

実用的な乱数生成と品質の検証

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 L11(2018-07-17 Tue)

最終更新: Time-stamp: "2018-07-18 Wed 08:42 JST hig"

今日の目標

- 逆関数法, 棄却法で, 一様分布以外の分布にしたがう乱数が生成できる
- 乱数の品質とは何か説明できる
- 処理系ごとに高品質な乱数生成法を探せる



<http://hig3.net>

L10-Q1

Quiz 解答:確率変数の変換

$$f_Y(y) = \begin{cases} 1 & (0 \leq y < 1) \\ 0 & (\text{他}) \end{cases} \text{ なので,}$$

①

$$E[R] = E[2\sqrt{Y}] = \int_0^1 2\sqrt{y} \, dy = \frac{4}{3}.$$

$$V[R] = E[R^2] - (E[R])^2 = E[4Y] - \left(\frac{4}{3}\right)^2 = 2 - \left(\frac{4}{3}\right)^2$$

② $R = 2\sqrt{Y}$ を解くと, $y = (r/2)^2$ ($0 \leq y < 1, 0 \leq r < 2$) なので,
 $P(0.2 < R < 0.8) = P(0.01 < Y < 0.16) = 0.15.$

③

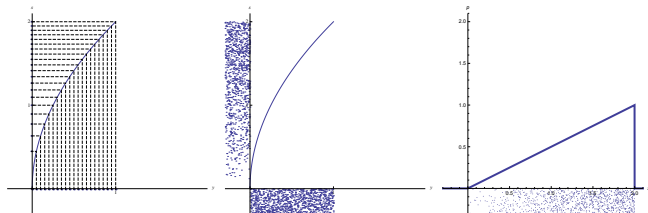
$$P(R < r) = P(Y < (r/2)^2) = \begin{cases} 0 & (r < 0) \\ r/2 & (0 \leq r < 2) \\ 1 & (2 \leq r) \end{cases}$$

4

$$F_R(r) = \int_0^r f_R(s) ds = (r/2)^2 \quad (0 \leq r < 2)$$

より, 両辺を微分して,

$$f_R(r) = \begin{cases} r/2 & (0 \leq r < 2) \\ 0 & (\text{他}) \end{cases}$$



L10-Q2

Quiz 解答: 確率変数の変換

$$f_Y(y) = \begin{cases} 1 & (0 \leq y < 1) \\ 0 & (\text{他}) \end{cases} \text{ なので,}$$

1

$$\begin{aligned} E[R] &= E[e^Y] = \int_{-\infty}^{+\infty} e^y f_Y(y) dy \\ &= \int_{-\infty}^0 e^y \cdot 0 dy + \int_0^1 e^y \cdot 1 dy + \int_1^{+\infty} e^y \cdot 0 dy \\ &= 0 + (e^1 - 1) + 0. \end{aligned}$$

$$\begin{aligned} V[R] &= E[R^2] - (E[R])^2 \\ &= E[e^{2Y}] - (E[e^Y])^2 \\ &= \int_0^1 e^{2y} \cdot 1 dy - (e^1 - 1)^2 \\ &= 0 + \frac{1}{2}(e^2 - 1) - (e^1 - 1)^2 \end{aligned}$$

② $r \leq 1$ のとき, $P(R < r) = 0$.

$1 < r < e$ のとき,

$$\begin{aligned} P(R < r) &= P(Y < \log r) = E[\mathbf{1}_{[\log Y < r]}(Y)] \\ &= \int_{-\infty}^{\log r} f_Y(y) dy = \log r \end{aligned}$$

$e \leq r$ のとき, $P(R < r) = 1$ よって,

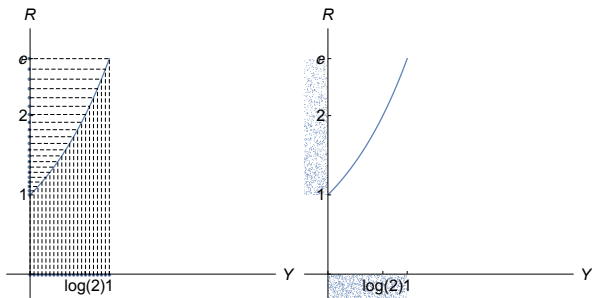
$$F_R(r) = \begin{cases} 0 & (r < 1) \\ \log r & (1 \leq r < e) \\ 1 & (e \leq r) \end{cases}$$

3

$$\int_{-\infty}^r f_R(r) dr = F_R(r)$$

より, 両辺を微分して,

$$f_R(r) = \begin{cases} 1/r & (1 \leq r < e) \\ 0 & (\text{他}) \end{cases}$$



L10-Q3

Quiz 解答: 確率変数の変換

- ① $Q \sim U(64, 100)$ より,

$$E[Q] = \int_{-\infty}^{+\infty} q f_Q(q) dq = \frac{64+100}{2} = 82.$$

$$V[Q] = \frac{1}{12}(100 - 64)^2 = 108.$$

②

$$P(Q > 82) = E[\mathbf{1}_{[Q>82]}(Q)] = \int_{82}^{100} \frac{1}{36} dq = \frac{1}{2}.$$

(別解) 一様分布で母平均値=中央値より大きい値を得る確率は $\frac{1}{2}$.

$$\textcircled{3} \quad f_R(r)dr = f_Q(q)dq, \quad r = q^{1/2}, \quad \frac{dr}{dq} = \frac{1}{2}q^{-1/2} \text{ より,}$$

$$\begin{aligned} f_R(r) &= \frac{1}{\frac{1}{2} \frac{1}{\sqrt{q}}} f_Q(q) = 2r f_Q(q) \\ &= \begin{cases} 2r \times \frac{1}{36} & (8 \leq r < 10) \\ 0 & (\text{他}) \end{cases} \end{aligned}$$

$$\textcircled{4} \quad \text{母平均値 } E[R] = \text{母期待値 } E[g(Q)] = E[\sqrt{Q}] \neq g(E[Q]).$$

$$E[R] = E[\sqrt{Q}] = \int_{-\infty}^{+\infty} \sqrt{q} f_Q(q) dq = \int_{64}^{100} \sqrt{q} \frac{1}{36} dq = \frac{244}{27}.$$

$$E[R^2] = E[Q] = 82.$$

$$V[R] = 82 - \left(\frac{244}{27}\right)^2 = \frac{242}{729}.$$

(別解) $f_R(r)$ を先に求めたなら

$$E[R] = \int_{-\infty}^{+\infty} r f_R(r) dr = \int_8^{10} r \frac{r}{18} dr = \frac{244}{27}.$$

$$E[R^2] = \int_{-\infty}^{+\infty} r^2 f_R(r) dr = \dots = 82.$$

5

$$P(R > 9) = P(R > g(81)) = P(Q > 81) = \frac{100-81}{36} = \frac{19}{36} \neq P(Q > 82).$$

(別解)

$$P(R > 9) = \int_9^{10} f_R(q) dr = \frac{19}{36}.$$

ここまで来たよ

10 連続型確率変数の変換

11 実用的な乱数生成と品質の検証

- 逆関数法による乱数生成
- 棄却法による乱数生成
- 標準正規分布にしたがう乱数の生成
- 乱数の品質

確率変数の関数

確率密度関数の変換のおぼえ方

西川確率統計注意 3.8

R, Q を確率変数, f_R, f_Q を確率密度変数, $r = g(q)$ を単調増加な関数とするととき, $f_Q(q) dq$ は変数変換しても不変: $f_R(r) dr = f_Q(q) dq$

$$f_R(r) = \frac{1}{\frac{dr}{dq}(q)} f_Q(q)$$

$$P(R < r(q)) = \int_{-\infty}^{r(q)} f_R(r) dr = \int_{-\infty}^q f_Q(q) dq = P(Q < q)$$

逆関数法

上の式を解いて求めた $r = g(y)$ を使うと, ほしい確率密度関数 $f_R(r)$ にしたがう R を, 一様分布 $U(0, 1)$ にしたがう Y から変換して生成できる.

L11-Q1

Quiz(逆関数法)

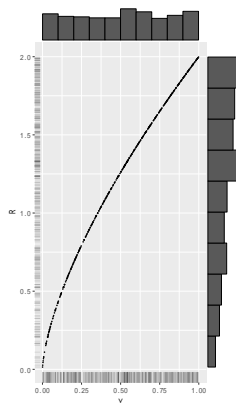
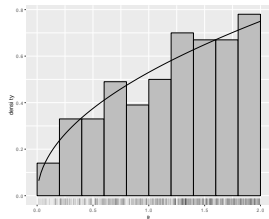
確率密度関数

$$f_R(r) = \begin{cases} \frac{3\sqrt{2}}{8}\sqrt{r} & (0 \leq r < 2) \\ 0 & (\text{他}) \end{cases}$$

に従う確率変数 R に対応する乱数を, $[0, 1)$ 一様乱数 y から $r = g(y)$ で作りたい. $g(y)$ を求めよう.


```
1 double getrandom(double y){  
2     return 2*pow(y,2.0/3);  
3 }
```

出力



L11-Q2

Quiz(逆変換法による擬似乱数生成)

次の確率密度関数 $f_R(r)$ に

$$f_R(r) = \begin{cases} -\frac{200}{21} \frac{1}{r^3} & (-5 \leq r < -2) \\ 0 & (\text{他}) \end{cases}$$

に従う確率変数 R に対応する乱数を, $[0, 1)$ 一様乱数 y から, 単調増加な $r = g(y)$ で作りたい. $g(y)$ を求めよう.

ここまで来たよ

10 連続型確率変数の変換

11 実用的な乱数生成と品質の検証

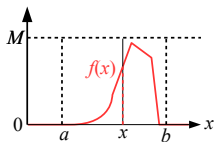
- 逆関数法による乱数生成
- 棄却法による乱数生成
- 標準正規分布にしたがう乱数の生成
- 乱数の品質

棄却法による乱数生成

連続型確率変数 X の確率密度関数 $f(x)$ が次を満たすとする。

$$0 \leq f(x) \leq M \quad (a \leq x < b)$$

$$f(x) = 0 \quad (x < a, b \leq x)$$



このとき、 X に対応する乱数を次で生成できる (効率とはもかく)。

棄却法

```

1  double getrandom(){
2     double x,y; double a,b,M=プログラマが設定;
3     while(1){
4         x=a+(b-a)*geuniform();
5         y=M*getuniform();
6         if( y < f(x) ){
7             break; /* 受理 accept */
8             /* 確率 f(x)/M=P(M*Y<f(r)), Y~U(0,M) */
9         } /* 棄却=reject, やりなおし */
10    }
11    return x;
12 }

```

だって、 x が返される確率密度って、 $f(r)$ に比例してるでしょ。

L11-Q3

Quiz(棄却法による乱数生成)

確率密度関数

$$f(x) = \begin{cases} 2(x-1) & (1 \leq x < 2) \\ 0 & (\text{他}) \end{cases}$$

に従う乱数を、棄却法を利用して返す関数 `double getrandom(void)` を C で書こう。

ただし、関数内では、 $[0, 1)$ 一様乱数を返す関数 `double getuniform()` を何度でも呼び出してよい。

平均して、1 回の実行 (乱数 1 個の生成) に、`double getuniform()` の何回の呼び出しを要するか考えよう。

(x, y) は $(1, 2) \times (0, 2)$ の面積 2 の長方形に一様に分布する。このうち、 $f(x) > y$ の面積 1 の領域の (x, y) に対しては x が返される。よって、`double getuniform()` 4 回の呼び出しに平均して 1 回乱数が返される。

ここまで来たよ

10 連続型確率変数の変換

11 実用的な乱数生成と品質の検証

- 逆関数法による乱数生成
- 棄却法による乱数生成
- 標準正規分布にしたがう乱数の生成
- 乱数の品質

標準正規分布にしたがう乱数の生成

逆関数法で、 $R \sim U(0, 1)$ から、 $X = g(R) \sim N(0, 1^2)$ となるような g があるといい。けどそんなうまい話はない。正規分布の確率密度関数が積分できないから。

正規分布は棄却法では $-a = b = \infty$ となっちゃう。

Python, R などでは、正規分布にしたがう乱数を生成する、ハイテクな逆関数法による関数があるのでそれを利用。

サンプルサイズ 1000 の標準正規分布にしたがう標本 in R

```
1 z<-rnorm(1000) # in R
```

サンプルサイズ 1000 の標準正規分布にしたがう標本 in Python

```
1 import numpy # in Python  
2 z=numpy.random.randn(1000)
```

サンプルサイズ 1 の標準正規分布にしたがう標本 in Excel

```
1 =norm.inv(rand(),0,1) // in Excel
```

ここまで来たよ

10 連続型確率変数の変換

11 実用的な乱数生成と品質の検証

- 逆関数法による乱数生成
- 棄却法による乱数生成
- 標準正規分布にしたがう乱数の生成
- 乱数の品質

double getuniform() を疑おう

これまで使っていた `double getuniform()` は本当に独立同分布 $U(0,1)$ にしたがっているのだろうか? → No.

```
1 #include <stdlib.h>
2 double getuniform(){
3     return rand()/(RAND_MAX+1.0);
4 }
```

見るからにだめ

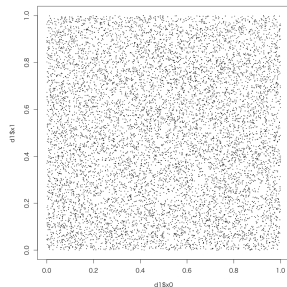
- 実数って言うけど, $1.0/(RAND_MAX+1.0)$ 幅の格子点.
- 乱数表の長さだけ呼び出したら, その後は繰り返しになる.

疑うべき点

- ヒストグラムは $U(0,1)$ に似てる? $P(\text{getrandom}() < y) = y$?
- $E[\text{getrandom}()] = 0.5$? $V[\text{getrandom}()] = 1/12.0$?
- $\text{getrandom}()$ と次の $\text{getrandom}()$ は独立同分布にしたがう?

擬似乱数列の独立性

$R(0) = \text{getuniform}()$ と $R(1)$ の
散布図



一様, 独立っぽい.

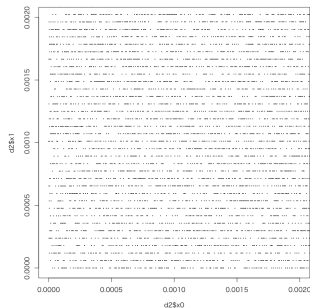
$N = 10000$ で 標本共分散
 $S_{R(0)R(1)} = 0.0002553512$.

西川確率統計定義 5.1

樋口さぶるお (数理情報学科)

しかし, 独立なら $\text{Cov}=0$ だけど,
 $\text{Cov}=0$ だから独立とは言えないの
だった.

もっとサンプルサイズを大きくし
て, $[0, 0.002) \times [0, 0.002)$ 部分を拡大.



実は `int rand()` は乱数表を見ているわけではない。

```
1  srand ( seed );
2  for ( t=0;t<100;t++){
3      print ("%d\n" , rand ( ));
4  }
```

は次と同じようなことをやってる。

線形合同法

```
1  int a=214013,b=2531011,m=32768; /* Visual C++*/
2  x=seed;
3  for ( t=0;t<100;t++){
4      x = (a*x + b)%m; /* ax + b mod m */
5      printf ("%d\n" , x);
6  }
```

前の項から次の項が決まる。これ自身、時系列
周期は最大でも $m \leq \text{RAND_MAX}$.

整数論 (離散数学)

今後、乱数を使うときのアドバイス

言い訳 `int rand()` は `stdlib.h` に入ってるから C では OS やコンパイラによらず使える.. そのため、この授業では `int rand()` から `double getuniform()` を作って使ってきた. しかし、`rand()` は低品質.

一般的なおすすめ

各処理系に一様分布や主要な確率分布にしたがう乱数生成関数が備わっている. 品質は様々.

- 近江崇宏先生 (東京大学) C 言語による乱数生成 http://www.sat.t.u-tokyo.ac.jp/~omi/random_variables_generation.html

MT=Mersenne Twister という数学的理論を使った高品質な乱数があれば/信頼できる 3rd party が提供していれば、それを使おう. 整数論 (離散数学)

- 松本真先生 (広島大学) Mersenne Twister <http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/mt.html>

いろんな処理系での一様乱数: `getuniform()` を置きかえよう

C on Linux, macOS `double drand48()` を `double getuniform()` のかわりに. MT ではない.

C++ `#include <random>` MT もある.

R `runif()` は MT.

Python `numpy.random.rand()` は MT.

Java `Math.random` メソッド, `java.util.Random` クラス. どちらも確率シミュレーションにたえる品質ではないので, 自分で MT を導入.

Excel `RAND()` 品質は怪しい.

お知らせ

提出場所

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



モバイルアプリ

<https://download.moodle.org/mobile>



通信量を抑えるスキャナアプリ CamScanner

<https://www.camscanner.com/>



今回の復習問題は任意. 100 ピーナッツに加算する 2 ピーナッツ x2 の実習的問題.
p121,p122.

- チューター/Math ラウンジ 月火水木昼 1-614
- 2018-07-20 金 2 補講 (自由参加)
- 2018-07-20 金 3 プロジェクト
- 2018-07-27 金 プレゼンテーション (15 ピーナッツ)
- 2018-07-31 火 ファイナルトライアル (筆記) 出題計画別紙.
- 2018-08-03 金 演習の期末試験はなし