モンテカルロ積分のためのサンプリング法

技術解説

杉田 洋*

Sampling method for Monte-Carlo integration

Key Words : Monte-Carlo method, numerical integration, Random Weyl sampling

私はここ数年モンテカルロ法の数学的基礎付けの研究に携わってきました．モンテカルロ法とは，疑似乱数を用いて確率変数を模倣し，それを数値計算に用いる様々な手法の総称です．

小論で紹介させて頂いた研究成果は，C/C++言語ライブラリとして[5]で公開しています．ご自由にご利用頂き，ご意見をお聞かせ頂ければ幸甚に存じます．

1．モンテカルロ積分

確率変数Xの平均(期待値)$\mathbf{E}[X]$の近似値を求めるために，Xと同分布の独立確率変数列$X_1, \ldots, X_N$を疑似乱数を用いて模倣し，それらの相加平均

$$\overline{X}_N := \frac{X_1 + \cdots + X_N}{N}$$

でもって$\mathbf{E}[X]$の近似値とする，という方法をモンテカルロ積分といいます．実際に実施されているモンテカルロ法の大部分はこのモンテカルロ積分です．

$\overline{X}_N$の分散$\mathbf{Var}[\overline{X}_N]$は

$$\mathbf{Var}[\overline{X}_N] = \frac{1}{N}\mathbf{Var}[X] \qquad (1)$$

となるので，チェビシェフの不等式により，

$$P(|\overline{X}_N - \mathbf{E}[X]| > \varepsilon) < \frac{\mathbf{Var}[X]}{N\varepsilon^2}, \quad \varepsilon > 0 \qquad (2)$$

が成り立ちます(ただしPは確率を表す)．これより，Nが十分大きいとき，$\overline{X}_N$の値は高い確率で$\mathbf{E}[X]$に近いことが分かります(大数の法則)．

たとえばもしXがm回の硬貨投げで定まるような確率変数とすれば，$\overline{X}_N$はNm回の硬貨投げで定まる確率変数となります．このように，モンテカルロ積分はランダム性を大きくすることによって近似の精度を上げる技術といえます．

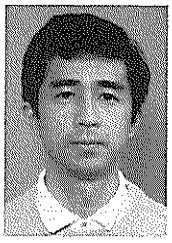
しかしながら，モンテカルロ積分においてサンプル数Nを大きくすることは，現実においては近似の精度を損ないかねません．それはランダムなサンプルを疑似乱数を用いて生成するからです．完全な疑似乱数は現時点においてもなお存在が証明されていません(その存在証明は有名な**P≠NP**予想と関係があります)．そのため，非常に大量の疑似乱数を生成すると，そのわずかな統計的偏りが増幅されて，結果として，モンテカルロ積分の精度がひどく落ちることがあるのです．

2．ペアごとに独立なサンプリング

実はごくわずかな疑似乱数を使うだけで，(2)と同等の確率的誤差評価を達成するようなサンプリング法が，計算機科学の分野では，以前からよく知られていました．それは同分布でペアごとに独立なサンプルを用いたサンプリング法です．

モンテカルロ積分で一般に用いられているサンプル$X_1, \ldots, X_N$は完全な独立性を仮定していますが，ペアごとに独立なサンプルとは，異なるどの二つの確率変数X_iとX_jは独立でも，3個以上の組$X_i, X_j, X_k, \ldots$は必ずしも独立でないような確率変数列のことです．同分布でペアごとに独立な確率変数列$X_1, \ldots, X_N$に対しても(1)が，従って確率的誤差評価(2)も，成り立つことが分かります．

たとえばm回の硬貨投げで定まるような任意の確率変数Xに対して，それと同分布でペアごとに独立



* Hiroshi SUGITA
1958年2月生
1986年京都大学大学院・理学研究科数学専攻・博士課程修了
現在，大阪大学・大学院・理学研究科・数学専攻，教授，理学博士，確率論
TEL 06−6850−5291
FAX 06−6850−5327
E-Mail sugita@math.sci.osaka-u.ac.jp

な確率変数列$X_1, \ldots, X_N$を作るために必要な最小限のランダム性はどれほどでしょうか.

少なくともX_1とX_2は独立でなければなりませんから，最低$2m$ビットのランダム性が必要なことは明らかです．ところが驚くべきことに，$2m$ビットのランダム性があれば，ペアごとに独立な確率変数列$X_1, \ldots, X_N, N \leq 2^m$，を作ることができるのです．(ここで制限$N \leq 2^m$は実質的な制限になっていません．なぜならサンプル数を2^mとしてよいのなら，m回の硬貨投げのすべての場合を数え上げて$\mathrm{E}[X]$の値をきっかり計算することができるのですから．つまりモンテカルロ積分は$N < 2^m$の場合に行えば十分です.)

しかし残念なことに，最小のランダム性でペアごとに独立な確率変数列を作るその方法は，有限体$GF(2^m)$の演算を用いていて，とくにその掛け算に多くの時間が必要になるためにモンテカルロ積分で実際に用いることはできません.

ランダム・ワイル・サンプリング

そこで，私は，有限回の硬貨投げで定まるような確率変数Xに対して，それと同分布でペアごとに独立な確率変数列を高速に作る方法－ランダム・ワイル・サンプリング(RWSと略称)－を[3]で発表しました．RWSのランダム性は最小ではありませんが，最小に近いもので，Xがm回の硬貨投げで定まる確率変数のとき，ペアごとに独立な確率変数列$X_1, \ldots, X_N$を生成するためにRWSが必要とするランダム性は$2(m+\lceil \log_2 N \rceil)$ビットです[1].

たとえば，$m=100$回の硬貨投げの関数として得られる確率変数の平均をサンプル数$N=10^7$で通常のサンプリングによって求める場合は，$100 \times 10^7 = 10^9$ビットのランダム性が必要ですが，RWSは僅か$2 \times \lceil 100 + \log_2 10^7 \rceil = 248$ビットのランダム性しか必要としません.

このようなランダム性の削減によって，RWSは以下のような特長を持ちます.

- 疑似乱数の質に大変鈍感である．すなわち質の高くない疑似乱数でも使用できる.
- 質は高いが遅い疑似乱数生成器を用いてもサンプルの生成速度にほとんど影響しない．その場合，信頼性の高いモンテカルロ積分が得られる.

数式を使って恐縮ですが，RWSを正確に述べることにします．$D_m := \{i/2^m ; i = 0,1,\ldots 2^m-1\} \subset [0,1)$とします．2進数展開を通じて$D_m$と$\{0,1\}^m$を同一視しましょう．$D_m$上の一様確率測度$P_m$は$\{0,1\}^m$上の確率測度と思えば，ちょうど$m$回の硬貨投げの確率法則に外なりません．$x \in [0,1)$に対し，$\lfloor x \rfloor_m := \lfloor 2^m x \rfloor / 2^m \in D_m$とします[2]．そこで次の定理を利用したのがRWSです.

定理.([3]) $x, \alpha \in \mathrm{D}_{m+j}$をそれぞれ$P_{m+j}$に従う独立な確率変数とする．このとき

$$Z_n := \lfloor (x + n\alpha) \text{の小数部分} \rfloor_m \in D_m,\ n = 1, \ldots 2^j,$$

と定義すれば，$\{Z_n\}_{n=1}^{2^j}$は分布P_mに従うペアごとに独立な確率変数列である.

動的ランダム・ワイル・サンプリング

RWSでは対象となる確率変数は予め決まった回数の硬貨投げの関数でなくてはいけません．それに対し，[4]で発表した動的ランダム・ワイル・サンプリング(DRWSと略称)は，そうした制限なしに，モンテカルロ法で扱い得る任意の確率変数に適用可能な，ペアごとに独立なサンプルによるモンテカルロ積分法です.

具体的にいうと，$\{\omega_i\}_{i=1}^{\infty}$, $\omega_i \in \{0,1\}$，を硬貨投げとするとき，たとえば，確率変数

$$X := \min\{n \in \mathbf{N} ; \omega_1 + \omega_2 + \cdots + \omega_n = 5\}$$

は硬貨を投げ続けて表に出た数が5となる最初の時刻nです．このとき，Xはいつも決まった回数だけの硬貨投げで値が決定するわけではありませんが，疑似乱数を使って模倣することは簡単にできます．そして，DRWSはこのようなXに対しても，ごく少量のランダム性でもって，ペアごとに独立なサンプルを生成して平均を求めることができるのです.

なお，その名が示す通り，DRWSはRWSを変形して構成されますが，詳しいことは[4]を参照して下さい.

[1] $\lceil x \rceil$はxを下回らない最小の整数を表す.

[2] $\lfloor x \rfloor$はxを越えない最大の整数を表す.

3. 疑似乱数

モンテカルロ積分を実行するときのもう一つの問題点はサンプルを抽出するときの手順です.

再びXは$m=100$回の硬貨投げで定まる確率変数とします. これに通常のモンテカルロ積分(サンプル数$N=10^7$)を実行しますと, $\overline{X}_N$を計算するには$Nm=10^9$ビットのランダム性が必要でした. 言い換えると, プログラムに長さ10^9のビット列を入力しなければなりません. これはどうやっても人がキーボードから入力できるビット列の長さではありません. そこで疑似乱数のお世話になるのです.

なおRWSでは, この場合, 入力は248ビットで済みますから, これはキーボードから十分打ち込める程度の長さです. ですから, この場合は疑似乱数は必要ありません. もちろん, (D)RWSを用いても, とてもキーボードから打ち込めないほど長いビット列の入力が必要になる場合もありますから, その場合は, やはり疑似乱数の助けが必要になります.

ワイル変換を用いた疑似乱数

私は, 確率論における一つの極限定理の応用としてある疑似乱数生成法を[2]で発表しました. その後, この疑似乱数はJIS[1]にも採用されました.

再び数式を用いて, その疑似乱数の根拠となった極限定理を述べます. $d_i(x)\in\{0,1\}$を実数$x\geq 0$の2進数展開の小数点以下第i桁目の数とします. そして, 任意の無理数αと正整数mに対して

$$Z_n^{(m)}(x):=\Big(\sum_{i=1}^{m} d_i(x+n\alpha)\Big) \bmod 2,\quad x\in[0,1),$$

とします. (これは$(x+n\alpha)$の小数部分の上位mビットのパリティです.)このとき次の定理が成り立ちます.

定理. ([2]) xを区間$[0,1)$の一様分布に従う確率変数とすれば, 確率変数列$\{Z_n^{(m)}(x)\}$は$m\to\infty$のとき, 硬貨投げの確率過程に分布収束する.

各mに対して, 確率変数列$\{Z_n^{(m)}\}$の任意の有限次元分布を理論的に計算することができるので, その結果から, $m\geq 90$くらいで$\{Z_n^{(m)}\}$は硬貨投げと区別することが実用上十分困難であると予想しています. それで[1, 2, 5]では, $\{Z_n^{(90)}\}$を疑似乱数として用いることを提案しています.

参考文献

[1] 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.

[2] H. Sugita, Pseudo-random number generator by means of irrational rotation, *Monte Carlo Methods and Applications*, VSP, 1-1 (1995), 35-57.

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

[4] H. Sugita, Dynamic random Weyl sampling for drastic reduction of randomness in Monte Carlo integration, *Mathematics and Computers in Simulation* 62(2003), 529-537.

[5] 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://idisk.mac.com/hiroshi_sugita/Public/imath/mathematics.html.