ランダムウォークと離散型擬似乱数

樋口さぶろお

龍谷大学理工学部数理情報学科

計算科学☆実習 B L01(2017-04-10 Mon)

最終更新: Time-stamp: "2017-04-10 Mon 17:30 JST hig"

今日の目標

- ランダムウォークとは何か説明できる
- C で離散型擬似乱数を生成できる

http://hig3.net

ここまで来たよ

科目の目標

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, 確率過程とは何か, 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

どんな人のための科目?

計算科学☆実習 B を履修した方がいい人

- 確率過程 (=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) が見ていない, 残り半分の世界を確率論的モデルで見たい人
- 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, プログラム作成の中で実感したい人
- ランダムアルゴリズムが使えるようになりたい人
- 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☆実習 B 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, 数値計算法及び実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

科目ののり

注文が多くめんどくさい科目です…
成績計算 科目の成績 100 ピーナッツは

- 25 ピーナッツ:平常点．毎回授業での quiz, 授業時間外の予習復習.
  - だいたい 10 講義の Quiz ほか
  - だいたい 15 実習時間内の課題提出 TA の現場チェックでなく教員の提出プログラムチェック．TA は間違いの発見に努めますが,「それで OK」とは言いません.
- 50 ピーナッツ:プチテスト群
  - 15 紙のプチテスト
  - 35=5+15+15 プログラミング実技の非参照プチテスト
  - (15=プレゼンテーション, 実技の 15 の悪い方を上書きするのに使えます)
- 25 ピーナッツ:紙のファイナルトライアル (外部記憶あり)
- その他追加ピーナッツ．その時に説明.

ファイナルトライアル時点で 35 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.

欠席届 $\boxed{\text{典型的には介護等実習}}$ 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. ファイナルトライアルが締切). 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が, 自分で追いついてね.

$\boxed{\text{チーム活動のある回は, メンバーと樋口に欠席を事前に連絡, 分担を調整}}$

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.

担当者ののり

- なまえ: 樋口さぶろお `hig-compsci`@math.ryukoku.ac.jp
- へや: 1-502
- オフィスアワー: 月 6(1-502), 金 $4\frac{1}{2} - 5\frac{1}{2}$(1-502). 訪問歓迎な時間: 月火木金昼 (1-502). お弁当持参歓迎. お湯あげます.
- Web ページ: `http://hig3.net` 実習の指示や, スケジュールもここから.

科目の1週間のタイムライン

1. 月 15:20(締切) 予習復習問題 (e ラーニング) 回答何度でも．最高点.
2. 月 4 講義 (7-002), Quiz(参照あり)
3. このころ実習のタスク公開
4. 火 23:55 先週の課題の一部の提出締切
5. 水 3 実習 (1-609), Quiz 返却, 時間内の最初に e ラーニングの練習問題あるかも.
6. 水 23:55 今週の課題の一部の提出締切

実習室に行ったら, `http://hig3.net`→ 計算科学☆実習 B へ.
実習はイヤフォン必須.

ここまで来たよ

1. はじめに
   - この授業どんなのり?
2. ランダムウォークと離散型擬似乱数
   - ランダムウォーク
   - 擬似乱数
   - 離散型確率変数に対応する擬似乱数

C 言語で数列の計算

数列 $\{X(t)\}$, 時刻 $t = 0, 1, 2, \ldots$.　　数値計算法

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

階差数列 $R(t+1) =$ 定数 なら $X(t)$ は等差数列.

C 言語で数列を第 0 項から順に計算, 出力すると?

```
int x,r,t;

t=0;
x=a;

printf("%d\n",t,x);
for(t=0;t<100;t++){
  r=(階差数列の一般項 R(t+1));
  x=x+r; /* X(t+1) を求めた */
  printf("%d,%d\n",t+1,x);
}
```

ランダムウォーク (確率過程の例)

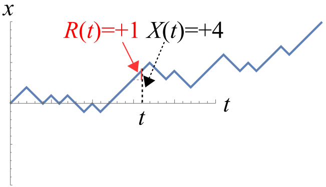
ランダムウォーク $\Leftrightarrow$ 階差数列 $R(t+1)$ が

現象の数理 A

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	$q(=1-p)$

塚田確率統計 §4.1 ベルヌーイ分布



ランダムウォークってどんなところに出てくる?

- 株価変動
-
-

ここまで来たよ

1. はじめに
   - この授業どんなのり?
2. ランダムウォークと離散型擬似乱数
   - ランダムウォーク
   - 擬似乱数
   - 離散型確率変数に対応する擬似乱数

離散型擬似乱数列の生成

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

(擬似) 乱数列

ある確率変数の標本になってる数列=ランダムな数列. コンピュータやサイコロや乱数表を使って作られる.

離散型擬似乱数列=ある離散型確率変数⋯ 塚田確率統計 §3.2

離散型確率変数の $R(t+1)$ の乱数を C 言語で生成しよう.

C 言語での乱数の使い方

```
#include <stdlib.h>

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

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

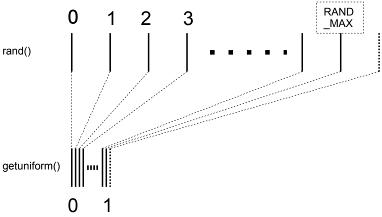
今の目的としては, 得られる値は, $+1, -1$ だけでいいんだけどな〜

$p = q = \frac{1}{2}$ だとしても,

偶数奇数で ± 1 にわけるのは「実は」危険

注: 現在は乱数を返すもっと高品質な関数があるが, rand, srand はどの C コンパイラでも提供されているので, 当面, これで考え方を説明.

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) 一様乱数にして使う.




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

[0, 1) 一様乱数は, 一様分布 U(0, 1) にしたがう連続型確率変数 X に対応.

確率密度関数 $f(x) = \begin{cases} 1 & (0 \le x < 1) \\ 0 & (他) \end{cases}.$

getuniform() の性質

- '値域' は $[0, 1)$ の実数. $0 \le \texttt{getuniform()} < 1$.
- $(0 \le \texttt{getuniform()} < r$ となる確率$) = \int_0^r 1\,\mathrm{d}x = r.\qquad (0 \le r \le 1)$

連続型確率変数の復習

確率統計☆演習 I(2016)L6　塚田確率統計 §3.3, §4.6

確率密度関数から事象の確率を求める

$$
\begin{aligned}
P(\text{事象}) = P(\text{条件}) &= \mathrm{E}[\mathbf{1}_{[\text{条件}]}(X)] \\
P(a \le X < b) &= \mathrm{E}[\mathbf{1}_{[a \le X < b]}(X)] \\
&= \int_{-\infty}^{+\infty} f(x) \mathbf{1}_{[a \le X < b]}(x)\, \mathrm{d}x = \int_a^b f(x)\, \mathrm{d}x
\end{aligned}
$$

L01-Q1

Quiz(連続的な確率変数の母平均値・母分散・母標準偏差・確率 (一様分布))

連続型確率変数 X は次の確率密度関数 $f(x)$ に従う.

$$f(x) = \begin{cases} 2 & (\frac{5}{2} \leq x < 3) \\ 0 & (\text{他}) \end{cases}$$

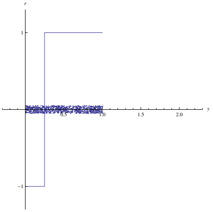
1. 母期待値 $\mathrm{E}[\cos(\pi X)]$ を求めよう.
2. 確率 $P(\frac{22}{8} < X < \frac{23}{8})$ を求めよう.

ある確率で ± 1 を返したい!

離散型確率変数 X.

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

```
/* 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());
```

ソースコード 1: 乱数

```
/*
rand1.c -- -1 or +1 を確率1/4, 3/4で選ぶ乱数
Time-stamp: "2013-04-09 Tue 18:57 JST hig"
*/
#define _CRT_SECURE_NO_WARNINGS // VC++2008用おまじない
#include <stdio.h>
#include <stdlib.h>   /* srand(), rand() を使うのに必要 */

/* 関数プロトタイプ宣言 */
double getuniform();
int getrandom(double y);

int main(){
  int seed;          /* 疑似乱数のシード */
  int t;             /* カウンタ */
  int tmax=100;      /* 疑似乱数を得る回数 */

  scanf("%d",&seed);
  srand(seed);          /* シードの設定 */
  for(t=0;t<tmax;t++){
    /* srand(seed);  */ /*ここに置くと? */
    printf("%f\n",getrandom(getuniform()));
  }
  return 0;
}

/** [0,1) 一様疑似乱数を返す */
double getuniform(){
  return rand()/(RAND_MAX+1.0);
}

/** -1 or +1 を確率1/4, 3/4 で返す乱数 */
int getrandom(double y){
  if( y < 0.25 ){
    return -1;
  } else {
    return +1;
  }
}
```

L01-Q2

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 `double 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

ここまで来たよ

1. はじめに
   - この授業どんなのり?
2. ランダムウォークと離散型擬似乱数
   - ランダムウォーク
   - 擬似乱数
   - 離散型確率変数に対応する擬似乱数

問題

L01-Q3

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.

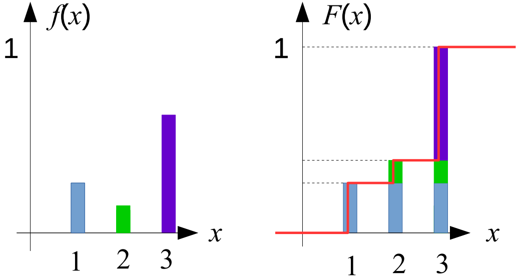
$$f(r) = \begin{cases} \frac{2}{8} & (r=1) \\ \frac{1}{8} & (r=2) \\ \frac{5}{8} & (r=3) \\ 0 & (\text{他}) \end{cases}.$$

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 $\int_a^b 1\,\mathrm{d}x$. $1,2,3$ について a,b をうまく調整していけばいい.

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.
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.

L01-Q4

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.

お知らせ

- 2017-04-12 水 3 実習 $\boxed{\text{教科書・イヤフォン持参}}$
- 2017-04-26 水 3 実習の春のプチテスト
- チューター/Math ラウンジ 月火水木昼 1-614
- 2017-06-18 統計検定

`https://manaba.ryukoku.ac.jp`