

ランダムウォークと擬似乱数生成

樋口さぶろお

龍谷大学工学部数理情報学科

計算科学☆演習 II L01(2015-04-10 Fri)

今日の目標

- ランダムウォークとは何か説明できる
- Cで擬似一様乱数を生成できる
- Cで離散的な値をとる乱数を生成できる



<http://hig3.net>

ここまで来たよ

1 はじめに

- この授業どんなのり?

2 ランダムウォーク

- ランダムウォーク
- 擬似乱数
- 擬似乱数の正しい/間違っただプログラム

講義+演習の目標

もう少し正確にはシラバスを見てね.

- 現象の確率モデルとは何か, 例をあげて説明できる.
- 確率モデルをオイラー表示とラグランジュ表示で表現し, 量を計算することができる.
- 確率モデルのシミュレーションのプログラムを作成し, その実行結果から, 表計算ソフトウェア・統計ソフトウェアを用いて統計的推定・検定を行うことができる
- ⋮
- チームで協力して問題を解決できる, 効率よく質問できる, 自分の学習方法を改善できる

計算科学☆演習 II を履修した方がいい人

- 確率過程論 stochastic process (=時間に依存する確率的現象) を知りたい人
- 微分方程式 (決定論的モデル) の見ていない, 残り半分の確率論的モデルを見たい人
- モデル駆動の研究が見ていない, データ駆動の研究の世界を見たい人
- 偶然性のあるゲームを仕組みからわかって作りたい人
- 確率を, プログラム作成と組み合わせて実感したい人
- コンピュータでデータの解析ができるようになりたい人

計算科学☆演習 II を履修しない方がいい人

- (単位をとっているかどうかに関わらず) 確率統計☆演習 I, 数値計算法・演習がぜんぜんわかってない感がある人, この機会にわかろうという決意のない人
- 講義・演習を片方しか履修できない人

計算科学 II(講義) ののり

必ず計算科学演習 II と同時に履修してね.

成績計算注文の多い科目です…科目の成績 100 ピーナッツは

- 30 ピーナッツ:毎回授業での quiz, 授業時間外の予習復習など
- 30 ピーナッツ:プチテスト
- 40 ピーナッツ:ファイナルトライアル
- その他追加ピーナッツ. その時に説明.

ファイナルトリアル時点で 20 ピーナッツ未満の人は, 本試験は (平均点を上げるために) 参加をおすすめしますが, 追試験は実施しません.

資料授業で配布. 授業後に欲しい人は <http://hig3.net> から各自ダウンロード. 1-503 前のレターボックスに残ってることも.

欠席届ピーナッツ的に考慮されたい場合は, 専用用紙に事情を説明する書類を貼って, 授業前後各 5 分に提出 (事前事後とも可. ファイナルトリアルが締切). 欠席に事前連絡は不要. 何回欠席しても期末試験受験資格を失うことはありません.

計算科学演習 II ののり

必ず計算科学 II と同時に履修してね.

成績計算注文の多い科目です…科目の成績 100 ピーナッツは

- 20 ピーナッツ:毎回の課題
- 80 ピーナッツ:プチテスト春 15+初夏 30+夏 35

実習室に行ったら, <http://hig3.net> → 計算科学☆演習 II へ.

担当者ののり

- なまえ: 樋口さぶろお hig-compsci2@math.ryukoku.ac.jp
- へや: 1-502
- オフィスアワー: 月 6(1-539), 金 6(1-502). 訪問歓迎な時間: 火木昼.
お弁当持参歓迎. お湯あげます.
- Web ページ: <http://hig3.net> 演習の指示や, スケジュールもここから.

1 週間のタイムライン

- ① 金 11:05 まで 演習の復習, 講義の予習問題 L(e ラーニング) 成績としては講義
- ② 金 2 計算科学 II(7-002), quiz
- ③ 金 6 樋口オフィスアワー (1-502)
- ④ 月 or 火 このころ演習のタスク公開
- ⑤ 月 6 樋口オフィスアワー (1-539)
- ⑥ 水 13:35 まで 演習の予習問題 E(e ラーニング) 成績としては演習
- ⑦ 水 3 計算科学演習 II(1-609), quiz 返却, タスク提出

ここまで来たよ

① はじめに

- この授業どんなのり?

② ランダムウォーク

- ランダムウォーク
- 擬似乱数
- 擬似乱数の正しい/間違っただプログラム

C 言語で数列の計算

数値計算法

数列 $\{X(t)\}$, 時刻 $t = 0, 1, 2, \dots$

初項 $X(0) = 0$

漸化式 $X(t+1) = X(t) + R(t+1)$.

階差数列 $R(t+1) = \text{定数}$ なら $X(t)$ は等差数列.

C 言語で数列を書くと?

```
int x;  
int r;  
int t;  
x=初項;  
for (t=0;t <100;t++){  
    printf("%d\n",x);  
    r=(階差数列の一般項 R(t+1));  
    x=x+r;  
}
```

ランダムウォーク

ランダムウォーク \Leftrightarrow 階差数列 $R(t+1)$ が

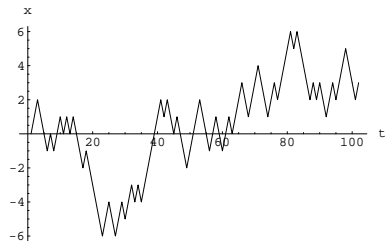
現象の数学 A, 確率統計 II

つまり $R(t+1)$ がランダム. 例えば, こんな場合.

$R(t+1)$	確率
+1	p
-1	$q(=1-p)$

ランダムウォークってどんな
ところに出てくる?

等差数列 vs ランダムウォーク



-
-
-
-

ここまで来たよ

① はじめに

- この授業どんなのり?

② ランダムウォーク

- ランダムウォーク
- 擬似乱数
- 擬似乱数の正しい/間違ったプログラム

擬似乱数列の生成

乱数列=ランダムな数列

$R(t+1)$ を C 言語でどう書く?

```
#include <stdlib.h>

/* 0以上 RAND_MAX 以下の正の整数をランダムに選んで返す関数 */
int rand();

/* その初期化 */
void srand(unsigned int seed);
```

RAND_MAX は M_PI みたいな定数. 値は処理系による. たとえば $2^{31} - 1$.

今の目的としては, 得られる値は, $+1, -1$ だけでいいんだけどな~

~>

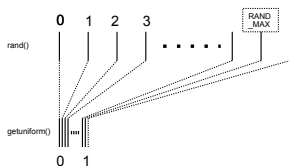
この授業の約束 (+世の中の習慣). `rand()` を生で使わず, いったん $[0, 1)$ 一様乱数 (0 と 1 の間の実数を同じ確率で返す) にして使う. →

`double getuniform()`

```
/* [0,1) 一様乱数 */  
double getuniform(){  
    return rand()/(1.0+RAND_MAX);  
}
```

`getuniform()` の性質

- ‘値域’ は $[0, 1)$. $0 \leq \text{getuniform}() < 1$.
- $(\text{getuniform}() < p \text{ となる確率}) = p$.



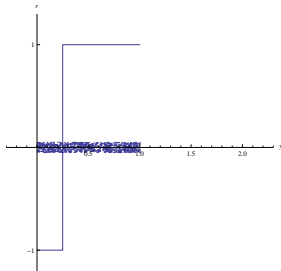
計算機の頭の中どうなってるの？

擬似乱数列 = 'ほぼ' ランダムな数列

ある確率で ± 1 を返したい!

```
/* 引数  $y$  が  $[0, 1)$  一様乱数なら,  $getrandom$  の返り値は  
   確率  $1/4$  で  $-1$ , 確率  $3/4$  で  $+1$ */
```

```
int getrandom(double y){  
    if( y < 0.25){  
        return -1;  
    } else {  
        return +1;  
    }  
}
```



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

ここまで来たよ

① はじめに

- この授業どんなのり?

② ランダムウォーク

- ランダムウォーク
- 擬似乱数
- 擬似乱数の正しい/間違っただプログラム

ソースコード 1: 乱数

```
1 /*
2 randl.c --- -1 or +1 を確率1/4, 3/4で選ぶ乱数
3 Time-stamp: "2013-04-09 Tue 18:57 JST hig"
4 */
5 #define _CRT_SECURE_NO_WARNINGS // VC++2008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 t; /* カウンタ */
16     int tmax=100; /* 擬似乱数を得る回数 */
17
18     scanf("%d",&seed);
19     srand(seed); /* シードの設定 */
20     for(t=0;t<tmax;t++){
21         /* srand(seed); */ /*ここに置くと? */
22         printf("%f\n",getrandom(getuniform()));
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 }
31
32 /** -1 or +1 を確率1/4, 3/4 で返す乱数 */
33 int getrandom(double y){
34     if( y < 0.25 ){
35         return -1;
36     } else {
37         return +1;
38     }
39 }
```

L01-Q1

Quiz(rand() の振る舞い)

次のプログラムで、A が出力される確率は？

```
if( getuniform()==getuniform() ){
    printf("A\n" );
} else {
    printf("B\n" );
}
```

- ① 0
- ② 0 に近い
- ③ 1/2
- ④ 1/2 くらい
- ⑤ 1 に近い
- ⑥ 1

L01-Q2

Quiz(rand() の振る舞い)

次のプログラムで、A が出力される確率は？

```
if( getuniform() < 0.1 ){  
    if( getuniform() < 0.2 ){  
        printf("A\n");  
    }  
}
```

- ① 0
- ② 0.02
- ③ 0.1
- ④ 0.2
- ⑤ 0.3
- ⑥ 1

L01-Q3

Quiz(擬似乱数の使いかた)

サンプルプログラムを参考に, 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)` を書こう.

返り値	確率
-1	1/3
0	1/2
+1	1/6

Hint: `getrandom()` のグラフは, なぜ, どのような形になるべき?

予習復習問題 次回の演習までの間には, まだ予習復習問題はありません.



manaba 出席カード提出

<https://attend.ryukoku.ac.jp>