2024 年度版

システムプログラミング実験

確率プログラミング　第 2 回

担当：中嶋 一貴

第 2 回 乱数生成法と確率過程

1　本日の目標

人など複数の独立した単体で構成される社会活動において，ばらつきを持つ個々の挙動は確率的にモデル化される．確率的な事象をシミュレーションする場合，その確率分布に従った乱数生成が基盤になる．本実験では，乱数生成の方法と確率統計の基本知識の習得を目指す．

2　実験の理論概要

2.1　確率分布の基礎

前回の実験では，一様分布に従う乱数の生成とシミュレーション応用を学んだ．本実験では，与えられた確率分布に従う乱数の生成法を学ぶ．まず，確率分布の基礎を述べる．実数値確率変数 X がある値 x を取る確率を $P(X = x)$ と書く．取りうる x の各値の確率全体を X の**確率分布** P_X という．取りうる値が離散値のとき離散確率分布といい，連続値のとき連続確率分布という．確率変数 X が x 以下になる確率の関数を**累積分布関数**といい，以下のように定義される．

$$F_X(x) = P(X \leq x) \tag{1}$$

確率変数 X が x より大きい値を取る確率の関数を **相補累積分布関数** といい，以下のように定義される．

$$\bar{F}_X(x) = P(X > x) \tag{2}$$

式 (1) と式 (2) は以下の関係にある．

$$\bar{F}_X(x) = 1 - \bar{F}_X(x)$$

また，確率分布 P_X が連続確率分布のとき，実数 a, b に対して

$$P(a \leq X \leq b) = \int_a^b f_X(x)dx$$

を満たす関数 f_X を確率密度関数という．

以下では，代表的な連続確率分布を紹介する．

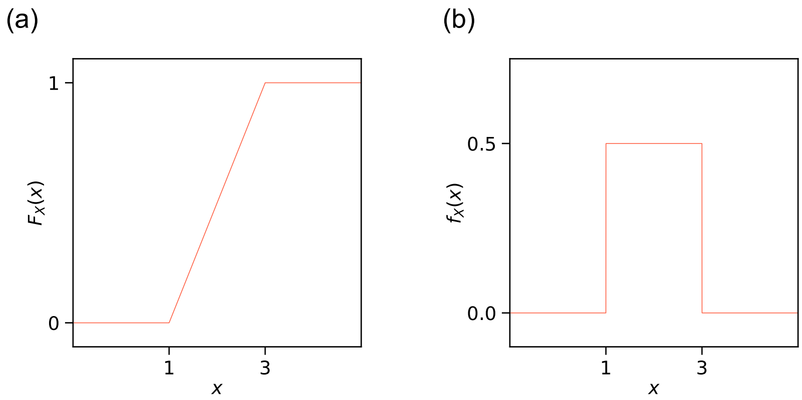

図 1: $(a, b) = (1, 3)$ のときの一様分布．(a) 累積分布関数．(b) 確率密度関数.

2.1.1　一様分布

実数 a, b が与えられたとき，確率変数 X が区間 $[a, b]$ 内の取りうる全ての値 x の確率が等しい分布である．累積分布関数 $F_X(x)$, 相補累積分布関数 $\bar{F}_X(x)$, および確率密度関数 $f_X(x)$ は以下である．なお，$-\infty < x < \infty$ である．

$$F_X(x) = \begin{cases} 0 & \text{for } x < a, \\ \frac{x-a}{b-a} & \text{for } a \leq x \leq b, \\ 1 & \text{for } x > b. \end{cases}$$

$$\bar{F}_X(x) = \begin{cases} 1 & \text{for } x < a, \\ \frac{b-x}{b-a} & \text{for } a \leq x \leq b, \\ 0 & \text{for } x > b. \end{cases}$$

$$f_X(x) = \begin{cases} 0 & \text{for } x < a \text{ or } x > b, \\ \frac{1}{b-a} & \text{for } a \leq x \leq b. \end{cases}$$

図 1 は $(a, b) = (1, 3)$ のときの一様分布の累積分布関数と確率密度関数を示す．

また，期待値と分散はそれぞれ以下である．

$$E[x] = \frac{1}{2}(a+b),$$
$$V[x] = \frac{1}{12}(b-a)^2.$$

2.1.2　指数分布

与えられた定数 $\lambda > 0$ に対して，指数分布の累積分布関数 $F_X(x)$, 相補累積分布関数 $\bar{F}_X(x)$, および確率密度関数 $f_X(x)$ は以下である．なお，$0 \leq x < \infty$ である．

$$F_X(x) = 1 - e^{-\lambda x},$$
$$\bar{F}_X(x) = e^{-\lambda x},$$
$$f_X(x) = \lambda e^{-\lambda x}.$$

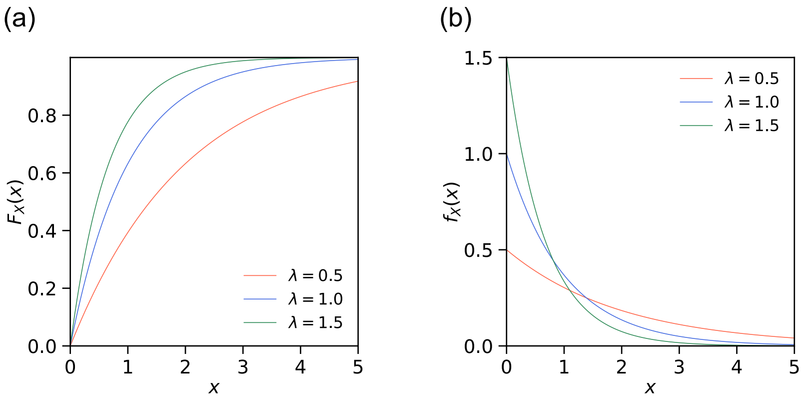

図 2: $\lambda = 0.5, 1.0, 1.5$ のときの指数分布．(a) 累積分布関数．(b) 確率密度関数．

図 2 は $\lambda = 0.5, 1.0, 1.5$ それぞれのときの指数分布の累積分布関数と確率密度関数を示す．また，期待値と分散はそれぞれ以下である．

$$
\begin{aligned}
E[x] &= \frac{1}{\lambda}, \\
V[x] &= \frac{1}{\lambda^2}.
\end{aligned}
$$

2.2　線形合同法

一様分布に従う確率変数を一様乱数という．ここでは，一様乱数を生成する 1 つの方法である線形合同法について述べる．$A < M, B < M, A > 0, B \geq 0$ を満たす定数 A, B, M をとる．乱数の種 X_0 を与えて，i 個目の乱数 X_i を以下の漸化式に従って生成する．

$$
X_{i+1} = (A \times X_i + B) \bmod M.
$$

なお，整数 m, n に対して m を n で割った余りを $m \bmod n$ と書く．各乱数 X_i は 0 以上 $M - 1$ 以下の整数である．

生成例を述べる．$A = 5, B = 7, M = 19$，$X_0 = 6$ とする．1 個目の乱数 X_1 は，

$$
\begin{aligned}
X_1 &= 5 \times 6 + 7 \bmod 19 \\
&= 18.
\end{aligned}
$$

となる．2 個目の乱数 X_2 は，

$$
\begin{aligned}
X_2 &= 5 \times 18 + 7 \bmod 19 \\
&= 2.
\end{aligned}
$$

となる．

生成される乱数列 $\{X_i\}_i$ は周期性を持つ．上記の例では，$18 \to 2 \to 17 \to 16 \to 11 \to 5 \to 13 \to 15 \to 6$ を繰り返す．周期は最大で M であり，以下の条件が満たされる場合に最大周期 M を持つ．

1. B と M が互いに素である．

2. $A-1$ が，M の持つ全ての素因数で割りきれる．

3. M が 4 の倍数である場合，$A-1$ も 4 の倍数である．

2.3　逆関数法

所望の確率分布に従う乱数を生成させる 1 つの方法として逆関数法がある．累積分布関数の逆関数を用いて，一様分布に従う確率変数から所望の確率分布に従う確率変数を生成させる．

所望の確率分布に従う確率変数を X とし，その累積分布関数を $F_X(x) = P(X \leq x)$ とする．$y = F_X(x)$ が連続な単調増加関数であれば，逆関数 $F^{-1}(y)$ が存在する．U を $[0,1]$ 上の一様分布に従う確率変数として，

$$X = F^{-1}(U)$$

と定義すれば，X は累積分布関数 F_X を持つ確率分布に従う確率変数となる．このことは

$$P(F^{-1}(U) \leq x) = P(U \leq F_X(x)) = F_X(x)$$

となることから確認できる．

逆関数法は，原理的には連続確率分布と離散確率分布に適用可能であるが，任意の確率分布に対して逆関数が容易に求まるわけではない．逆関数が容易に求まる例として，指数分布に従う乱数生成を紹介する．$\lambda > 0$ の指数分布に従う確率変数 X を生成させたいとする．その累積分布関数は，

$$F_X(x) = 1 - e^{-\lambda x}$$

であり，逆関数は

$$F^{-1}(y) = -\frac{1}{\lambda}\log_e(1-y)$$

である．したがって，λ の指数分布に従う確率変数 X を生成するには，$[0,1)$ 上の一様分布に従う確率変数 U をとり，

$$X = -\frac{1}{\lambda}\log_e(1-U)$$

と定義すればよい．なお，log 関数の定義上，U が値 1 をとらないように U を定義していることに注意する．

3　確率過程

確率過程とは，確率的法則に従って時間などの条件によって変化する確率変数の数理モデルである．株価や為替の変動などを数学的に記述するモデルとして利用される．ここでは，離散的確率過程の 1 つであるランダムウォークを考える．ランダムウォークでは，時刻 t にある地点にいるとき，時刻 $t+1$ の位置がありうる選択肢からランダムに決定される確率過程である．数直線上のある点に対して，$t=0$ の地点を 0 とする．$0<p<1$, $d>0$ に対して，時刻 $t=0,1,2,\ldots$ で確率 p で正の方向に d，確率 $1-p$ で負の方向に d 進むとする．ここで，時刻 t に点が正の方向に進む距離を X_t とすると，

$$P(X_t = d) = p$$
$$P(X_t = -d) = 1-p$$

であり，時刻 t における点の位置 S_t は

$$S_t = X_1 + X_2 + \cdots + X_t$$

となる．$\{S_t\}_t$ を 1 次元ランダムウォークという．S_t の期待値は

$$\begin{aligned}
E[S_t] &= E\left[\sum_{i=1}^{t} X_i\right] \\
&= \sum_{i=1}^{t} E[X_i] \qquad (3) \\
&= \sum_{i=1}^{t} [P(X_i = d) \times d + P(X_i = -d) \times (-d)] \\
&= 0
\end{aligned}$$

となる．式 (3) では期待値の線形性を用いた．次に S_t の分散を求める．

$$\begin{aligned}
V[S_t] &= E[(S_t - E[S_t])^2] \\
&= E[S_t^2] \\
&= E[(X_1 + X_2 + \cdots + X_t)^2] \\
&= E\left[\sum_{i=1}^{t} X_i^2 + 2 \sum_{1 \le i \le j \le t} X_i X_j\right] \\
&= \sum_{i=1}^{t} E[X_i^2] + 2 \sum_{1 \le i \le j \le t} E[X_i X_j]
\end{aligned}$$

ここで

$$\begin{aligned}
E[X_i^2] &= P(X_i = d) \times d^2 + P(X_i = -d) \times d^2 \\
&= d^2 \\
E[X_i X_j] &= E[X_i]E[X_j] \\
&= 0 \qquad (4)
\end{aligned}$$

式 (4) は時刻 i に進む距離と時刻 j に進む距離が独立であることから成り立つ．よって

$$V[S_t] = td^2.$$

課題 2-1: 逆関数法による乱数生成

逆関数法を用いてパラメータ $\lambda > 0$ の指数分布に従う乱数を独立に n 個生成する関数を作成せよ．関数名は rnd_exp とし，λ と n を引数として受け取るとする．

なお，次の手順に従って関数を作成すること．

1. 区間 $[0, 1)$ の一様乱数を 1 個生成する処理を書く．なお，各言語の乱数ライブラリの関数を利用してよい．例えば C++ では乱数ライブラリ 'std::random' 内の関数 'std::random_device', 'std::mt19937' および 'std::uniform_real_distribution' を利用してよい．
2. 区間 $[0, 1)$ の 1 つの一様乱数と λ が与えられ，λ の指数分布に従う乱数を 1 個生成する処理を書く．2.3 章をよく読むこと．
3. 上記の処理を組み合わせて，λ の指数分布に従う乱数を独立に n 個生成する処理を書く．

レポートには，作成した関数のソースファイル名を言及すること．例えばファイル名が 'xyz.cpp' の場合，'xyz.cpp を参照のこと．' と書けばよい．ファイルが複数ある場合，全てのファイル名を言及すること．なお，考察を書く必要はない．

課題 2-2: 逆関数法による乱数生成

課題 1-1 で作成した関数 rnd_exp が λ の指数分布に従う乱数を生成しているか，以下の 2 つの観点から，2.2.2 章で述べた理論値と比較することによって検証せよ．

1. 乱数の平均値と分散．

2. 乱数の分布．

レポートには，$\lambda = 1.0, 1.5, 2.0$ の 3 通りそれぞれについて，関数 rnd_exp を用いて生成した n =10,000 個の乱数に対する検証結果と考察を記載せよ．乱数の平均値と分散のシミュレーション結果は，有効数字 3 桁で報告すること．また，確率密度関数とともに乱数の確率密度をプロットした図をレポートに載せること（横軸の範囲は $0 \leq x \leq 5$ とせよ）．$\lambda = 0.5$ の場合のプロット例を図 3 に示す．

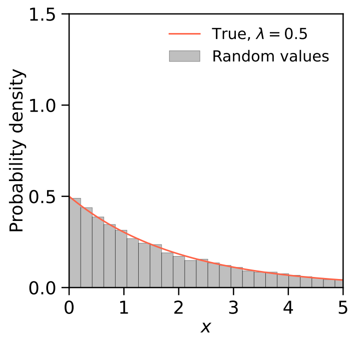

図 3: $\lambda = 0.5$ の指数分布の確率密度関数（‘True, $\lambda = 0.5$’ のラベル）と，逆関数法により生成された乱数の確率密度（‘Random values’ のラベル）．

課題 2-3: ランダムウォーク

3 章のランダムウォークのシミュレーションを実行せよ．その際，$p = 0.5, d = 1$ とし，1 回のランダムウォークは 1,000 ステップ（つまり $t = 0, 1, \dots, 1,000$）とすること．レポートには以下の 3 つを記載せよ.

1. 1 回のランダムウォークのシミュレーションで得られた点の位置 S_t $(t = 0, 1, \dots, 1,000)$ の軌跡の図．図 4 に軌跡のプロット例を示す.

2. 100 回の独立なシミュレーションで得られる $t = 1,000$ における点の位置 $S_{1,000}$ の平均値と分散（有効数字 3 桁）.

3. シミュレーション結果の考察.

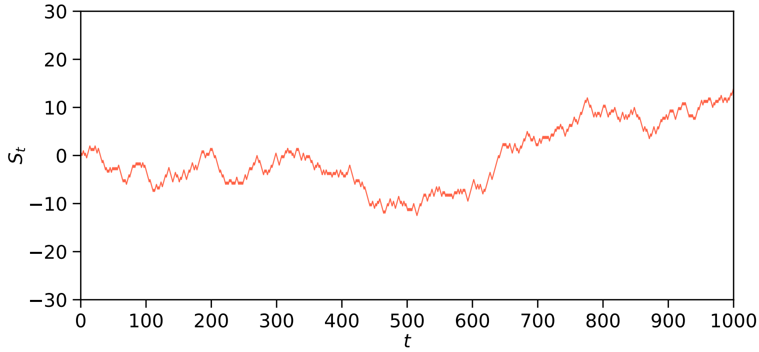

図 4: $p = 0.5, d = 0.5$, ステップ数 1,000 の 1 回のランダムウォークの軌跡．横軸は時刻 t, 縦軸は時刻 t の点の位置 S_t を示す.

課題 2-4: 線形合同法

一様乱数の生成方法としてメルセンヌ・ツイスタという手法が広く用いられている．これに対して，線形合同法はいくつかの欠点を持つことが知られている．欠点の 1 つとして，連続する乱数の組（例えば組 (X_i, X_{i+1})）は一様に分布しないことが挙げられる．

$n = 100,000$ とする．線形合同法とメルセンヌ・ツイスタそれぞれの方法を用いて n 個の乱数 $\{X_i\}_{i=1}^{n}$ を独立に生成する．ここで，各乱数 X_i は 0 以上 $M-1$ 以下の整数とする．そして，それぞれの乱数列に対して，$n-1$ 個の乱数の組の列 $\{(X_i, X_{i+1})\}_{i=1}^{n-1}$ を求める．レポートには，次の 4 つを記載せよ．

1. 実験で用いた，定数 M の値および線形合同法の定数 X_0, A, B の値．
2. それぞれの方法で生成された乱数の組の集合をプロットした図．図 5 にプロット例を示す．
3. それぞれの方法で生成された固有の乱数の組の数．なお，組の数の最大値は M^2 である．
4. シミュレーション結果の考察．

次の手順に従って取り組むとよい．

1. 線形合同法により 0 以上 $M-1$ 以下の乱数を n 個生成する関数 rnd_lgc を作成する．2.2 章をよく読むこと．
2. メルセンヌ・ツイスタにより 0 以上 $M-1$ 以下の整数の乱数を n 個生成する関数 rnd_mt を作成する．なお，各言語の乱数ライブラリの関数を利用してよい．例えば C++ では乱数ライブラリ 'std::random' 内の関数を利用してよい．
3. $n = 100,000$ とする．rnd_lgc を用いて 0 以上 $M-1$ 以下の乱数を独立に n 個生成する．同様に，rnd_mt を用いて 0 以上 $M-1$ 以下の乱数を独立に n 個生成する．
4. 生成したそれぞれの乱数列 $\{X_i\}_{i=1}^{n}$ に対して，$n-1$ 個の乱数の組 $\{(X_i, X_{i+1})\}_{i=1}^{n-1}$ がいくつの固有の組を生成するか，調べる．

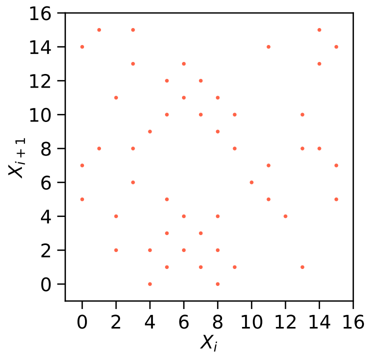

図 5: メルセンヌ・ツイスタにより生成された乱数の組 $\{(X_i, X_{i+1})\}_{i=1}^{n}$ の集合のプロット例．なお，このプロット例では，$M = 16$ とし，n を 100,000 より非常に小さい数で取っている．

レポート提出方法

以下を kibaco の課題ページから提出すること．

- 課題の内容を記載したレポート．
  - 課題 2-1, 2-2, 2-3, 2-4 の内容は必須である．各課題について，回答が正しくない場合でも，自力でコードの実装・数値結果の出力・結果の検証と考察がされていれば，部分点が付与される．なお，白紙や無回答の場合，その課題の得点は 0 点となる．
  - ファイル名は “apl_pp_xxx_yyy.pdf” とする．xxx には学修番号，yyy には氏名を入力する．例えば学修番号が 21012345 で氏名が Kazuki Nakajima の場合，“apl_pp_21012345_kazuki_nakajima.pdf” とする．漢字やひらがなをファイル名に含めないこと．
- 各課題を解くために使用したソースコード．
  - ソースコードが複数に分かれている場合は，zip ファイルにまとめて提出すること．
  - 各ソースコードが，誰のもので，どの課題に対応しているかわかるようにファイル名を設定すること．例えば，氏名が Kazuki Nakajima で課題 1-1 に対する C++ コードのファイル名は “kadai_1_1_kazuki_nakajima.cpp” とする．

注意事項

- **本課題の提出締め切りは 1 週間後（2024/11/06）の 12:00 である．**
- レポート作成時には，テクニカルライティングの資料を確認し，全体をよく検証してから提出すること．