishman (1978, p. 459).

---

If space is not an issue then the alias method is the procedure of choice. One then may view \( EJ_m - 1 \) as the cost of maintaining the monotone property.

3. **THE CASE OF LARGE \( n \)**

If \( \{q_i\} \) has infinite support (\( n = \infty \)) then neither the alias method nor our cutpoint method alone suffices to perform sampling. This insufficiency also may occur if \( n \) is merely large relative to space availability. Here Kronmal and Peterson (1979) suggest using the alias method "for a finite (but large) range of the desired discrete distribution and a special tail-generating method for the tail beyond". Ahrens and Dieter (1973) discuss the tail-generating methods for several common parametric families of distributions. More recently Ahrens and Kohrt (1981) described a cutpoint method with a more dense frequency of cutpoints in the tails.

Our own proposals for this situation take two forms. Note that in principle (2.4) suggests that \( EJ_m \) may be determined for infinite \( n \) if \( EX \) is finite. Consider \( n^* < n \) such that a procedure is available for computing \( \{q_i; i > n^*\} \). The cutpoint
method then applies directly with tables used for \( \{q_i; i \leq n^*\} \)
and the available procedure for \( i > n^* \). The mean number of com-
parisons remains the same while the mean cost in time is propor-
tional to

\[
\mu_1 = EJ_m + c_1 \ E(J_i X > n^*) \Pr(X > n^*)
\]  

(3.1)

where \( c_1 \) is the (computer dependent) increase in time required to
evaluate \( q_i \) for \( i > n^* \) relative to the time required for a table
lookup of \( q_i \) when \( i \leq n^* \).

As an alternative, suppose that in addition to tabling
\( \{I_L; L=1,\ldots,m\} \) we elect to table \( \{q_{I_1}, q_{I_2}, \ldots, q_{I_m}\} \) instead of
\( \{q_1, q_2, \ldots, q_n\} \). To sample \( X \), one proceeds, as before,
to select \( I_L \) but now calculates \( q_{I_{L+1}}, q_{I_{L+2}}, \ldots \) as needed.
Such calculations may be faster and more accurate due to the
nearby starting value \( q_{I_L} \). The mean cost in time is propor-
tional to

\[
\mu_2 = (1 + c_2)EJ_m - c_2
\]  

(3.2)

where \( c_2 \) is the (computer dependent) relative increase in time
required to evaluate \( q_i \) for \( i \neq I_L \) as compared to the time re-
quired for a table lookup of \( q_{I_L} \).

Clearly \( \mu_2 \leq \mu_1 \) if and only if

\[
c_2 \ E(J_{m-1}) - c_1 \ E(J_{m} X > n^*) \Pr(X > n^*) \leq 0 .
\]  

(3.3)

In addition, the second procedure requires \( 2m \) storage locations
while the first requires \( m+n^* \). For most applications \( c_1 > c_2 > 0 \)
and as \( n^* \) increases the left hand side of expression (3.3) mo-
notonically increases from a negative value of \((c_2 - c_1)EJ_m - c_2\).
when \( n^* = 0 \), to a positive limit of \( c \frac{EJ}{m} - c_2 \), if \( EK < 0 \).

Again, the user is faced with a tradeoff between time and space requirements.

These remarks are intended to be suggestive rather than definitive. Individual decisions should be guided by the requirements and resources of the application intended. Although the dominance of any of the methods described here with respect to time and storage remains a question, one should keep in mind that all the cutpoint methods described preserve the monotone property.

4. ALGORITHMS

In this section we present algorithms for implementing the cutpoint method when sampling from the d.f. of the random variable \( X \). It is assumed throughout that \( Q \) is the name of an array or function such that \( Q(i) = q_i \). In addition, \( x \) denotes the smallest integer greater than or equal to \( x \).

Given the positive integer \( M = m \), the algorithm CMSET in Figure 1 returns the array \( \{ I(l) = I_l; l=1,...,M \} \), as described in Section 2. If desired, the array \( \{ QI(l) = q_I; l=1,...,M \} \) suitable for use as described in Section 3 is also returned; otherwise, one deletes statement 8 of CMSET. Note that if the d.f. of \( X \) is tabulated, the array \( Q \) is not destroyed by CMSET.

Algorithm CM in Figure 2 enables one to sample for \( X \), once the setup in algorithm CMSET has been effected. An algorithm for sampling from the uniform distribution on \([0,1)\) is used in the
first step of CM. The random variable \( X \) need not have finite support in order for CM to function correctly. The value of \( X \) upon return from CM is the variate desired.

Figure 2 About Here

Algorithms for the setup and use of the alias method are given in Kronmal and Peterson (1979) and Ahrens and Kohrt (1981). Care must be taken with these algorithms to preserve \( \{q_i\} \) and avoid the use of large arrays during initialization.
TABLE 1. EXAMPLE OF CUTPOINT METHOD

<table>
<thead>
<tr>
<th>i</th>
<th>$p_i$</th>
<th>$q_i$</th>
<th>$m$</th>
<th>$m+n$</th>
<th>$C_{m,n}$</th>
<th>$EJ_m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.01</td>
<td>.01</td>
<td>1</td>
<td>9</td>
<td>8.00</td>
<td>5.31</td>
</tr>
<tr>
<td>2</td>
<td>.04</td>
<td>.05</td>
<td>2</td>
<td>10</td>
<td>4.50</td>
<td>3.31</td>
</tr>
<tr>
<td>3</td>
<td>.07</td>
<td>.12</td>
<td>3</td>
<td>11</td>
<td>3.33</td>
<td>2.31</td>
</tr>
<tr>
<td>4</td>
<td>.15</td>
<td>.27</td>
<td>4</td>
<td>12</td>
<td>2.75</td>
<td>2.06</td>
</tr>
<tr>
<td>5</td>
<td>.28</td>
<td>.55</td>
<td>5</td>
<td>13</td>
<td>2.40</td>
<td>1.71</td>
</tr>
<tr>
<td>6</td>
<td>.19</td>
<td>.74</td>
<td>6</td>
<td>14</td>
<td>2.17</td>
<td>1.64</td>
</tr>
<tr>
<td>7</td>
<td>.21</td>
<td>.95</td>
<td>7</td>
<td>15</td>
<td>2.00</td>
<td>1.45</td>
</tr>
<tr>
<td>8</td>
<td>.05</td>
<td>1.00</td>
<td>8</td>
<td>10</td>
<td>1.88</td>
<td>1.44</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>16</td>
<td>24</td>
<td>1.44</td>
<td>1.19</td>
</tr>
</tbody>
</table>
FIGURE 1. ALGORITHM CMSET

1. \( L = 0. \)
2. \( J = 0. \)
3. \( J = J + 1. \)
4. \( Q_J = M^*Q(J). \)
5. IF \( Q_J \leq L \) THEN GO TO 3.
6. \( L = L + 1. \)
7. \( I(L) = J. \)
8. \( QI(L) = Q_J/M. \)
9. IF \( L < M \) THEN GO TO 5.
10. RETURN.

FIGURE 2. ALGORITHM CM

1. SAMPLE U FROM UNIFORM \([0,1)\).
2. \( X = I([M^*U]). \)
3. IF \( U \leq Q(X) \) THEN RETURN.
4. \( X = X + 1. \)
5. GO TO 3.
5. References


**Title:** Sampling From A Discrete Distribution While Preserving Monotonicity

**Authors:** George S. Fishman and Louis R. Moore, III

**University of North Carolina**
Chapel Hill, North Carolina 27514

**Performing Organization Name and Address:**
Operations Research Program
Office of Naval Research
Arlington, VA

**Report Date:** February, 1982

**Number of Pages:** 13

**Distribution Statement:** Distribution of this document is unlimited.

**Key Words:**
- sampling
- alias method
- cutpoint method

**Abstract:**
This paper describes a cutpoint method for sampling from an n-point discrete distribution that preserves the monotone relationship between a uniform deviate and the random variate it generates. This property is useful when developing a sampling plan to reduce variance in a Monte Carlo or simulation study. The alias sampling method generally lacks this property and requires 2n storage locations while the proposed cutpoint sampling method requires mn storage locations, where m denotes the number of cutpoints. The
20. The expected number of comparisons with this method is derived and shown to be bounded above by \((m + n - 1)/n\). The paper describes an algorithm to implement the proposed method as well as two modifications for cases in which \(n\) is large and possibly infinite.