モンテカルロ法の数学的定式化*

杉田 洋

大阪大学大学院理学研究科数学専攻

1 はじめに

モンテカルロ法が様々な分野で実際的成果を上げていることは認めても，多くの真摯な研究者はモンテカルロ法に対して次のような根本的な疑念を持っている．

> モンテカルロ法には乱数が必要である．しかしコンピュータプログラムでは乱数を生成することはできない．そこで乱数の代わりに(コンピュータプログラムによって生成される)疑似乱数を用いてモンテカルロ法を実行するが，こうした便宜的方法は，当然，数学的正当性を持ち得ない．

じつは，この疑念はモンテカルロ法の直感的な解釈による素朴な思い込みから生じた誤解に過ぎない．小論では，モンテカルロ法を数学的に適切に定式化することによって，この疑念の解消を目指す．

乱数と疑似乱数という概念についてはすでに明確な定義が知られている．まず，乱数という概念は1960年代にコルモゴロフ(Kolmogorov)らが“計算の複雑さ”という考えを用いて定義した([3, 6, 8])．そして疑似乱数という概念は1980年代に計算機科学とくに暗号理論の文脈でブラム(Blum)らが定義した([1, 2, 18])．しかし残念ながら，それらの定義は今までモンテカルロ法の理論と実践に活かされることはなかった．それはそれらの定義が不適切だったからではなく，むしろモンテカルロ法それ自体に対する従来の捉え方が不適切だったからである．ここでは，それら乱数や疑似乱数の定義を拠り所として，それらと理論的整合性を持つようなモンテカルロ法の数学的定式化を提案する．

我々の定式化の下で結論として次のことが見えてくる．

> モンテカルロ法には必ずしも乱数は必要でなく疑似乱数で十分かもしれない．実際，モンテカルロ積分の場合には，乱数と同じ働きをする疑似乱数が存在し，大きすぎない規模のモンテカルロ積分においてすでに実用化されている．

---

* ver.20121122．小論は [14] の翻訳抜粋．

モンテカルロ積分とは，大数の法則を利用した数理統計的手法による数値積分法のことである．モンテカルロ法の重要な応用の大部分はモンテカルロ積分であるから，我々の結論にはとても大きな意味がある．

なお，小論は手短に説明することを念頭において書いたので，随所に数学的に厳密でない記述がある．詳細は [14] を参照せよ．

2 概要

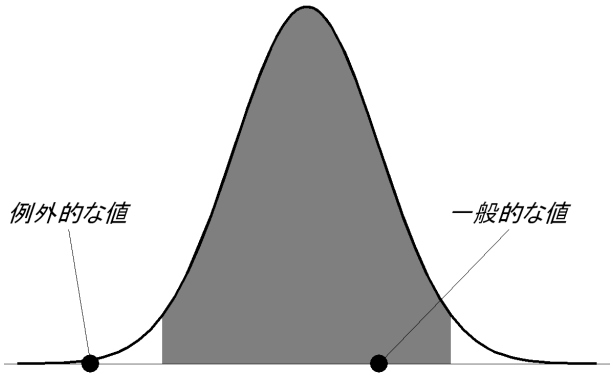
最初に概要を述べる．後の節でそれぞれの事項を掘り下げて説明する．

モンテカルロ法 (Monte-Carlo method) とは，確率変数のサンプリングをコンピュータを用いて行うことによって数学的問題を (主として数理統計学における意味で) 数値的に解く手法をいう．モンテカルロ法では，それぞれの具体的な問題に応じて確率空間 $(\Omega, \mathcal{F}, P)$ — ここでは有限回の硬貨投げの確率空間

$$(\{0,1\}^L,\ 2^{\{0,1\}^L},\ P_L), \quad L \geq 1,$$

とする — と，その上で定義された確率変数 $S : \{0,1\}^L \to \mathbb{R}$ を設定する．そして，一つの $\omega \in \{0,1\}^L$ を選んで S の ω における値 $S(\omega)$ を算出する．以下，これを**サンプリング** (sampling，標本抽出) と呼ぶ．

図 1: S の分布と一般的な値 (概念図)



モンテカルロ法は，その名の由来のとおり，**賭け**の一種である．プレーヤー (以下，アリスと呼ぶ) の目的は S の**一般的な値** — S のとる値としてごくありふれた値，例外的でない値 — をサンプリングすることである (§ 3.1)．アリスは S の例外的な値をサンプリングしてしまうリスクを承知の上で**自分の意思で** ω を選ぶ．彼女のリスクは

$$A := \left\{ \omega \in \{0,1\}^L \,\middle|\, S(\omega) \text{ は } S \text{ の例外的な値} \right\}$$

の確率 $P_L(A)$ でもって評価される．

S が例外値をとることは滅多にないから $P_L(A) \ll 1$ である．[†1] 当然，アリスは自分が S の一般的な値を手に入れることはほとんど確実だと思うだろう．L が小さいときは確かにそのとおりである．しかし，$L \gg 1$ (§ 3.2 の例題では $L = 10^8$) の場合は事情が異なる．その場合，アリスが一つの $\omega \in \{0,1\}^L$ を選ぶときの具体的な方法が問題となる．実際，$\omega \in \{0,1\}^L$ を指定するには，$L = 10^8$ ともなればそれは約 12MB のデータにもなるので,[†2]情報量の大きさからいってコンピュータを用いるよりない．しかしこれだけの情報量のデータをキーボードから直接打ち込むのはあまりに膨大な時間と労力がかかるので事実上不可能である．何らかの工夫が必要である．ところが，じつはどのような工夫をしようとも，アリスが (膨大な時間と労力をかけずに) 自分の意志で選ぶことのできる $\omega \in \{0,1\}^L$ は非常に少数であり，従って特殊なものに限られてしまう (§ 4)．そのため，$P_L(A) \ll 1$ であったとしても，アリスが S の一般的な値をほとんど確実に手に入れることができる，とはとてもいい切れないのである．このことから，リスク評価 $P_L(A)$ に実質的な意味を持たせるためには，$\{0,1\}^L$ の大部分を占める「アリスの意志では選ぶことのできないような ω」— そのような ω を**乱数**と呼ぶ — によるサンプリングが必要であると考えるのは自然である．[†3]

しかし一方，現実の問題で扱う S は何らかの意味のある量を表す確率変数であり，任意の確率変数ではない．すなわち，アリスが自分の意志で選ぶことのできる $\omega \in \{0,1\}^L$ は確かに特殊だが，S も確率変数 ($\{0,1\}^L$ 上の関数) 全体の中ではきわめて特殊なのである．もしかしたら，特殊な S なら特殊な ω によって — つまり乱数でなくても — 一般的な値をサンプリングすることができるかもしれない．それを実現しようと企てるのが疑似乱数生成器である．

疑似乱数生成器とは短い $\{0,1\}$-列を長い $\{0,1\}$-列に引き伸ばす写像のことである．たとえば疑似乱数生成器 $g : \{0,1\}^n \to \{0,1\}^L$，$n < L$，を用いたモンテカルロ法では，アリスは**種**と呼ばれる $\{0,1\}$-列 $\omega' \in \{0,1\}^n$ を自分の意志で選ぶ．(ここで任意の $\omega' \in \{0,1\}^n$ をコンピュータのキーボードから人が容易に打ち込める程度に n は小さい必要がある．) コンピュータは ω' をもとに疑似乱数 $g(\omega') \in \{0,1\}^L$ を算出し，さらにこれを S に代入して $S(g(\omega'))$ を出力する．

疑似乱数生成器 g を用いることによって，アリスは「$S(g(\omega'))$ が S の一般的な値であるかどうか」という別の新しい賭けを行うことになる．そのリスクは確率 $P_n(g(\omega') \in A)$ によって評価される．もちろん，この確率は g に依存するが，もし $P_n(g(\omega') \in A) \ll 1$ であるような g を見つけることができたなら，膨大な時間と労力をかけずに実際に高い確率でアリスは S の一般的な値を手に入れることができる．この場合，長大な乱数は必要でない．このような g を A **に対して安全な疑似乱数生成器**という (§ 5.2)．モンテカルロ法におけるサンプリングの問題はこのような安全な疑似乱数生成器を見出すことで解決される．

一般の確率変数 S については，その例外値を与える ω の集合 A に対して安全な疑似乱数生成器をどのように構成したらよいかは，その候補はいくらか挙がっている

[†1] $a \ll b$ は「a は b よりずっと小さい」の意，また $a \gg b$ は「a は b よりずっと大きい」の意．

[†2] 400 字詰原稿用紙で約 15,000 枚分の情報量．

[†3] そのため，物理乱数を用いる立場もあるが，この小論では以下に述べるように疑似乱数生成器による解決について論じる．

ものの，知られていない (§ 5.3)．しかし，疑似乱数生成器の用途を**モンテカルロ積分**に限った場合，すなわち S がある i.i.d. 確率変数列[†4]の標本平均であるような場合，S の例外値を与える ω の集合 A に対して安全な疑似乱数生成器が具体的に構成できる．そして大きすぎない規模のモンテカルロ積分の場合はすでに実用化されている (§ 6.2)．

3　賭けとしてのモンテカルロ法

数学の問題はもちろん確実な方法で解けるに越したことはない．しかし，非常に複雑な問題あるいは詳しい情報が不足しているような問題では確率的ゲーム，つまり賭け，として定式化し，正しい解が求まらないリスクを承知の上で解を推定することが実際的である．モンテカルロ法はまさにそのような場合の一つである．

3.1　プレーヤーの目的

数学的にはモンテカルロ法を賭けとして定式化するのが適切である．この賭けのプレーヤー，アリスの目的は与えられた確率変数の一般的な値— 典型的な，ごくありふれた，特殊でない，例外的でない値 — をサンプリングすることである．(それが問題解決に結び付くようにあらかじめ問題を設定しておく．詳しくは後のいくつかの例を参照せよ．) もちろん，不運にも一般的でない値，すなわち例外的な値をサンプリングしてしまうこともある．数学的には，そのような場合の起こる確率をできるだけ正しく見積もること — リスクの評価 — が要請される．

次に非常に小規模な (コンピュータを必要としない) モンテカルロ法の例を示す．

例 1　r を未知の整数とする．100 枚のカードがあり，そのうち 99 枚には r が，残りの 1 枚には $r+1$ が，書いてある．アリスがその中から 1 枚だけを選び，そのカードに書かれた数でもって r の値を推定する．アリスが不運にも r の値を正しく推定し損ねる確率は 1/100 である．

この例を賭けの形式で述べるならば以下のようになる:「アリスの目的は r の書かれたカード (今の場合の “一般的な値”) を選ぶことである．選んだカードの数が r ならばアリスの勝ち，$r+1$ ならばアリスの負け，である．このとき，アリスの負ける確率は 1/100 である (リスクの評価).」

なお，一般にモンテカルロ法ではプレーヤーの目的が達成されたかどうかはサンプリングの後でも分からないのが普通である．たとえば例 1 では，リスクは正確に評価されるものの，アリスの推定した r の値が正しいかどうか — つまり賭けとしての勝ち負け— は，カードを選んだ後も知ることはできない．[†5]

[†4] i.i.d. は独立同分布 (*i*ndependently *i*dentically *d*istributed) の意．

[†5] 人生における様々な選択も多くの場合，賭けであろう．果たして自分の選んだ手が良い手だったかどうか，結局分からないということはよくあるではないか．．．．

例外的な値を求めたい場合もないわけではない．たとえば，複雑な確率変数 X の最小値をモンテカルロ法で探索するという場合がある．これは X の滅多に実現しない値を求めることになる．このような場合，別の確率変数 S をうまく定義して，S の一般的な値が X のその滅多に起きない値になるようにする．次の例を見よ．

例 2　$\Pr(X < c) = 1/10000$ であると仮定しよう．すなわち，X は c 未満の値を取り得るがその確率はきわめて小さい．ここで $\{X_k\}_{k=1}^{40000}$ を X の独立なコピーとし，$S := \min_{1\le k\le 40000} X_k$ とする．このとき

$$\Pr(S < c) = 1 - \left(1 - \frac{1}{10000}\right)^{40000} \approx 1 - e^{-4} = 0.981\ldots$$

となる．[†6] だから，S は高い確率 0.98 で c 未満の値を取る．

3.2　一つの例題

例 1 を実行するのには，100 枚のカード以外に何も特別な道具は必要ない．しかし，現実のモンテカルロ法で扱う賭けは大規模であり，高い情報処理能力を持つコンピュータが必要である．ここでは次の例題をモンテカルロ法で解くことを中心に考えて行く．

例題　硬貨投げを 100 回行うとき表が続けて 6 回以上出る確率 p を求めよ．

モンテカルロ法では数理統計学における推定の方法を適用する．「硬貨投げを 100 回行う」という試行を独立に N 回繰り返して，そのうち「表が続けて 6 回以上出る」という事象が起った回数を S_N とする．このとき N が十分大きければ大数の法則により S_N/N の値が高い確率で求めたい p のよい推定値となる．より具体的には

例 3　$N := 10^6 = 1,000,000$ とする．$S_{10^6}/10^6$ の平均と分散は

$$\mathbf{E}\left[\frac{S_{10^6}}{10^6}\right] = p, \quad \mathbf{V}\left[\frac{S_{10^6}}{10^6}\right] = \frac{p(1-p)}{10^6} \le \frac{1}{4\cdot 10^6}$$

なので**チェビシェフの不等式** (Chebyshev's inequality) により

$$\Pr\left(\left|\frac{S_{10^6}}{10^6} - p\right| \ge \frac{1}{200}\right) \le \frac{1}{4\cdot 10^6}\cdot 200^2 = \frac{1}{100}, \tag{1}$$

が成り立つ．いい換えれば，$S_{10^6}/10^6$ の一般的な値をサンプリングできれば，それは p の近似値になっている．

[†6] $x \approx y$ は x と y の値がほぼ等しいの意．

例 3 では (1) によってリスクが評価されている，と考えてよいだろう．

もちろん，本物の硬貨を $100 \times 10^6 = 10^8 = 1$ 億回投げて S_{10^6} の値を計算するのではない．コンピュータを用いる．数学的形式に則って考えるために，例 3 の S_{10^6} を硬貨投げの確率空間 $(\Omega := \{0,1\}^{10^8}, 2^\Omega, P_{10^8})$ 上で次のように実現しよう: まず，関数 $X : \{0,1\}^{100} \to \{0,1\}$ を

$$X(\xi_1, \ldots, \xi_{100}) := \max_{1 \le n \le 100-5} \prod_{k=n}^{n+5} \xi_k, \qquad (\xi_1, \ldots, \xi_{100}) \in \{0,1\}^{100},$$

と定義する．これは $(\xi_1, \ldots, \xi_{100})$ の中で 1 が 6 個以上続いた箇所があるとき $X = 1$ そうでないとき $X = 0$ であることを意味する．次に $X_k : \{0,1\}^{10^8} \to \{0,1\}$，$k = 1, 2, \ldots, 10^6$，を

$$X_k(\omega) := X(\omega_{100(k-1)+1}, \ldots, \omega_{100k}), \quad \omega = (\omega_1, \ldots, \omega_{10^8}) \in \{0,1\}^{10^8},$$

と定義し，さらに $S_{10^6} : \{0,1\}^{10^8} \to \mathbb{Z}$ を

$$S_{10^6}(\omega) := \sum_{k=1}^{10^6} X_k(\omega), \quad \omega \in \{0,1\}^{10^8},$$

と定義する．S_{10^6} の例外値を与える ω の集合 A_0 を

$$A_0 := \left\{ \omega \in \{0,1\}^{10^8} \;\middle|\; \left| \frac{S_{10^6}(\omega)}{10^6} - p \right| \ge \frac{1}{200} \right\} \tag{2}$$

とすれば，(1) より $P_{10^8}(A_0) \le 1/100$ となる．そこで例 3 は次のような賭けとして捉えることができる．

例 4　一つの $\omega \in \{0,1\}^{10^8}$ をアリスが選んだとき，$\omega \notin A_0$ ならば勝ち，$\omega \in A_0$ ならば負け，という賭けを考える．この賭けでアリスが負ける確率は 1/100 以下である．

4　乱数の問題

例 1 と例 4 は規模の違いを除けば，数学上の違いはない．しかし，この規模の違いこそが実際に後者の賭けを行うときに本質的な問題を引き起こす．

今，アリスが例 4 の賭けを実行するために一つの $\omega \in \{0,1\}^{10^8}$ を選ぼうとしているとしよう．しかし，じつは $\{0,1\}^{10^8}$ の元のうちアリスが自分の意志で選ぶことができる元はきわめて少数であり，従って特殊な元なのである．だから，たとえリスク評価が $P_{10^8}(A_0) \ll 1$ であっても，アリスが実際にこの賭けに勝つことは必ずしも容易ではない．

事情を説明しよう．各々の $\omega \in \{0,1\}^{10^8}$ は $10^8 = 1$ 億ビット，すなわち約 12MB のデータであり，アリスがこれを選ぶにはコンピュータを用いるよりないが，それとて直接キーボードからコンピュータに入力することなどとてもできない．たとえば

アリスはせいぜい 1,000 ビットのデータ (アルファベット 125 字分) までならキーボードから直接コンピュータへ入力できると仮定しよう．コンピュータは彼女の入力をもとに $\{0,1\}^{10^8}$ の一つの元をあるアルゴリズムによって生成するとする．[†7] このとき，アリスが自分の意志で選ぶことのできる $\omega \in \{0,1\}^{10^8}$ の個数はせいぜい 2^{1001} に過ぎない．(なぜなら l-ビットのデータ全体の個数が 2^l で従って l-ビット以下のデータ全体の個数が $2^0+2^1+2^2+\cdots+2^l = 2^{l+1}-1$ だから．) $\{0,1\}^{10^8}$ の元の個数は 2^{10^8} もあるのだから，アリスの選ぶことのできる $\omega \in \{0,1\}^{10^8}$ がいかに僅かであるかが分かるだろう．たとえ，アリスが 10^8-10 ビットのデータまで入力できると仮定しても，アリスの選ぶことのできる ω の個数はせいぜい 2^{10^8-9} であり，これでも $\{0,1\}^{10^8}$ 全体の個数 2^{10^8} の 1/512 に過ぎない．いい換えると $\{0,1\}^{10^8}$ 全体の少なくとも 511/512 は 10^8-9 ビット以上の入力がどうしても必要な $\{0,1\}$-列なのである．一方で言うまでもなく，任意の $\{0,1\}^{10^8}$ の元は「それ自身を入力する」という方法で 10^8 ビットの入力で指定することができる．

短い入力で指定できる $\{0,1\}$-列は，その列に何らかの規則性があるからそうできる，と考えられる．逆に規則性がないと指定するために長い入力が必要になるはずである．そこで，それ自身の長さと同程度の長さの入力がなければ指定できない $\{0,1\}$-列を**乱数**と呼ぶ．[†8] 乱数を指定するには，結局，それ自身を入力するより簡潔な方法は存在しない．$L \gg 1$ のとき $\{0,1\}^L$ の元のうち圧倒的多数は乱数である．

例 4 の賭けのリスク評価 $P_{10^8}(A_0) \leq 1/100$ は，どの $\omega \in \{0,1\}^{10^8}$ も同様に確からしく選ぶことができる，ということを前提にしている．だから，$P_{10^8}(A_0) \leq 1/100$ に実質的な意味を持たせるためには，ω は主として圧倒的多数を占める乱数たちから選ばれるべきであろう．これがモンテカルロ法に乱数が必要だといわれる理由である．しかし，乱数は，それほど多いにもかかわらず，指定することがあまりに大変なために，事実上，アリスはそのうち一つさえ選ぶことができない．この不条理こそ，大規模なモンテカルロ法におけるサンプリングの本質的な困難なのである．

5 疑似乱数生成器

疑似乱数生成器は乱数を用いずに確率変数 S の一般的な値を高い確率でサンプリングするための道具である．その持つべき性質，すなわち安全性，について述べる．

5.1 定義と役割

例 4 の賭けを実行するとき，$S_{10^6}(\omega)$ を計算するためにアリスはとにかく一つの $\omega \in \{0,1\}^{10^8}$ を選ばなければならない．それには何か道具が必要である．ここでは最もよく用いられる道具，すなわち疑似乱数生成器を用いる場合を考えよう．

[†7] じつは，そうしたアルゴリズムこそが後に述べる疑似乱数生成器と呼ばれるものである．

[†8] 長い $\{0,1\}$-列を短い入力で指定できるとすると，その短い入力は元の長いデータを “圧縮したもの” ということができる．この意味で乱数とは “圧縮不可能なデータ” である．

定義 5　$n < L$ のとき，関数 $g : \{0,1\}^n \to \{0,1\}^L$ を**疑似乱数生成器** (pseudo-random generator) という．ここで g の入力 $\omega' \in \{0,1\}^n$ を**種** (seed)[†9]，出力 $g(\omega') \in \{0,1\}^L$ を**疑似乱数** (pseudo-random number) という．

疑似乱数というのは数列であるが，数学的対象としては，それを生み出す関数の方が重要である．それが疑似乱数生成器である．疑似乱数生成器は**初期化**[†10]という手続きを経て疑似乱数を生成する．初期化というのは種 $\omega' \in \{0,1\}^n$ を一つ選ぶことである．プログラムはそれをもとに疑似乱数 $g(\omega') \in \{0,1\}^L$ を生成する．実用のためには，当然，$n \geq 1$ は種 ω' がキーボードから入力可能な程度に小さいことが必要である．また，関数 g を実現するプログラムが許容できる程度に短く，早く動作することが必要である．

例 6　例 4 において，アリスがたとえばある疑似乱数生成器 $g : \{0,1\}^{238} \to \{0,1\}^{10^8}$ を使うとしよう.[†11] アリスは g の種 $\omega' \in \{0,1\}^{238}$ を一つ選んでキーボードからコンピュータに入力する．ω' は 238 ビット (アルファベット 40 文字分) のデータだから，キーボードから入力するのは困難ではない．そこでコンピュータは $S_{10^6}(g(\omega'))$ を計算する．

疑似乱数生成器を用いる理由は，アリスの入力すべきデータ $\omega \in \{0,1\}^{10^8}$ がキーボードから入力するにはあまりにも長大だからである．入力データが短くて済む場合は疑似乱数生成器は必要ない．たとえば例 1 で 100 枚のカードの中から 1 枚選ぶとき，誰が疑似乱数生成器の利用を考えるだろうか．

5.2　安全性

例 6 の場合を引き続き考える．アリスは疑似乱数生成器 g の種 $\omega' \in \{0,1\}^{238}$ を自由に選ぶことができる．今の場合，リスクは確率

$$P_{238}\left(\left|\frac{S_{10^6}(g(\omega'))}{10^6} - p\right| \geq \frac{1}{200}\right) \tag{3}$$

であり，次にこれを計算する必要がある．もちろん，確率 (3) は g に依存する．もし，この確率 — すなわちアリスの選んだ ω' から計算された $S_{10^6}(g(\omega'))$ が S の例外的な値である確率 — が大きいとすると，アリスは目的を達成することが困難になる．それでは困る．

そこで，次の (少し曖昧な) 定義を設けよう; 疑似乱数生成器 $g : \{0,1\}^n \to \{0,1\}^L$, $n < L$, が集合 $A \subset \{0,1\}^L$ に対して**安全** (secure) であるとは

$$P_L(\omega \in A) \approx P_n(g(\omega') \in A)$$

[†9] 初期値ともいう．
[†10] ランダマイズ (randomize) ともいう．
[†11] 238 という半端な数の由来は例 9 で明らかになる．

が成り立つことをいう．

例 6 において，もし $g : \{0,1\}^{238} \to \{0,1\}^{10^8}$ が (2) で定義された集合 A_0 に対して安全であれば，アリスが自分の意志で選ぶことのできる $\omega' \in \{0,1\}^{238}$ の大多数に対して $S(g(\omega'))$ は S の一般的な値を与えることが分かる．この場合，長大な乱数は必要でない．いい換えると，S の一般的な値をサンプリングしたい際に g を用いてもリスクが大きくならないという意味で，このような g を安全な疑似乱数生成器と呼ぶわけである．モンテカルロ法におけるサンプリングの問題 — すなわち乱数の問題 — はこのような安全な疑似乱数生成器を見出すことで解決される．

一般に，できるだけ多くの集合 A に対して安全であるような g が望ましい疑似乱数生成器といえる．しかしすべての A に対して安全であるような疑似乱数生成器は存在しない．実際，疑似乱数生成器 $g : \{0,1\}^n \to \{0,1\}^L$ が与えられたとき

$$A_g := g(\{0,1\}^n) \subset \{0,1\}^L$$

とすれば，$P_L(\omega \in A_g) \leq 2^{n-L}$ であるが，$P_n(g(\omega') \in A_g) = 1$ となるので g は A_g に対して安全ではない．従って，疑似乱数生成器の安全性を論じるときは集合 A のクラスを制限して考えなければならない．

5.3 計算量的に安全な疑似乱数生成器

疑似乱数生成器の安全性を論じる上で，対象となる集合の最も大きなクラスは，以下に述べる「計算量的に判定可能であるような集合」のクラスであると考えられる．

疑似乱数生成器 $g : \{0,1\}^n \to \{0,1\}^L$，$n < L$，が集合 $A \subset \{0,1\}^L$ に対して安全であるかどうかを判定する具体的手続きについて考えよう．そのためには，何はさておき，与えられた $\omega \in \{0,1\}^L$ が $\omega \in A$ を満たすかどうかを判定するプログラムを書かなければならない．そもそも部分集合 $A \subset \{0,1\}^L$ の総数は 2^{2^L} だから，そのようなプログラムも同じ数だけ必要である．このときプログラムの長さは長いものでは 2^L ビットに達することが § 4 で述べた議論と同様にして分かる．さらに，大部分の A のプログラムがほぼ 2^L ビットの長さに達することも分かる．$L = 10^8$ の場合だと，大部分の $A \subset \{0,1\}^{10^8}$ のプログラムの長さはほぼ 2^{10^8} ビットである．

明らかに，そのように長いプログラムが必要となる A に対しては $\omega \in A$ であるかどうか実際には判定できない．そこで，$\omega \in A$ が計算量的に判定可能であるような A に対してのみ安全であるような疑似乱数生成器があればそれで十分である．そのような疑似乱数生成器は**計算量的に安全** (computationally secure，または**暗号理論的に安全** (cryptographically secure)) である，といわれる．ただし，暗号理論において，疑似乱数生成器の計算量的安全性は，ここに述べた空間計算量 (プログラムの長さ) ではなく，時間計算量 (プログラムの実行時間) をもとに定式化されている．[†12]

例 6 の場合に g がもし計算量的に安全であれば，$S_{10^6}(\omega)$ と $S_{10^6}(g(\omega'))$ の分布は十分近い．実際，S_{10^6} という関数が実際に計算できる以上，たとえば各 $c_1, c_2 \in \mathbb{Z}$ に対

[†12] 短いプログラムでも反復計算が多ければ実行時間が実行不可能なほど莫大になり得るから，時間計算量を基準にすることはより実際的な意味がある．

して $A(c_1, c_2) := \{\omega | c_1 \le S_{10^6}(\omega) \le c_2\}$ とすれば，$\omega \in A(c_1, c_2)$ は計算量的に判定可能である．従って p' の値が何であっても

$$P_{10^8}\left(\left|\frac{S_{10^6}(\omega)}{10^6} - p'\right| \ge \frac{1}{200}\right) \approx P_{238}\left(\left|\frac{S_{10^6}(g(\omega'))}{10^6} - p'\right| \ge \frac{1}{200}\right)$$

が成り立つ．

計算量的に安全な疑似乱数生成器は，理論上，最も汎用的で最も完全な疑似乱数生成器といえよう．ただし，その存在は計算量理論の難しい問題の一つであり現時点では不明である．また，この概念は正確に述べればある種の漸近的性質であって，個々の具体的な問題に対して，計算量的に安全な疑似乱数生成器が確実に有用であることを保証するものではない．

6 モンテカルロ積分

X を m 回の硬貨投げの関数，すなわち $\{0,1\}^m$ 上の関数とし，X の平均

$$\mathbf{E}[X] = \frac{1}{2^m}\sum_{\xi\in\{0,1\}^m} X(\xi)$$

を求める問題を考える．ここで m が小さいときは上式の右辺を直接計算すればよいが，m が大きいとき (たとえば $m = 100$)，計算量が莫大になりそれは事実上不可能になる．**モンテカルロ積分** (Monte-Carlo integration) とは，そのような場合に，大数の法則を用いて確率変数の平均を推定する手法をいう (例 3)．およそ科学的なモンテカルロ法では，確率変数の分布に関する何らかの特性量を計算することを目的としていて，それらはほとんどの場合，モンテカルロ積分である．

6.1 i.i.d.-サンプリング

例 3 を一般的な設定の下で述べると以下のようになる．X の独立な N 個のコピー $\{X_k\}_{k=1}^N$ の和を S_N とする．S_N は Nm 回の硬貨投げの関数であるが，具体的に書けば

$$X_k(\omega) := X(\omega_k), \quad \omega_k \in \{0,1\}^m, \quad \omega = (\omega_1, \ldots, \omega_N) \in \{0,1\}^{Nm}, \tag{4}$$

$$S_N(\omega) := \sum_{k=1}^N X_k(\omega). \tag{5}$$

このとき S_N/N と X の平均 (それぞれ P_{Nm} と P_m による積分) は等しく ($\mathbf{E}[S_N/N] = \mathbf{E}[X]$)，分散は $\mathbf{V}[S_N/N] = \mathbf{V}[X]/N$ を満たす．S_N/N をサンプリングすることににによって X の平均を推定する方法を **i.i.d.-サンプリング**[†13] と呼ぶ．このときのリスクをチェビシェフの不等式

$$P_{Nm}\left(\left|\frac{S_N(\omega)}{N} - \mathbf{E}[X]\right| \ge \delta\right) \le \frac{\mathbf{V}[X]}{N\delta^2} \tag{6}$$

[†13] $\{X_k\}_{k=1}^N$ が i.i.d. 確率変数列なのでそう呼ぶ．

で評価しよう.[†14] このことは

$$A_1 := \left\{ \omega \in \{0,1\}^{Nm} \;\middle|\; \left| \frac{S_N(\omega)}{N} - \mathbf{E}[X] \right| \geq \delta \right\} \tag{7}$$

としたとき，$\omega \in \{0,1\}^{Nm}$ を選んで $\omega \notin A_1$ ならば勝ち，$\omega \in A_1$ ならば負け，という賭けを考えていることになる.

6.2 ランダム-ワイル-サンプリング

モンテカルロ積分の場合，サンプリングの対象となる確率変数は S_N であり，確率変数の中でも非常に特殊な形をしている．その事実を利用すれば A_1 に対し安全な疑似乱数生成器を具体的に構成することができる.

それを述べるために記号を導入する； 各 $m \geq 1$ に対して

$$D_m := \{ i2^{-m} \mid i = 0, \ldots, 2^m - 1 \} \subset \mathbb{T}^1, \tag{8}$$

とする．集合族 $\mathcal{I}_m := \{ [a,b) \mid a,b \in D_m \}$ を含むような最小の (完全) 加法族を $\mathcal{B}_m$ と書く．すなわち，$\mathcal{B}_m$ の元は $\mathcal{I}_m$ の有限個の元の和集合である．D_m 上の一様確率測度を $P_{(m)}$ で表す．最後に各 $x \in \mathbb{T}^1$ に対して

$$\lfloor x \rfloor_m := \lfloor 2^m x \rfloor / 2^m \in D_m$$

とおく.

定義 7 (cf. [10, 16]) $j \in \mathbb{N}^+$ とし

$$Z_k(\omega') := \lfloor x + k\alpha \rfloor_m \in D_m, \quad \omega' = (x,\alpha) \in D_{m+j} \times D_{m+j}, \quad k = 1, 2, 3, \ldots, 2^{j+1},$$

を定義する．これを用いて疑似乱数生成器 $g : \{0,1\}^{2m+2j} \to \{0,1\}^{Nm}$，ただし $N \leq 2^{j+1}$，を

$$\begin{aligned} g(\omega') &:= (Z_1(\omega'), Z_2(\omega'), \ldots, Z_N(\omega')) \in D_m^N \cong \{0,1\}^{Nm}, \\ \omega' &= (x,\alpha) \in D_{m+j} \times D_{m+j} \cong \{0,1\}^{2m+2j} \end{aligned} \tag{9}$$

と定義する．この疑似乱数生成器を用いる数値積分法を**ランダム-ワイル-サンプリング** (random Weyl sampling，以下 RWS と略す) と呼ぶ.[†15]

定理 8 (9) の疑似乱数生成器 $g : \{0,1\}^{2m+2j} \to \{0,1\}^{Nm}$ は (5) の S_N に対して

$$\begin{aligned} \mathbf{E}[S_N(g(\omega'))] &= \mathbf{E}[S_N(\omega)] \,(= N\mathbf{E}[X]), \\ \mathbf{V}[S_N(g(\omega'))] &= \mathbf{V}[S_N(\omega)] \,(= N\mathbf{V}[X]), \end{aligned}$$

[†14] $\mathbf{E}[X]$ が未知であるのと同様に $\mathbf{V}[X]$ も未知であるのが普通だろう．その意味でこのリスク評価は完全ではない．しかし状況によっては $\mathbf{V}[X]$ の上からの評価が得られれば (たとえば 例 3 のように，X が有界の場合など)，このリスク評価は完全になる.

[†15] [10, 16] で紹介されているランダム-ワイル-サンプリングより，わずかばかり改良されている.

を満たす．ただし ω'，ω はそれぞれ P_{2m+2j}，P_{Nm} に従うものとする．これより (6) と同様のチェビシェフの不等式

$$P_{2m+2j}(g(\omega') \in A_1) \;=\; P_{2m+2j}\left(\left|\frac{S_N(g(\omega'))}{N} - \mathbf{E}[X]\right| \geq \delta\right) \;\leq\; \frac{\mathbf{V}[X]}{N\delta^2}$$

が成り立つ．この意味で g は (7) の A_1 に対して安全な疑似乱数生成器である．

証明. *Step 1.* (cf. [10, 16]) $D_{m+j} \times D_{m+j}$ 上の一様 (直積) 確率測度 $P_{(m+j)} \otimes P_{(m+j)}$ の下で $\{Z_k\}_{k=1}^{2^{j+1}}$ はペアごとに独立であり，各 Z_k は D_m で一様分布することを示そう．そのためには $F, G : \mathbb{T}^1 \to \mathbb{R}$ が $\mathcal{B}_m$-可測のとき，$1 \leq k < k' \leq 2^{j+1}$ に対して

$$\mathbf{E}[F(Z_{k'})G(Z_k)] = \int_0^1 F(t)dt \int_0^1 G(s)ds \tag{10}$$

となることを示せばよい．ここに $\mathbf{E}$ は $P_{(m+j)} \otimes P_{(m+j)}$ による平均値を表す．

平均の定義と F, G が $\mathcal{B}_m$-可測であることより

$$\begin{aligned}
\mathbf{E}[F(Z_{k'})G(Z_k)] &= \frac{1}{2^{2m+2j}} \sum_{q=1}^{2^{m+j}} \sum_{p=1}^{2^{m+j}} F\left(\frac{p}{2^{m+j}} + \frac{k'q}{2^{m+j}}\right) G\left(\frac{p}{2^{m+j}} + \frac{kq}{2^{m+j}}\right) \\
&= \frac{1}{2^{2m+2j}} \sum_{q=1}^{2^{m+j}} \sum_{p=1+kq}^{2^{m+j}+kq} F\left(\frac{p}{2^{m+j}} + \frac{(k'-k)q}{2^{m+j}}\right) G\left(\frac{p}{2^{m+j}}\right) \\
&= \frac{1}{2^{2m+2j}} \sum_{q=1}^{2^{m+j}} \sum_{p=1}^{2^{m+j}} F\left(\frac{p}{2^{m+j}} + \frac{(k'-k)q}{2^{m+j}}\right) G\left(\frac{p}{2^{m+j}}\right). \qquad (11)
\end{aligned}$$

ここで，$0 < k' - k = 2^i l \leq 2^{j+1} - 1$，ただし $0 \leq i \leq j$ かつ l は奇数，としよう．すると

$$\frac{1}{2^{m+j}} \sum_{q=1}^{2^{m+j}} F\left(\frac{p}{2^{m+j}} + \frac{(k'-k)q}{2^{m+j}}\right) = \frac{1}{2^{m+j}} \sum_{q=1}^{2^{m+j}} F\left(\frac{p}{2^{m+j}} + \frac{lq}{2^{m+j-i}}\right). \tag{12}$$

各 $r = 1, 2, 3, \ldots, 2^{m+j-i}$ に対して $lq_r \equiv r \,(\mathrm{mod}\, 2^{m+j-i})$ なる q_r を一つ定めれば l は奇数だから

$$\begin{aligned}
&\#\{1 \leq q \leq 2^{m+j} \,|\, lq \equiv r (\mathrm{mod}\, 2^{m+j-i})\} \\
&\quad = \#\{1 \leq q \leq 2^{m+j} \,|\, lq \equiv lq_r (\mathrm{mod}\, 2^{m+j-i})\} \\
&\quad = \#\{1 \leq q \leq 2^{m+j} \,|\, l(q - q_r) \equiv 0 (\mathrm{mod}\, 2^{m+j-i})\} \\
&\quad = \#\{1 \leq q \leq 2^{m+j} \,|\, q \equiv q_r (\mathrm{mod}\, 2^{m+j-i})\} \\
&\quad = 2^i.
\end{aligned}$$

このことから

$$\frac{1}{2^{m+j}} \sum_{q=1}^{2^{m+j}} F\left(\frac{p}{2^{m+j}} + \frac{lq}{2^{m+j-i}}\right) \;=\; \frac{1}{2^{m+j-i}} \sum_{r=1}^{2^{m+j-i}} F\left(\frac{p}{2^{m+j}} + \frac{r}{2^{m+j-i}}\right)$$

$$= \frac{1}{2^{m+j-i}} \sum_{r=1}^{2^{m+j-i}} F\left(\frac{r}{2^{m+j-i}}\right)$$
$$= \int_0^1 F(t)dt. \tag{13}$$

従って (11)(12)(13) から

$$\begin{aligned}
\mathbf{E}[F(Z_{k'})G(Z_k)] &= \frac{1}{2^{m+j}} \sum_{p=1}^{2^{m+j}} \left(\frac{1}{2^{m+j}} \sum_{q=1}^{2^{m+j}} F\left(\frac{p}{2^{m+j}} + \frac{(k'-k)q}{2^{m+j}}\right)\right) G\left(\frac{p}{2^{m+j}}\right) \\
&= \frac{1}{2^{m+j}} \sum_{p=1}^{2^{m+j}} \left(\frac{1}{2^{m+j}} \sum_{q=1}^{2^{m+j}} F\left(\frac{p}{2^{m+j}} + \frac{lq}{2^{m+j-i}}\right)\right) G\left(\frac{p}{2^{m+j}}\right) \\
&= \int_0^1 F(t)dt \cdot \frac{1}{2^{m+j}} \sum_{p=1}^{2^{m+j}} G\left(\frac{p}{2^{m+j}}\right) = \int_0^1 F(t)dt \int_0^1 G(s)ds.
\end{aligned}$$

以上で (10) が示された.

Step 2. まず，各 $Z_k(\omega')$ は $\{0,1\}^m$ 上で一様分布するから

$$\mathbf{E}[S_N(g(\omega'))] = N\mathbf{E}[X]$$

であることが分かる．次にペアごとの独立性によって

$$\begin{aligned}
\mathbf{V}[S_N(g(\omega'))] &= \mathbf{E}\left[\left(\sum_{k=1}^{N} (X(Z_k(\omega')) - \mathbf{E}[X])\right)^2\right] \\
&= \sum_{k=1}^{N}\sum_{k'=1}^{N} \mathbf{E}\left[(X(Z_k(\omega')) - \mathbf{E}[X])\,(X(Z_{k'}(\omega')) - \mathbf{E}[X])\right] \\
&= \sum_{k=1}^{N} \mathbf{E}\left[(X(Z_k(\omega')) - \mathbf{E}[X])^2\right] \\
&\quad +2 \sum_{1\le k<k'\le N} \mathbf{E}\left[(X(Z_k(\omega')) - \mathbf{E}[X])\,(X(Z_{k'}(\omega')) - \mathbf{E}[X])\right] \\
&= N\mathbf{V}[X].
\end{aligned}$$

以上から，g は要請された性質を持つことが分かる． □

例 9 RWS を用いて § 3.2 の例題を解くことができる．例 6 (§ 5.1) において，$m = 100$, $N = 10^6$，$j = 19$，とした定義 7 の疑似乱数生成器 $g : \{0,1\}^{238} \to \{0,1\}^{10^8}$ を用いるとき，[†16] リスク評価は

$$P_{238}\left(\left|\frac{S_{10^6}(g(\omega'))}{10^6} - p\right| \ge \frac{1}{200}\right) \le \frac{1}{100} \tag{14}$$

[†16] ここでは $2^{j+1} = 2^{20} > 10^6 = N$ であり $2m + 2j = 238$．現実のモンテカルロ法ではサンプルサイズ N があらかじめ定まっていることは少なく，数値実験を行いながら適切な N を見つけていく場合が多い．そのような場合に備えて実際には j を少々大きめに与えておくとよい.

のように得られる (cf. (3)). アリスは自分の意思でどの種 $\omega' \in \{0,1\}^{238}$ でも容易に選ぶことができるから，この場合は長大な乱数は必要ない.

具体的に $S_{10^6}(g(\omega'))$ を求めた例を挙げよう. (アリスの代わりに) 筆者が選んだ種 $\omega' = (x, \alpha) \in D_{119} \times D_{119} \cong \{0,1\}^{238}$ は次の通りである (2 進数表示) ;

$$
\begin{aligned}
x &= \texttt{0.1110110101 1011101101 0100000011 0110101001 0101000100} \\
&\quad \texttt{0101111101 1010000000 1010100011 0100011001 1101111101} \\
&\quad \texttt{1101010011 111100100} \\
\alpha &= \texttt{0.1100000111 0111000100 0001101011 1001000001 0010001000} \\
&\quad \texttt{1010101101 1110101110 0010010011 1000000011 0101000110} \\
&\quad \texttt{0101110010 010111111}
\end{aligned}
$$

このときコンピュータよって $S_{10^6}(g(\omega')) = 546,177$ と計算された. 従って，この場合,

$$\frac{S_{10^6}(g(\omega'))}{10^6} \;=\; 0.546177$$

が求める確率 p の推定値である.

注意 10　RWS の場合，アリスがどんな種 $\omega' = (x, \alpha) \in \{0,1\}^{2m+2j}$ を選ぶべきでないか，について少しだけ助言をすることができる. それは，とくに α を簡単な数にしないことである. 極端な場合 $\alpha = (0, 0, \ldots, 0) \in \{0,1\}^{m+j}$ と選ぶと RWS はほぼ間違いなく失敗することがすぐ分かる.

注意 11　もっと大規模なモンテカルロ積分で $2m + 2j \gg 1$ の場合だと，再び乱数の問題によって，アリスは RWS の種 $\omega' \in \{0,1\}^{2m+2j}$ さえ自分の意思で選ぶことができなくなる. その場合は，さらに別の補助的な疑似乱数生成器 $g' : \{0,1\}^n \to \{0,1\}^{2m+2j}$, $n \ll 2m + 2j$, を用いて $\omega' \in \{0,1\}^{2m+2j}$ を選ぶ破目になる. この場合，合成された疑似乱数生成器 $g \circ g' : \{0,1\}^n \to \{0,1\}^{2m+2j} \to \{0,1\}^{Nm}$ が A_1 に対して安全になるかどうかは分からない.

例 12　例 2 で扱った確率変数 X の最小値を探索する場合にペアごとに独立な確率変数によるサンプリングを利用してみよう. $\Pr(X < c) = 1/10000$ とし，$X'_1, X'_2, \ldots, X'_{40000}$ を X のペアごとに独立なコピーとする. このとき $S' := \sum_{k=1}^{40000} \mathbf{1}_{\{X'_k < c\}}$ とすれば

$$\mathbf{E}[S'] = 4, \qquad \mathbf{V}[S'] = 40000\mathbf{V}[\mathbf{1}_{\{X'_k<c\}}] = 40000\left(1 - \frac{1}{10000}\right)\frac{1}{10000} \;<\; 4.$$

だからチェビシェフの不等式より

$$\Pr(S' \geq 1) \;\geq\; \Pr\left(|S' - 4| < 4\right) \;\geq\; 1 - \frac{4}{4^2} \;=\; \frac{3}{4}.$$

これより $\min_{1 \leq k \leq 40000} X'_k$ は少なくとも確率 3/4 以上で c 以下の値をとることが分かる.

7　数理統計学の視点から

7.1　無作為なサンプリング

我々はモンテカルロ法を賭けと考え，プレーヤーのアリスが自分の意志で疑似乱数の種 $\omega' \in \{0,1\}^n$ を選ぶ，という観点で論じてきた．しかし数理統計学の視点から見ると，ω' がアリスの意志で選ばれるというのは困ったことである．なぜなら，結果に客観性を持たせるために数理統計学では**無作為なサンプリング** (random sampling)[†17]を行うことを重要と考えるからである．実際，RWS の場合は，注意 10 で述べたことを逆手にとって，悪い種 ($\alpha = (0,0,\ldots,0) \in \{0,1\}^{m+j}$) を選んで賭けにわざと負けることができる．すなわち，プレーヤーの意思で結果が左右されることが起こり得るのである．

サンプリングの客観性を厳密に論ずることはもちろん数学の守備範囲を超えている．ここでは，たとえば，本物の硬貨の表裏の出方が誰の意思にも影響されないことを仮定した上で議論することにしよう．このとき，たとえば例 9 の ω' を選ぶときは，硬貨を 238 回投げて，順に，表だったら 1，裏だったら 0，を記録していく．そうして長さ 238 の $\{0,1\}$-列ができたら，それを疑似乱数生成器 g の種 ω' として $S(g(\omega'))$ を計算する．これで，無作為なサンプリングが実行される．じつは，例 9 の ω' は，実際，そのようにして得られた種である．

この方法で非常に長い $\omega \in \{0,1\}^L$ を無作為に選ぶことはあまりに膨大な時間と労力がかかるので事実上不可能である．重要なのは，疑似乱数生成器の利用によって無作為に選ばなければならない $\{0,1\}$-列の長さがきわめて短くなるため，この方法が実際に実行可能になる，ということである．[†18]

7.2　疑似乱数の検定

[5] をはじめ，夥しい数の疑似乱数の検定の多くは次の手順で行われている．

(1) 検定項目 (連の検定，ポーカー検定など) を決める．

(2) 無作為に選ばれた複数の種 ω' をもとに疑似乱数生成器 g によって疑似乱数 $g(\omega')$ を生成し，その結果，棄却されるものの割合を計算する．

(3) 棄却されるものの割り合いがその検定の危険率程度であれば，g を採択し，それを大きく超えるようであれば棄却する．

[†17] ここでいう random sampling はプレーヤの意思に依らないサンプリングを意味する．i.i.d.-サンプリングや RWS のように，確率変数のサンプリングを行うという意味で random sampling という用語を使う場合があるので注意されたい．(ここでは，モンテカルロ法を賭けだと規定しているので，確率変数のサンプリングはプレーヤの意思で行うことを基本としている．)

[†18] この解決方法はじつは古くからある乱数表の使い方 — 乱数表のページや数を拾い始める箇所の決め方は無作為に，あとはルールに従って決定論的に数を拾っていくというやり方 ([5] p.7) — によく似ている．

上の手順において，(1) で選ばれた検定の棄却域を A とすれば，(2) で行っていることは確率 $P_n(g(\omega') \in A)$ を推定する作業と解釈できる．そして (3) ではそれが危険率 $P_L(\omega \in A)$ と近いかどうかを調べていることになる．従ってこうした検定作業は，じつは g の集合 A に対する安全性の検査であるといえるだろう．

8 おわりに

安全な疑似乱数生成器が有用であることは論を待たない．それに比べて，乱数の概念はそれを知ったところで有益でないと考える読者も多いかもしれない．そこで，最後に乱数と確率論の関係について述べておこう．

確率論ではランダムな変量を確率変数という形式で表現するが，これは適当な確率空間 — 硬貨投げの確率空間 $(\{0,1\}^L, 2^{\{0,1\}^L}, P_L)$ としよう — で定義された関数 $X : \{0,1\}^L \to \mathbb{R}$ のことであり，それ自体はランダム性と無縁の概念である．X をランダムな変量と考える際には，$\omega \in \{0,1\}^L$ が P_L に従ってランダムに選ばれ，その結果として $X(\omega)$ がランダムな値をとる，と解釈する．しかし，確率論では確率変数を常に関数として扱うので，$\omega \in \{0,1\}^L$ が選ばれる過程は関知しない．ランダム性は ω が選ばれる過程にこそ存在するので，その意味で確率論は「ランダムとは何か」という問題を避けながら成立していると言えるだろう．

コルモゴロフは，晩年，パーキンソン氏病という難病を患いながらも，このように自身の確率論が避けてきた「ランダムとは何か」という問題に没頭したという ([17] p.115)．そしてついに彼は，乱数という概念を数学的に定式化しこの難問を解いた．乱数を認識するということは，確かに理論的には確率論にとって不要であるものの，それは確率論そのものの深い理解のために非常に有益である．[†19] このことを以下で説明しよう．

コルモゴロフによればランダム性とはすなわち乱数の性質である．乱数の存在は標本空間 $\{0,1\}^L$ が巨大な集合である場合に限り意味を持つ．だから，まず，$L \gg 1$ の場合を調べることがランダム性の研究には重要であることが認識されよう．それで以下，$L \gg 1$ とする．このとき $\{0,1\}^L$ の元のうち大多数を占める乱数を我々は自らの意志では，事実上，選ぶことはできないから，人為的には一様確率測度 P_L に従って ω を選ぶことは不可能である．従って確率測度として P_L を設定するということは，$\{0,1\}^L$ の元が人為的でない方法 — 無作為な方法 — で選ばれることを前提としているわけで，それゆえ，確率空間 $(\{0,1\}^L, 2^{\{0,1\}^L}, P_L)$ はランダムな現象を解明するための枠組みとなり得るのである．

一般に個々の乱数について何らかの性質を知ることは困難である．しかし，乱数は $\{0,1\}^L$ の元のうち大多数を占めるのだから，乱数の性質を知りたければ $\{0,1\}^L$ の元のうち大多数が持つ性質を調べるのが手っ取り早い．それはすでに確率論で詳しく調べられている．大数の法則，中心極限定理をはじめとする様々な極限定理が記述している性質がそうである．恐らく，極限定理こそがランダム性について具体的

[†19] それゆえ数学辞典第 3 版 [9] では存在した “Kolmogorov-Chaitin の複雑性 (コルモゴロフ複雑度)” という用語が第 4 版では削除されたことを大変残念に思う．

に調べることのできるほとんど唯一の数学的表現形式であろう．極限定理が確率論の中心的研究対象と言われる所以である．[†20]

真に驚くべきことは，乱数の概念が発見されるはるか以前から，慧眼の確率論研究者たちが極限定理の重要性を認識し，その研究に努めてきたことである．

参考文献

[1] L. Blum, M. Blum and M. Shub, A simple unpredictable pseudorandom number generator, *SIAM J. Comput.*, **15-2** (1986), 364–383.

[2] M. Blum and S. Macali, How to generate cryptographically strong sequences of pseudorandom bits, *SIAM J. on Computing*, vol. **13**, (1984) 850–864. A preliminary version appears in *Proceedings of the IEEE Foundations of Comput. Sci.* (1982), 112–117.

[3] G.J. Chaitin, Algorithmic information theory, *IBM J. Res. Develop.*, **21** (1977), 350–359.

[4] 舟木直久，確率論，朝倉書店，(2004).

[5] JIS Z 9031 乱数発生及びランダム化の手順，日本規格協会，2001 年改正.

[6] A.N. Kolmogorov, *Selected works of A. N. Kolmogorov. Vol. III. Information theory and the theory of algorithms*, Edited by A. N. Shiryayev [A. N. Shiryaev]. Translated from the 1987 Russian original by A. B. Sossinsky. Mathematics and its Applications (Soviet Series), 27. Kluwer Academic Publishers Group, Dordrecht, (1993) xxvi+275 pp.

[7] M. Luby, *Pseudorandmness and cryptographic applications*, Princeton Computer Science Notes, Princeton University Press, (1996).

[8] P. Martin-Löf, The definition of random sequences, *Inform. Control*, **9** (1966), 602–619.

[9] 数学辞典 (第 4 版)，484 (XX-9) “乱数とモンテカルロ法”，岩波書店，(2007), 1589–1591.

[10] H. Sugita, Robust numerical integration and pairwise independent random variables, *Jour. Comput. Appl. Math.*, **139** (2002), 1–8.

[11] H. Sugita, Dynamic random Weyl sampling for drastic reduction of randomness in Monte Carlo integration, *Math. Comput. Simulation*, **62** (2003), 529–537.

[12] 杉田洋，複雑な関数の数値積分とランダムサンプリング,「数学」岩波書店 **56-1**, (2004) 1–17.

[13] H. Sugita, Security of Pseudo-random Generator and Monte-Carlo Method, *Monte Carlo Methods and Appl.*, **10-3**, VSP(2004), 609–615.

[14] H. Sugita, *Monte Carlo Method, Random Number, and Pseudorandom Number*, MSJ Memoirs vol.25 (2011), xiv+133 pp.

[15] H. Sugita, The Random Sampler, 疑似乱数生成と動的ランダム-ワイル-サンプリングのための C/C++言語ライブラリ，下記にて公開:
`http://www.math.sci.osaka-u.ac.jp/~sugita/mathematics.html`.

[†20] たとえば教科書 [4] の「まえがき」と第 1 章「確率論を学ぶにあたって」の 1 ページ目に，確率論の真の主題は極限定理である，と明言されている．

[16] H. Sugita and S. Takanobu, Random Weyl sampling for robust numerical integration of complicated functions, *Monte Carlo Methods and Appl.*, **6-1** (1999), 27–48.

[17] 高橋陽一郎+志賀浩二，確率論をめぐって，日本評論社，(1992).

[18] A. Yao, Theory and applications of trapdoor functions, *Proceedings of the IEEE Foundations of Comput. Sci.*, (1982), 80–91.