数理解析研究所講究録
第 498 巻 1983 年 39-48

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でたらめに抜き出す場合となると、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 $\{1, 2, \cdots, N\}$ からの大きさ 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。話の要点は、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=固定観念となっているが、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。

必要な乱数は，n 次元正整数値確率変数 $\underline{X}=(X_1,\cdots,X_n)$，

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

の実現値とみなされるものである．$n\le N/2$ として一般性を失わない．そうでなければ選ばれない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，整順列化し，同じ数が複数個現われたならば，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手続きが問題となる．衝突数の期待値は，古典的占有分布から求められ，$N(1-1/N)^n-N+n \doteqdot n^2/2N$ である．

方針B．　確率変数 $(Z_1,\cdots,Z_N)$ を上記の $\underline{X}$ を用いて次のように定義する．$X_\alpha=j$ となる $\alpha\ (1\le\alpha\le n)$ が存在すれば $Z_j=1$，さもなければ $Z_j=0$．そうすると，

$$\Pr[Z_j=1\mid \textstyle\sum_{i=1}^{j-1}Z_i=k]=(n-k)/(N-j+1),$$
$$0\le k\le \min(j-1,n),\quad j=1,2,\cdots,N.$$

これは，Nが大きくないとき，$n/N \fallingdotseq 1/2$ のときには有効な方法である．N回の手間を要するが，1回の仕事は(0, 1)一様乱数を生成し $(n-k)/(N-j+1)$ と比較するだけである．

方針C． $(X_1-1,\ X_2-X_1-1,\ \cdots,\ X_j-X_{j-1}-1,\ \cdots,\ X_n-X_{n-1}-1)$ の実現値とみなされる乱数を生成する．

$X = X_1 - 1$ の確率分布は，

$$p(x; N, x) := \Pr[X = X_1 - 1 = x] = \binom{N-x-1}{n-1} \Big/ \binom{N}{n}$$
$$= \binom{-1}{x}\binom{-n}{N-n-x} \Big/ \binom{-n-1}{N-n} = \frac{n(N-x-1)^{(n-1)}}{N^{(n)}} = \frac{n}{N}\frac{(N-n)^{(x)}}{(N-1)^{(x)}},$$
$$0 \leq x \leq N-n.$$

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，

$$\Pr[X_j - x_{j-1} - 1 \mid X_{j-1} = x_{j-1}] = p(x; N - x_{j-1},\ n-j+1)$$
$$1 \leq j \leq n \qquad (\text{ただし}\ x_0 = 0),$$

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

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

という性質をもち，$\lambda \fallingdotseq n/N$ のときには幾何分布

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

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が他と異質で簡単であるので，最初に紹介する。

算法D1

1. (0,1)一様乱数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)一様乱数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，$(X_1,\cdots,X_n)$を生成するための手間は$O(n)$であると論じている。これは$O(n)$の可能性を示すにはうまい方法であるが，$X$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

にかぞえているので，実際にはあまり遅くない。

NHB と D2 は $r=(n-1)/(N-1)$ の幾何乱数の棄却法で類似しているが，NHB の方が squeeze を徹底しており，算法が少し複雑になるが効率が良い。NHA はさらに効率を良くするために $r=n/N$ 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，$X \geq C+1$ の条件の下での 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 の効率は良いが平均の手間は，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 が大きくないという条件の下で $O(n)$ を達成できる。D1 は逆で，n^2/N が大きくない条件の下で $O(n)$ を達するのである。ところで n^2/N が大きくないならば，方針Aに戻って，衝突の期待数が小さいのであるから，整順列化さえうまくできれば，簡単で良い方法となるはずである。

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に効率が良い。これは慶応大学，島田禎人・大野義夫により

開発研究された.

<u>算法 S-sort.</u>

1. 区間 [$\min x_i$, $\max x_i$] を n 等分する. $C(1) := 0, \ldots, C(n) := 0$ とする.

2. $j = 1, 2, \ldots, n$ 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. $j = 1, 2, \ldots, n$ にたいして

 4.1 $y(D(k(j))) := x_j$; $D(k(j)) := D(k(j)) - 1$.

5. $y(1), \ldots, y(n)$ 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 $C(i)$ の中に大きな数がなければ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でS-sortを採用することにより，n^2/Nが大きくないときに手間が$O(n)$の乱数生成法を作り，NHAまたはNHBを補完する．整順列化によって衝突が発見されたときの処置によって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ならば，そこで新しい乱数を生成する．NVN2では単純挿入の途中では衝突数を数えるに止め，あとから新しい乱数を，衝突しない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．衝突回数が多くなると前者の方が効率がよいが，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 $x_1, \cdots, x_n$，生成する．
2. $x_1, \cdots, x_n$ をS-sortする．ただし小区間は［1，N］のn等分とする（整数区間であるから近似的な等分である．）単純挿入による整順列化の過程で衝突数kを数える．
3. $l=1, \cdots, k$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，$y_1, \cdots, y_{l-1}$または$x_1, \cdots, x_n$と衝突すれば再生成する．
4. $x_1, \cdots, x_n$の整順列と，$y_1, \cdots, y_k$の整順列とを併合する．そのとき$x_1, \cdots, x_n$の中の重複を除く．

この算法である小区間に落ちる乱数の数が $\sqrt{n}$ を越えると，その部分を単純挿入法により整順列とするのに n を越える手間とする．しかし各区間に落ちる乱数の数の期待値は1であるので，手間は $O(n)$ ですむ．これを次節で示す．

5. S-sort の評価．

もしもデータが一様分布からの大きさ n の確率標本であると，小区間に落ちるデータ数の期待値は $\sqrt{n}$ よりずっと小さく，だいたい $\log n$ 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 $(n; 1/n, \cdots, 1/n)$ のとき，つまり

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

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

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

$$\Pr[W_{(n)} \leq w] = \left(\textstyle\sum_{s=0}^{w} e^{-1}/s!\right)^n$$

$$= \left(\frac{1}{\Gamma(w+1)}\int_0^1 t^w e^{-t}\,dt\right)^n \qquad (*)$$

に従う．$n \to \infty$ のときこの確率分布は2整数値の上に退

化することが知られている．（C. W. Anderson 1970）．その分位数値は $\Pr[W_j \leq w]$ 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 α の点を u_α とすると，

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

これを解いて

$$x = u_{1/n}^2 \left(\left(\sqrt{1 + 4/u_{1/n}^2} + 1 \right)/2 \right)^2$$
$$\fallingdotseq u_{1/n}^2 \left(1 + 1/u_{1/n}^2 \right)$$

n を大きくしたときの $u_{1/n}^2$ の増加は $\log n$ の位より少しだけ早い．

河田孝夫教授より教示いただいた Dobosiewicz (1978)，中野裕明，他 (1982) は quick-sort の変形で，二分化のときにやはり等分割を考えるもので，S-sort に類似している．

文献

[1] Knuth, D. E. (1981) The Art of Computer Programming, Vol 2, 2nd ed., Addison-Wesley, Problem 3.4.2—8 (渋谷訳，準数値算法／乱数，サイエンス社)

[2] Kawarasaki, J. and Sibuya, M. (1982) Random numbers for simple random sampling without replacement,

Keio 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.