擬似乱数の評価

高橋 絢那 (指導教員：萩田真理子)

1　はじめに

擬似乱数とは, 一見乱数列のように見えるが, 実際には計算機による確定的な計算により, 求められている数列である.

良い擬似乱数の条件には, 周期が長いこと・高速に生成できること・統計的検定に耐えられること, があげられる.

本研究では, rand(), CST により生成した擬似乱数に対して, ランダムウォークを用いて正の区域の滞在時間を測定し, χ^2 検定を行い, 各擬似乱数の乱数性を評価する.

長い間標準擬似乱数として使われてきたが乱数性は低いことが知られている rand() と, 最新の擬似乱数生成法 CST により生成された擬似乱数, それぞれに同じ検定を行うことによって CST の乱数性を評価することを目的としている.

2　言葉の定義

ランダムウォーク

ランダムウォークとは, 原点を出発点としてコイントスを繰り返し「表が出たら $(1,1)$ 方向に 1 歩進み, 裏が出たら $(1,-1)$ 方向に 1 歩進む」というルールで生成される折れ線グラフのことである. コイントスが n 回のとき, n 歩のランダムウォークという. 本研究では, ビットの 1 をコインの表, ビットの 0 をコインの裏と見立てて使用する.

$X_1, \cdots, X_n$ は独立な確率変数で, $Pr(X_i = 1) = Pr(X_i = 0) = \frac{1}{2}$ を満たすものとするとき $S_0 = 0, S_i = \sum_{j=1}^{i} X_j$ で定義される確率変数列 $S_0, S_1, \cdots$. S_n が n 歩のランダムウォークである.

正の区域の滞在時間

$y \geq 0$ の上半平面を歩いた歩数を正の区域の滞在時間という.

コイントスで作ったランダムウォークの長さ n が偶数であれば, 正の区域の滞在時間と負の区域の滞在時間はともに偶数になる.

n 歩のランダムウォークにおいて, 正の区域の滞在時間が k 時間となる確率を $P_{k,n}$ と表す. n が偶数の時の $P_{k,n}$ の実現確率は以下の式で表される.

$$P_{k,n} = u_{2k} \cdot u_{2n-2k} \quad , u_{2k} = \begin{pmatrix} 2k \\ k \end{pmatrix} \cdot \frac{1}{2^{2k}} \qquad (1)$$

擬似乱数からランダムウォークを生成した場合, この確率分布に従わなければ乱数性が低いと言える.

3　擬似乱数生成法

本研究で扱う擬似乱数の生成法は以下の 2 つである.

rand()

C 言語の 70〜90 年代の標準擬似乱数.
線形合同法であり, 以下の式で定義されている.

$$x_{n+1} := ax_n + c \bmod M$$
$$a = 1103515245,\ c = 12345, M = 2^{31}$$

周期は $M = 2^{31}$.

CST(Combined Small Twister)

2010 年, 山形大学の西村先生によって提案された高速で周期の短い乱数. 高速で生成できる, 周期 $2^{64}-1$ の擬似乱数 x_i と周期 $2^{89}-1$ の擬似乱数 y_i を各ビットごとに排他的論理和で足し合わせたもの.

$$x_{i+2} = LROT((x_{i+1} >> 7) \oplus x_{i+1}), 9)$$
$$\oplus((x_i >> 12) \oplus x_i)A \quad (i = 0, 1, ...)$$

$$A = \begin{pmatrix} 0 & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \\ a_{32} & a_{31} & & \cdots & a_1 \end{pmatrix}$$

ただし, $>>$ は右方向のビットシフト.

$$y_{i+3} = LROT((y_{i+2} >> 9) \oplus y_{i+2}), 17)$$
$$\oplus((z >> 12) \oplus z)B$$
$$z = U(y_i, 25)|L(y_{i+1}, 7) \quad (i = 0, 1, ...)$$

$$B = \begin{pmatrix} 0 & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \\ b_{32} & b_{31} & & \cdots & b_1 \end{pmatrix}$$

ただし, U は上位桁を, L は下位桁をとる.

CST：$x_i \oplus y_i$.　周期は約 2^{153}.

4　検定方法

100 個の擬似乱数それぞれから, 任意の 1 ビットを取り出し, これを元に 100 歩のランダムウォークを生成, 正の滞在時間を計算する. 正の滞在時間ごとにランダムウォークを 5 つのグループに分けて, (1) 式より求めた期待度数と実際の観測度数から χ^2 値を計算し, 自由度 4 の χ^2 検定を行う. 有意水準は 0.05 とし, 棄却域は $\chi^2 \geq 9.488$ である.

ここで, 5 つのグループはほぼ同確率でランダムウォークが分けられるように設定した.

χ^2 検定とは, 観測度数と期待度数から計算して求めた χ^2 値を, χ^2 分布を用いて評価する検定.

χ^2 値は以下の式で求める.

$$\chi^2 = \sum_{i=1}^{m} \frac{(O_i - E_i)^2}{E_i} \qquad (2)$$

O_i : 観測度数, E_i: 期待度数

m : グループ数 (本検定では 5)

乱数性の低い擬似乱数は, 期待度数と観測度数の差が大きくなるため, χ^2 値も大きくなる. よって, 棄却されやすい.

つまり, より多くの桁を棄却できる検定が, 乱数性を評価するのに適した良い検定である.

5 検定結果

5.1 rand()

100 個の擬似乱数を 50 本, 100 本, 200 本, 500 本, 1000 本と扱う本数を変えて, それぞれの場合での任意ビットの χ^2 値を求めた.

その結果を表 1 に示す:(自由度 4, 有意水準 0.05 で棄却されるところはグレー背景)

表 1: rand() 擬似乱数の任意ビットの χ^2 値

	50 本	100 本	200 本	500 本	1000 本
2^0 桁	69.969	139.938	279.876	699.690	1399.380
2^1 桁	40.821	81.782	163.565	408.912	817.824
2^2 桁	5.302	11.760	23.586	58.828	117.930
2^3 桁	20.110	40.460	75.852	191.746	378.762
2^4 桁	7.887	13.989	25.863	65.855	130.993
2^5 桁	4.867	8.729	18.464	47.642	96.054
2^6 桁	11.183	18.295	23.410	59.762	112.989
2^7 桁	1.050	2.678	2.129	4.768	10.109
2^8 桁	5.672	7.757	3.096	10.809	22.462
2^9 桁	17.038	16.091	5.717	4.903	5.373
2^{16} 桁	3.497	2.268	1.918	1.451	2.718
2^{24} 桁	1.512	2.720	1.982	6.999	4.227
2^{30} 桁	4.000	1.837	3.640	0.214	0.750

表 1 から, 1000 本の擬似乱数を扱うと下位 9 桁まで棄却されることがわかる. よって, 下位 9 桁には偏りがあることがわかった.

また, 50 本程度では棄却される桁は少なく, かつ棄却できる桁が飛び飛びで現われている. 扱う本数を多くするに従って, 棄却される桁が増えていき, 1000 本程度では下位 9 桁まで棄却される. このことから, 1000 本程度扱えば乱数性の低さを効果的に検出できると考えられる.

よって, rand() で生成された 100 個の擬似乱数の検定においては, 1000 本以上を扱うと擬似乱数の乱数性を評価するのに適した良い検定となるということがわかった.

5.2 CST(Combined Small Twister)

rand() と同様に, 100 個の擬似乱数を 50 本, 100 本, 200 本, 500 本, 1000 本と扱う本数を変えて, それぞれの場合での任意ビットの χ^2 値を求めた.

その結果を表 2 に示す:(自由度 4, 有意水準 0.05 で棄却されるところはグレー背景)

表 2: CST 擬似乱数の任意ビットの χ^2 値

	50 本	100 本	200 本	500 本	1000 本
2^0 桁	1.401	5.413	2.996	2.786	3.961
2^8 桁	0.432	1.670	4.060	6.317	4.662
2^{16} 桁	0.901	1.426	2.150	2.953	2.951
2^{24} 桁	4.281	1.383	3.910	2.820	3.761
2^{30} 桁	4.250	4.553	2.540	1.515	3.223
2^{31} 桁	2.062	2.187	1.599	5.576	3.315

表 2 から, CST で生成された擬似乱数 100 個については扱う本数によらず, また, 取り出す桁によらず, 有意水準 0.05 で棄却されるところはないことがわかる.

本検定では, rand() で生成された擬似乱数の乱数性の低さを検出することはできたが, CST で生成された擬似乱数の乱数性の低さは検出されなかった.

つまり, 本検定では rand() よりも CST で生成された擬似乱数の方が乱数性が高いという結果が得られた.

6 まとめと今後の課題

本研究では, 長い間標準擬似乱数として使われてきたが乱数性は低いことが知られている rand() と, 最新の擬似乱数生成法 CST によって生成された擬似乱数に対して同じ検定を行い, その結果を比較することで CST の乱数性を評価した. その結果, 本研究の検定方法においては, rand() により生成された擬似乱数よりも CST により生成された擬似乱数の方が乱数性が高いと言えた.

今後は, 別の検定方法を用いて違う角度から CST を評価してみたい. また, 今回扱えなかった多くの擬似乱数生成法についても同様の検定を行って, 乱数性を比較してみたい.

謝辞

山形大学の西村拓士先生に, 最新の擬似乱数生成法 CST を本研究のために提供していただいたことを感謝いたします.

参考文献

[1] 津野義道：ランダム・ウォーク 乱れに潜む不思議な現象, 牧野書店, 2002

[2] 草間時武：統計学, サイエンス社, 1975

[3] 東京大学教養学部統計学教室 編：統計学入門, 東京大学出版会, 1991