

擬似乱数列と C での使用例

樋口さぶろお

龍谷大学工学部数理情報学科

計算科学☆実習 B L02(2020-04-08 Wed)

最終更新: Time-stamp: "2020-05-09 Sat 13:57 JST hig"

今日の目標

- 擬似乱数列とは何か粗く説明できる
- C で連続一様分布の擬似乱数列を生成できる
- C で離散型確率分布の擬似乱数列を生成できる



ここまで来たよ

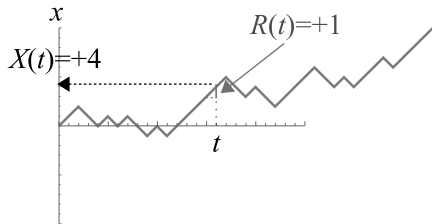
- 擬似乱数列と C での使用例
 - ランダムウォーク
 - 擬似乱数列
 - 離散型確率分布にしたがう擬似乱数

ランダムウォーク (確率過程の例)

確率変数 $X(t)$ がランダムウォークの座標 (数列 $\{X(t)\}$), 確率変数 $R(t)$ が独立同分布にしたがう (階差数列 $\{R(t)\}$)

$$\text{漸化式 } X(t) = X(t-1) + R(t) \quad (t = 1, 2, \dots), \quad \text{初項 } X(0) = a.$$

現象の数理 A



例.

$R(t)$	確率
+1	p
-1	$q(=1-p)$

岩薩林 確率・統計ベルヌーイ分布, $n = 1$ 二項分布 (p.66)

どうやってコンピュータで?

ここまで来たよ

- 擬似乱数列と C での使用例
 - ランダムウォーク
 - 擬似乱数列
 - 離散型確率分布にしたがう擬似乱数

離散型確率分布にしたがう擬似乱数列の生成

モンテカルロ法

確率的/決定的な量を計算するのに、確率変数の標本抽出を実際にコンピュータで **(擬似) 乱数** ((pseudo) random number) を使って行う方法

だからモンテカルロ法には擬似乱数列が必要.

(擬似) 乱数列

$R(1), R(2), R(3), R(4), \dots$ 独立同分布にしたがう確率変数としたとき、サイコロ (やコンピュータや乱数表) を使って作る, そのサンプル (標本).

擬似乱数列を C 言語で生成しよう.

確率統計☆演習 I(2019)L05

岩薩林 確率・統計 §3.1

確率統計☆演習 I(2019)L07

C 言語での乱数の使い方

学習用で '低品質'. ゲームくらいならいいけど, 正確な科学技術計算に使う方法はまた後で Linux, macOS では高品質な `drand48()` や `mt=メルセンヌツイスター`

`include <stdlib.h>` すると使えるライブラリ関数

```
1 int rand(); /* 0以上 RAND_MAX以下の整数を  
2           同確率  $1/(1+RAND\_MAX)$  で返す関数 */  
3 void srand(unsigned seed); /* その初期化. まて次回以降. */
```

`RAND_MAX` は `M_PI` みたいな定数. 値はコンパイラ依存. 例 $2^{31} - 1$.

一様離散分布 $0,1,\dots,RAND_MAX$ にしたがう擬似乱数列

ソースコード 1: 擬似乱数

```
1  /*
2  rand0.c --- 0,1,2,.., RANDMAX の離散一様分布
3  Time-stamp: "2020-04-08 Wed 07:48 JST hig"
4  */
5  #define _CRT_SECURE_NO_WARNINGS // Visual C++用おまじない
6  #include <stdio.h>
7  #include <stdlib.h> /* srand(), rand() を使うのに必要 */
8
9
10 int main(){
11     int seed; /* 擬似乱数のシード */
12     int r;
13     int n;
14     int nmax=10;
15
16     scanf("%d",&seed);
17     srand(seed); /* シードの設定 */
18     for(n=0;n<nmax;n++){
19         r=rand();
20         printf("%d\n",r);
21     }
22     return 0;
23 }
```

連続一様分布 $U(0, 1)$ にしたがう擬似乱数列

ソースコード 2: 擬似乱数

```
1  /*
2  randa.c — [0,1)一様乱数を返す関数
3  Time-stamp: "2020-04-08 Wed 10:04 JST hig"
4  */
5  #define _CRT_SECURE_NO_WARNINGS // Visual C++用おまじない
6  #include <stdio.h>
7  #include <stdlib.h> /* srand(), rand() を使うのに必要 */
8
9  /* 関数プロトタイプ宣言 */
10 double getuniform();
11
12 int main(){
13     int seed; /* 擬似乱数のシード */
14     int n; /* カウンタ 標本内通し番号 */
15     int nmax=10; /* 擬似乱数を得る回数=サンプルサイズN */
16     double r;
17
18     scanf("%d",&seed);
19     srand(seed); /* シードの設定 */
20     for(n=0;n<nmax;n++){
21         r=getuniform();
22         printf("%f\n",r);
23     }
24     return 0;
25 }
26
27 /** [0,1)一様擬似乱数を返す */
28 double getuniform(){
29     return rand()/(RAND_MAX+1.0);
30 }
```


ここまで来たよ

- 擬似乱数列と C での使用例
 - ランダムウォーク
 - 擬似乱数列
 - 離散型確率分布にしたがう擬似乱数

離散型確率分布にしたがう擬似乱数列

ソースコード 3: 擬似乱数

```
1 /*
2 rand1.c --- -1 or +1 を確率1/4, 3/4で選ぶ乱数
3 Time-stamp: "2020-04-08 Wed 09:08 JST hig"
4 */
5 #define _CRT_SECURE_NO_WARNINGS // Visual C++用おまじない
6 #include <stdio.h>
7 #include <stdlib.h> /* srand(), rand() を使うのに必要 */
8
9 /* 関数プロトタイプ宣言 */
10 double getuniform();
11 int getrandom(double y);
12
13 int main(){
14     int seed; /* 擬似乱数のシード */
15     int n; /* カウンタ 標本内通し番号*/
16     int nmax=100; /* 擬似乱数を得る回数=サンプルサイズN */
17     int r;
18
19     scanf("%d",&seed);
20     srand(seed); /* シードの設定 */
21     for(n=0;n<nmax;n++){
22         /* srand(seed); */ /*ここに置くと? */
23         r=getrandom(getuniform());
24         printf("%d\n",r);
25     }
26     return 0;
27 }
28 /** [0,1) 一様擬似乱数を返す */
29 double getuniform(){
30     return rand()/(RAND_MAX+1.0);
31 }
32
33 /** -1 or +1 を確率1/4, 3/4 で返す乱数 */
34 int getrandom(double y){
35     if( y < 0.25 ){
36         return -1;
37     } else {
38         return +1;
39     }
40 }
```

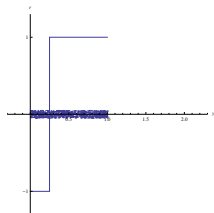
しくみ

離散型確率変数 X .

$$\text{確率関数 } f(x) = \begin{cases} \frac{1}{4} & (x = -1) \\ \frac{3}{4} & (x = +1) \end{cases}$$

```

1  /* 引数 y が [0, 1) 一様乱数なら,
2     getrandom の返り値は
3     確率 1/4 で -1, 確率 3/4 で +1 */
4  int getrandom(double y){
5     if( y < 0.25){
6         return -1;
7     } else {
8         return +1;
9     }
10 }
```



```
r=getrandom(getuniform());
```

マイいかさまコイン関数を書こう

L02-Q1

Quiz(擬似乱数の使いかた)

引数 y として $[0, 1)$ 一様乱数が与えられたとき, 下の確率で値を返す `int getrandom(double y)` を, サンプルプログラムを参考に書こう.

返り値	確率
0.6	0.7
0.4	0.3

マイいかさま三角エンピツ関数を書こう

L02-Q2

Quiz(離散的な乱数の生成)

離散的確率変数 R の確率分布は次であたえられる。

$$f(r) = \begin{cases} \frac{2}{8} & (r = 1) \\ \frac{1}{8} & (r = 2) \\ \frac{5}{8} & (r = 3) \\ 0 & (\text{他}) \end{cases}.$$

引数 y として $[0, 1)$ 一様乱数を与えるとき、上の確率分布に従う乱数 r を返す関数

`int getrandom(double y)` を定義しよう。

$a \leq y < b$ のとき、1 を返すとすると、1 が返される確率は $\int_a^b 1 dx$ 。
 $r = 1, 2, 3$ について a, b をうまく調整していけばいい。

動画解説

<https://youtu.be/Yc6bzcrfLeo>

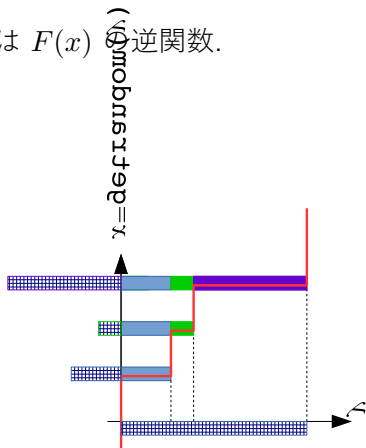
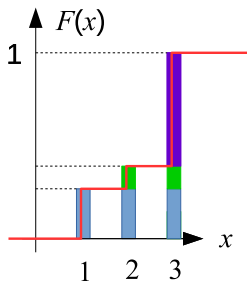
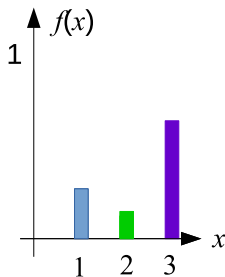


コース後半に自然につながるやり方 (逆関数法) の紹介

長さ 1 を, 棒の長さにあわせて場合分け.

$$\text{累積分布関数 } F(x) = \sum_{x'=-\infty}^x f(x')$$

けっきょく, `int getrandom(double y)` は $F(x)$ の逆関数.



L02-Q3

Quiz(期待値)

離散型確率変数 R は、値 $R = 0$ を確率 $2/13$ で、値 $R = 3$ を確率 $4/13$ で、値 $R = 4$ を確率 $7/13$ でとる。

引数 y として $[0, 1)$ 一様乱数を与えるとき、上の確率分布に従う乱数 r を返す関数 `int getrandom(double y)` を定義しよう。

予習復習問題のやり方+今後の予定

Note Math Moodle

<https://note.math.ryukoku.ac.jp/moodle>



お知らせ

- 2019-04-14 火 4 実習的 時間内にやります。
- Math ラウンジチャンネル
- 樋口オフィスアワーチャンネル原則火 5

Moodle App for iOS/Android



URL をきかれたら <https://note.math.ryukoku.ac.jp/moodle> で登録。