

## 単純非復元抽出のための乱数

慶應義塾大 理工 数理科学

渋谷 政昭

(MASAAKI SIBUYA)

## 1. 問題

人々が乱数を使用し始めた主な動機はサンプリングである。その使用法はすでに十分に確立されているように思われているが、大規模データからのサンプリングで、たとえば逐次ファイルにある  $10^3 \sim 10^7$  の規模のデータの一部をひたすらに抜き出す場合とすると、そのための乱数の効率良い生成法がなお問題となる。この場合には  $\{1, 2, \dots, N\}$  から大きさ  $n$  の単純非復元抽出標本を、大きさの順に並べたものが必要である。

改善の可能性を Knuth (1981) が示唆し、Kawarasaki and Sibuya (1982) と Vitter (1982) が解いたが、素朴な方法で  $S$ -sort と呼ぶ整順列化の方法を用いると、さらに改善できることを報告する。話の要旨は、 $n$  個の数の整順列化には  $O(n \log n)$  の手間を要することが計算機科学の常識 = 固定観念とらっているが、 $n$  個の数が一様分布に近いならば  $O(n)$  の手間で整順列化できる、という事実である。

必要乱数は,  $n$ 次元正整数値確率変数  $X = (X_1, \dots, X_n)$ ,

$$p_0(x_1, \dots, x_n; N, n) := \Pr[X = (x_1, \dots, x_n)] \\ = \begin{cases} 1/\binom{N}{n}, & 1 \leq x_1 < \dots < x_n \leq N \quad \text{の時,} \\ 0, & \text{その他,} \end{cases}$$

の実現値とみなされるものである.  $n \leq N/2$  として一般性を失わず. そうでない場合は逆はれる... ものを指定すればよい.

## 2. 生成法の方針

方針 A. 整数区間  $[1, N]$  上の一様乱数を生成し, 整順列にし, 同じ数が複数個現われたら, つまり衝突が生じたならば再生成する. 整順列とする算法および衝突処理のうまさの問題とする. 衝突数の期待値は, 古典的占有分布から求められ,  $N(1 - 1/N)^n - N + n \doteq n^2/2N$  である.

方針 B. 確率変数  $(Z_1, \dots, Z_N)$  を上記の  $X$  を用いて次のように定義する.  $X_\alpha = j$  とする  $\alpha$  ( $1 \leq \alpha \leq n$ ) が存在すれば  $Z_j = 1$ , さもないと  $Z_j = 0$ . そうすると,

$$\Pr[Z_j = 1 \mid \sum_{i=1}^{j-1} Z_i = k] = (n-k)/(N-j+1),$$

$$0 \leq k \leq \min(j-1, n), \quad j = 1, 2, \dots, N.$$

これは,  $N$  が大きく,  $n/N \neq 1/2$  のときには有効な方法である.  $N$  回の抽選を要するが, 1回の抽選は  $(0, 1)$  - 様乱数を生成し  $(n-k)/(N-j+1)$  と比較するだけである.

方針 C.  $(X_1-1, X_2-X_1-1, \dots, X_j-X_{j-1}-1, \dots, X_n-X_{n-1}-1)$  の実現値とみられる乱数を生成する.  
 $X = X_1-1$  の確率分布は,

$$\begin{aligned} p(x; N, n) &:= \Pr[X = X_1-1 = x] = \binom{N-x-1}{n-1} / \binom{N}{n} \\ &= \frac{(-1) \binom{-n}{N-n-x}}{\binom{-n-1}{N-n}} = \frac{n(N-x-1) \binom{n-1}{N-n}}{N \binom{n}{N-1}} = \frac{n}{N} \frac{(N-n) \binom{n}{N-1}}{\binom{n-1}{N-1}}, \\ &0 \leq x \leq N-n, \end{aligned}$$

これは負の超幾何分布の特別な場合である.  $X_j$  の意味から明らかだが,

$$\begin{aligned} \Pr[X_j - x_{j-1} - 1 \mid X_{j-1} = x_{j-1}] &= p(x; N - x_{j-1}, n - j + 1) \\ 1 \leq j \leq n \quad (n \geq 1, x_0 = 0), \end{aligned}$$

であり, パラメータの値を変えよことにより,  $X_1, X_2, \dots, X_n$  を生成できる. この確率分布は

$$\Pr[X = s+x \mid X \geq s] = p(x; N-s, n)$$

とこの性質をもち,  $\lambda \doteq n/N$  のときには幾何分布

$$g(x, \lambda) := \lambda(1-\lambda)^x, \quad x = 0, 1, \dots; \quad 0 < \lambda < 1,$$

に非常に近い。

方針 C に従い Kawarasaki & Sibuya は算法 NHA, NHB を, Vitter は算法 D1, D2 を独立に開発した。D1 が他と異なり簡単であるので、最初に紹介する。

### 算法 D1

1.  $(0, 1)$ -様乱数  $U$  を生成し  $X = \lfloor N(1-U^{1/n}) \rfloor$  とおく。 $(1-U^{1/n})$  は確率密度  $n(1-u)^{n-1}$ ,  $0 < u < 1$ , をもつベータ乱数である。
2.  $(0, 1)$ -様乱数  $V$  を生成する。
3.  $V \leq (N-n+1)(1-X/(N-n+1))^{n-1} / N(1-X/N)^{n-1}$  ならば  $X$  を採択する。
4.  $V \leq (N-n+1)(N-n)^{(X)} / N(N-1)^{(X)} (1-X/N)^{n-1}$  ならば  $X$  を採択する。
5. さもないければ 1 に戻る。

Vitter は, 上記の各段の実行時間を  $d_1, d_2, d_3, d_4$  とすると, 乱数 1 個生成の平均時間は  $(N/N-n+1)(d_1+d_2+3d_3+d_4)$  であり, これにより  $X_1$  が生成できるので,  $(X_1, \dots, X_n)$  を生成するための時間は  $O(n)$  であると論じている。これは  $O(n)$  の可能性を示すにはよい方法であるが,  $X$  の生成に  $1/n$  乗の計算があるし, 第 3 段の  $n-1$  乗の計算も 1 回

にすぎないというので、実際にはあまり遅くない。

NHB と D2 は  $\lambda = (n-1)/(N-1)$  の幾何分布の棄却法で類似しているが、NHB の方が squeeze を徹底してあり、算法が少し複雑になるが効率がよい。NHA はさらに効率を良くするために  $\lambda = n/N$  の幾何分布を用い、棄却はしない。この幾何分布乱数が  $C = L(2N-n)/(n+1)$  以下ならば採択、超えたならば適当な確率で  $C+1$  跳ばし、 $X \geq C+1$  の条件の下で  $X$  を生成する。その他の場合には  $[0, C]$  の間を適当な小さな確率で埋める。

NHA, NHB の効率はよいが平均の方向は、正定数  $C_1, C_2$  を用いて  $C_1 n + C_2 N/n^2$  の形で表わされ、 $N/n^2$  が大きくなるという条件の下で  $O(n)$  を達成できる。D1 は逆に、 $n^2/N$  が大きくなる条件の下で  $O(n)$  を達成するのである。ところで  $n^2/N$  が大きくなるならば、方針 A に戻って、衝突の期待数が小さいのであるから、整順列とすべきでよい。簡単でよい方法とすることはできる。

### 3. S-sort.

有限の数列  $x_1, \dots, x_n$  が与えられたとき、これを以下の算法で整順列とする。結果の整順列が等差数列に近いとき、効率がよい。これは慶応大学、島田規人・大野義夫により

開発研究された。

### 算法 S-sort.

1. 区間  $[\min x_i, \max x_i]$  を  $n$  等分する.  $C(1) := 0, \dots, C(n) := 0$  とする.
2.  $j = 1, 2, \dots, n$  にたいして,
  - 2.1  $k(j)$  を求める.  $k(j)$  の定義は  $x_j$  が第  $k(j)$  小区間に入る. ( $1 \leq k(j) \leq n$ )
  - 2.2  $C(k(j)) = C(k(j)) + 1$ . 小区間に入る  $x_i$  の数の計数.
3.  $D(j) = \sum_{i=j}^n C(i)$ . 第  $j$  -  $n$  区間の  $x_i$  の数.
4.  $j = 1, 2, \dots, n$  にたいして
  - 4.1  $y(D(k(j))) := x_j$ ;  $D(k(j)) := D(k(j)) - 1$ .
5.  $y(1), \dots, y(n)$  を単純挿入法により整順列とする.

この算法では  $x_i$  のための配列だけでなく,  $C(i)$ ,  $y(i)$ , ついでに  $k(i)$  のための4つの配列を用いる. 第4段の結果, 各小区間相互の間では大きさの順に並んでおり, 小区間内の整順列化さえ行えばよい. したがって  $C(i)$  の中に大きさの数があつては単純挿入法で早く整順列にできる.

4. 素朴な方法の見直し.

方針 A 2:  $S$ -sort を採用することにより,  $n^2/N$  が小さくなるように方向が  $O(n)$  の乱数生成法を作り, NHA または NHB を補完する. 要順列化により衝突が発見されたときの処置により 2 つの版 NVN1 と NVN2 を開発した. NVN1 では単純挿入法の途中で衝突を見つけたりはば, そこで新しい乱数を生成する. NVN2 では単純挿入の途中で衝突数を数えるに止め, 所とから新しい乱数  $\varepsilon$ , 衝突していることを確かめながら生成し, 併合法 (merge) により一つにまとめる. 衝突回数が多いと前者の方が効率が良いが, 差は小さく, 後者の方がプログラムが短い. ここでは NVN2 の概略を述べる.

### 算法 NVN2.

1.  $[1, N]$  一様乱数を  $n$  個  $x_1, \dots, x_n$ , 生成する.
2.  $x_1, \dots, x_n$  を  $S$ -sort する. ただし小区間  $[1, N]$  の  $n$  等分とする (整数区間であるから近似似的等分である.)  
単純挿入による整順列化の過程で衝突数  $k$  を数える.
3.  $l=1, \dots, k$  にたいして  $[1, N]$  一様乱数  $y_l$  を生成し, 単純挿入により整順列化し,  $y_1, \dots, y_{l-1}$  または  $x_1, \dots, x_n$  と衝突するものは再生成する.
4.  $x_1, \dots, x_n$  の整順列と,  $y_1, \dots, y_k$  の整順列とを併合する. そのとき  $x_1, \dots, x_n$  の  $y$  の重複を除く.

この算法ではある小区間に落ちる乱数の数が  $\sqrt{n}$  に超えなく、その部分を単純挿入法により整順列とするのに  $n$  に超える時間と要する。しかし各小区間に落ちる乱数の数の期待値は 1 であるので、時間は  $O(n)$  である。これを次節で示す。

### 5. $\delta$ -sort の評価.

もしも  $T$ - $\lambda$  が一様分布からの大きさ  $n$  の順序標本であるとき、小区間に落ちる  $T$ - $\lambda$  数の期待値は  $\sqrt{n}$  よりずっと小さく、せいぜい  $\log n$  に比例する大きさである。

その証明の概略は次の通りである: 問題は対称な多項分布でパラメータが  $(n; 1/n, \dots, 1/n)$  のとき、つまり

$$\Pr[(W_1, \dots, W_n) = (w_1, \dots, w_n)] = n! / n^n \prod_{j=1}^n w_j!$$

のときに  $\max(W_1, \dots, W_n)$  の期待値を評価することである。

$n$  が大きいとき  $(W_1, \dots, W_n)$  は漸近的に独立で、平均 1 のポアソン分布に従うから  $\max(W_1, \dots, W_n) = W_{(n)}$  の分布関数は漸近的に

$$\begin{aligned} \Pr[W_{(n)} \leq w] &= \left( \sum_{s=0}^w e^{-1}/s! \right)^n \\ &= \left( \frac{1}{\Gamma(w+1)} \int_0^1 t^w e^{-t} dt \right)^n \quad (*) \end{aligned}$$

に従う。  $n \rightarrow \infty$  のときこの順序分布は正整数値の上に置



化することを知っている。(C.W. Anderson 1970). その  
 2乗数値は  $P_n[W_j \leq w]$  の上側確率  $1/n$  前後の2点であ  
 る. (\*) のガンマ分布を正規近似し, 正規分布の上側確率  $\alpha$   
 の点を  $u_\alpha$  とすると,

$$x - u_{1/n} \sqrt{x} = 1.$$

これを解いて

$$\begin{aligned} x &= u_{1/n}^2 \left( \left( \sqrt{1 + 4/u_{1/n}^2} + 1 \right) / 2 \right)^2 \\ &\doteq u_{1/n}^2 \left( 1 + 1/u_{1/n}^2 \right) \end{aligned}$$

$n$  を大きくしたときの  $u_{1/n}^2$  の増加は  $\log n$  の位より少  
 ない。

坪田孝天教授より教示いただいた Dobosiewicz (1978),  
 中野智明, 他 (1982) は quick-sort の変形を, 3分法と  
 以下にやはり等分割を考えるもので, S-sort に類似してい  
 る.

## 参考文献

- [1] Knuth, D. E. (1981) The Art of Computer Programming,  
 Vol 2, 2nd ed., Addison-Wesley, Problem 3.4.2-8 (読谷談,  
 2乗数値算法/乱数, サイエンス社)
- [2] Kawarasaki, J. and Sibuya, M. (1982) Random  
 numbers for simple random sampling without replacement,

Kerio Math. Sem. Rep. No. 7, 1-9.

- [3] Vitter, J.F. (1982) Faster methods for random sampling, Tech. Rep. CS-82-21, Brown University.
- [4] Anderson, C.W. (1970) Extreme value theory for a class of discrete distributions with applications to some stochastic processes, J. Appl. Probability, 7, 99-113.
- [5] Dobosiewicz, W. (1978) Sorting by distributive partitioning, Information Processing Letters, 7, 1-6.
- [6] 中野智明, 大久保英嗣, 津田孝夫 (1981) 区間分割によるソート法について, 情報処理学会全国大会 第23回 (後期) 報告書 3K-1.