情報数学II
第9回 確率・統計2

小野順貴
(ONO, Nobutaka)
東京都立大学　システムデザイン学部
情報科学科　教授
onono@tmu.ac.jp

任意の確率密度分布に従う乱数の生成

- 一様分布は計算機で比較的生成しやすい
- これに対し、様々な工学上の問題のシミュレーションや数値実験において、任意の確率密度分布に従う乱数を生成したいこともよく生じる。
- よく用いられる2つの手法
  - 逆関数法
  - 棄却法

逆関数法の原理

- 変数変換により確率密度分布を変形する
- 積分における変数変換(復習)
  - $y = f(x)$

$$\underline{p(y)dy} = p(f(x))dy = p(f(x))\frac{dy}{dx}dx = \underline{p(f(x))f'(x)}dx$$

yの確率密度関数　　　　xの確率密度関数

逆にかくと

$$\begin{array}{rccc} \text{変数の変換:} & x & \rightarrow & y = f(x) \\ \text{確率密度関数の変換:} & p(f(x))f'(x) & \rightarrow & p(y) \end{array}$$

$$\begin{array}{rccc} \text{変数の変換:} & y & \rightarrow & x = f^{-1}(y) \\ \text{確率密度関数の変換:} & p(y) & \rightarrow & p(f(x))f'(x) \end{array}$$

逆関数法

- 確率密度分布 $p(x)$ に従う乱数の生成法
  1. 累積分布関数を求める
  $$F(x) = \int_{-\infty}^{x} p(x)dx$$
  2. [0,1]上の一様分布に従う乱数 y を生成する
  3. $F(x)$ の逆関数を用いて $x = F^{-1}(y)$ と変換する

  $\rightarrow$ このとき、x は $p(x)$ に従う

逆関数法の証明

２ページ前の変数変換を見ながら以下を用いると

$$u(y) = \begin{cases} 1 & (0 \leq y \leq 1) \\ 0 & (\text{otherwise}) \end{cases}$$

$$F(x) = \int_{-\infty}^{x} p(x)dx$$

確率変数 x が従う確率密度分布は

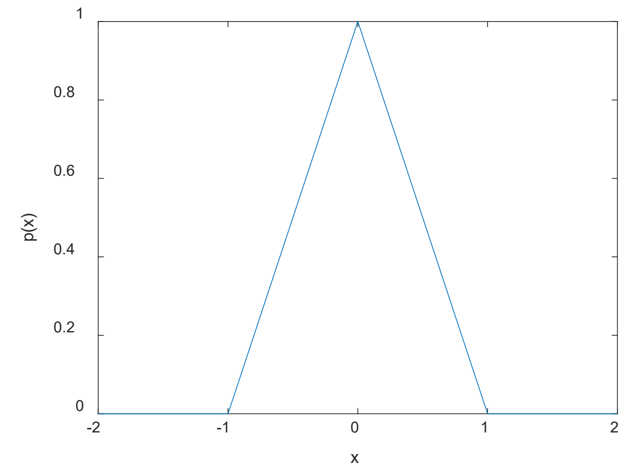
$$\underline{u(F(x))}F'(x) = p(x)$$

F(x)は累積密度関数なので常に
0≦F(x)≦1
よって u(F(x))=1

逆関数法の例題(1/2)

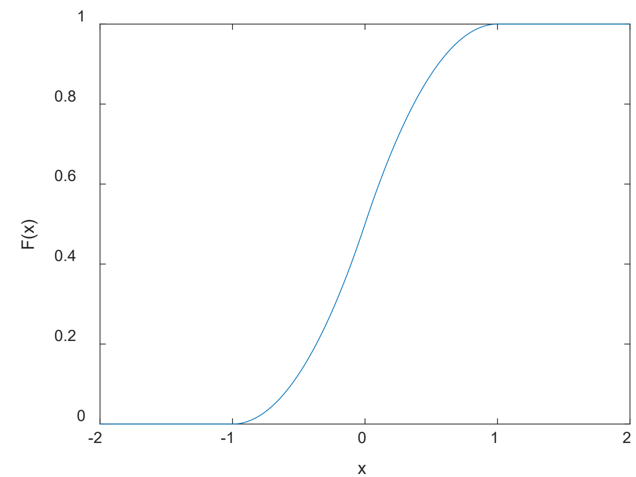
- 以下の確率密度分布に従う乱数を生成する。

$$p(x) = \begin{cases} 0 & (x < -1) \\ x+1 & (-1 \le x \le 0) \\ -x+1 & (0 \le x \le 1) \\ 0 & (1 < x) \end{cases}$$



- 累積密度分布の計算

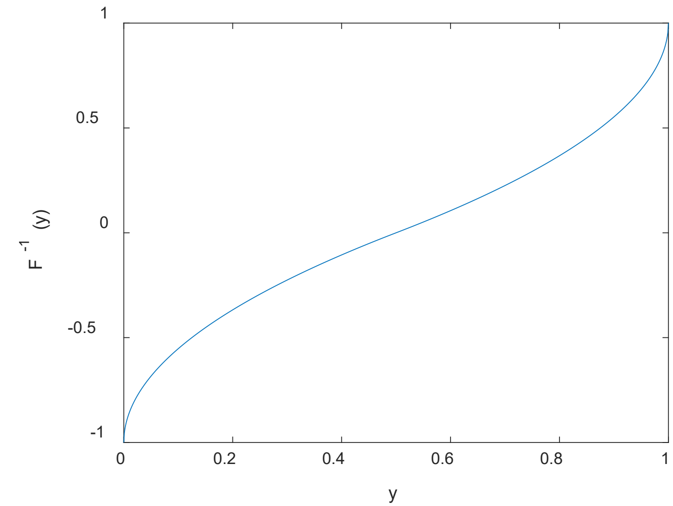
$$\begin{aligned} F(x) &= \int_{-\infty}^{x} p(x)dx \\ &= \begin{cases} 0 & (x < -1) \\ (1/2)(x^2+2x+1) & (-1 \le x \le 0) \\ -(1/2)(x^2-2x-1) & (0 \le x \le 1) \\ 1 & (1 < x) \end{cases} \end{aligned}$$



逆関数法の例題(2/2)

- 累積密度分布の逆関数
  - $y = (1/2)(x^2 + 2x + 1)$ などを x について解く

$$F^{-1}(y) = \begin{cases} \sqrt{2y} - 1 & (0 \leq y \leq 1/2) \\ -\sqrt{2 - 2y} + 1 & (1/2 \leq y \leq 1) \end{cases}$$



MATLABによる場合分け

- MATLABでは、配列の要素の値による場合分けの計算を、if文を使わずに行うことができる
- 例：

```
>> A=[1,2,3,4,5];
>>(A<3)              % 括弧付きで配列に対する条件をかく
ans =
  1×5 の logical 配列
   1   1   0   0   0   ← (A<3)という論理式の真偽値が得られる
```

- 使用例1：

```
>> A(A>3)=7;         % A>3である要素を全部 7にする
```

- 使用例2：

```
>> A=[1,2,3,4,5];
>> B=zeros(size(A));
>> flag=(A<3);
>> B(flag)=A(flag).^2; % A>3である要素に対し、B=A^2とする
```

MATLAB練習17:逆関数法

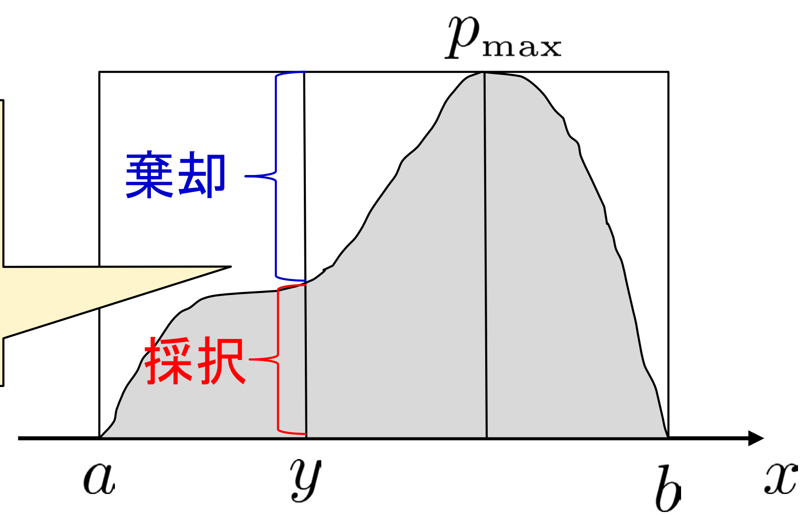
```
>> y=rand(1,10000);                 % [0,1]上の一様分布に従う乱数を10000個生成
>> x=zeros(size(y));
>> flag=(y<0.5);
>> x(flag)=sqrt(2*y(flag))-1;       % y<0.5であれば、x=sqrt(2y)-1
>> flag=(y>=0.5);
>> x(flag)=-sqrt(2-2*y(flag))+1;    % y>=0.5であれば、x=sqrt(2-2y)+1
>> histogram(x,40);                 % 乱数 y のヒストグラムを確認
```

棄却法：一様分布を用いる場合

- $[a, b]$ 上の確率密度分布 $p(x)$ に従う乱数の生成法

0. $p(x)$ の最大値を $p_{\max}$ とする
1. $[a, b]$上の一様分布に従う乱数 y を生成する
2. $[0, p_{\max}]$ 上の一様分布に従う乱数 u を生成する
3. $u \leq p(y)$だったら y を採用、そうでなかったら y は棄却して 1. に戻ってやり直し

生成した乱数yを
適当な割合で捨てる
（棄却する）ことで
目標の確率密度に
あわせている



棄却率が大きいと
効率が悪い
→
なるべく棄却しないで生成したい

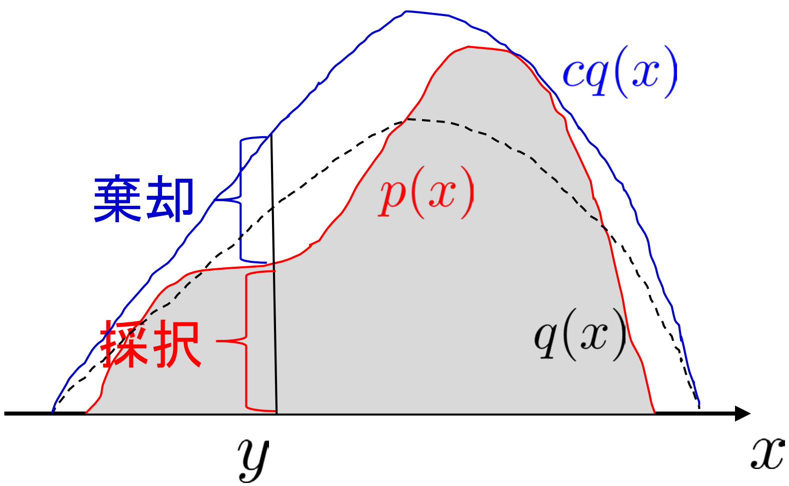
棄却法：一般の確率密度分布を用いる場合

- 一様分布に限らず、確率密度関数 $p(x), q(x)$が
適当な実数 c を用いて

$$p(x) \leq cq(x)$$

を満たすならば、$q(x)$ に従う乱数から、以下のようにして
$p(x)$ に従う乱数を生成できる

1. $q(y)$ に従う変数 y を生成する
2. $[0, cq(y)]$ 上の一様分布に従う乱数 u を生成する
3. $u \leq p(y)$ だったら y を採用、
そうでなかったら y は棄却して
1. に戻ってやり直し



p(x)に近いq(x)を選べると
効率を改善できる

MATLAB練習18:棄却法

- p.6の確率密度分布に従う乱数を棄却法で生成してみる

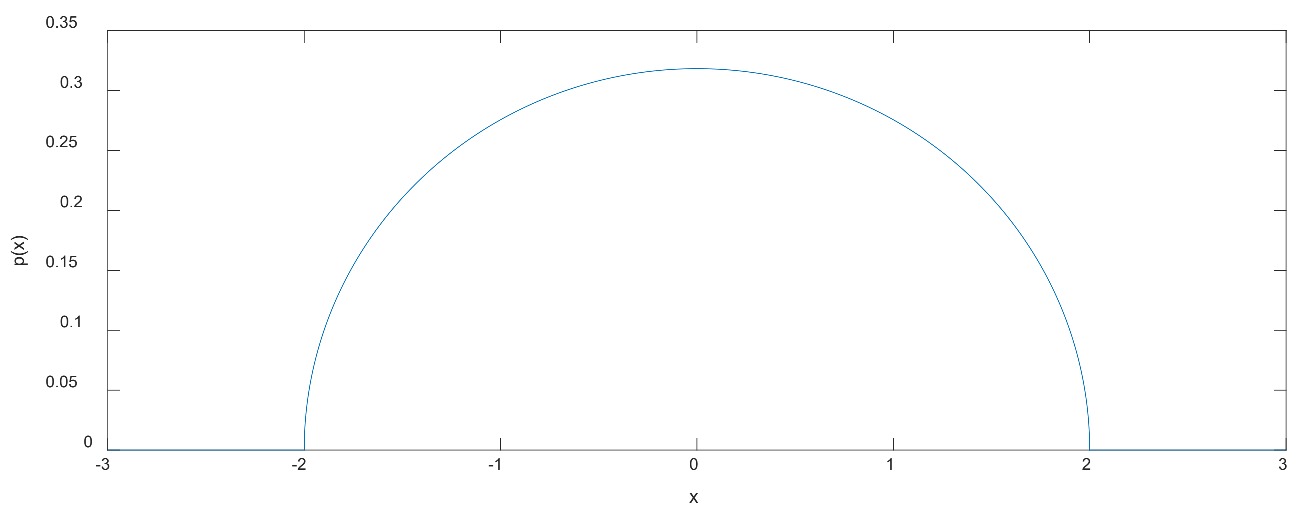
```
a=-1; b=1; N=10000; pmax=1; i=1;
while i<=N
  y=rand()*(b-a)+a;
  u=rand()*pmax;
  if (y<0)
    py=y+1;
  else
    py=-y+1;
  end
  if u<py
    ys(i)=y;
    i=i+1;
  end
end
histogram(ys,40)
```

赤字は密度関数によって変える部分
(a, b): xが取りうる値の上限下限
pmax: p(x)の最大値
py: p(x)の関数形を記述。p.10にあわせてyで書いていることに注意
p

MATLAB演習09

- 棄却法により、以下の確率密度分布に従う乱数を生成せよ。

$$p(x) = \begin{cases} 0 & (x < -2) \\ \dfrac{1}{2\pi}\sqrt{4 - x^2} & (-2 \leq x \leq 2) \\ 0 & (2 < x) \end{cases}$$



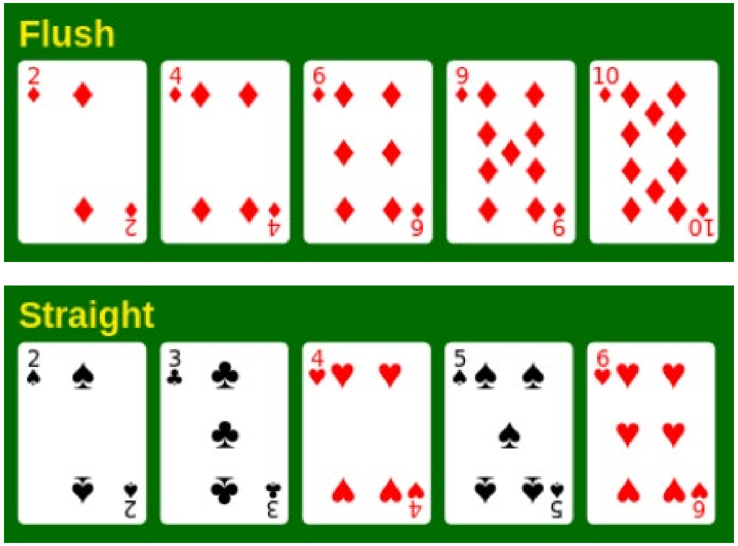
乱数を用いたシミュレーション

- 実際に確率を計算するのが困難、もしくは複雑すぎる場合には、理論的な確率計算の代わりに、乱数を用いた多数の試行を計算機上で行い、確率を見積もる手法がとられる。
- モンテカルロ法とも呼ばれる。

例題

- ポーカーで手札が配られたときに
  - フラッシュ（スートが同一）
  - ストレート（数字が連続）

  である確率はそれぞれ
  どの程度か？



（Wikipediaより抜粋）

- 発展問題
  - ストレートフラッシュの
    確率を求めてみよ
    （上の2つの事象は独立ではないので
    単純な積にはならない）

フラッシュの確率シミュレーション

```
c(1, 1:13)=1;
c(1,14:26)=2;
c(1,27:39)=3;
c(1,40:52)=4;

N=100000;        % 試行回数
count=0;         % フラッシュの回数を数える変数
for i=1:N
  idx=randperm(52);        % 乱数で置換を生成
  h(1,1:5)=c(1,idx(1:5)); % ５枚カードをひくのをシミュレーション

  % フラッシュの判定
  if (h(1,1)==h(1,2) & h(1,1)==h(1,3) & h(1,1)==h(1,4) & h(1,1)==h(1,5))
    count=count+1;
  end
end
```

ストレートの確率シミュレーション

```
n(1, 1:13)=1:13;
n(1,14:26)=1:13;
n(1,27:39)=1:13;
n(1,40:52)=1:13;

N=100000;       % 試行回数
count=0;        % ストレートの回数を数える変数
for i=1:N
  idx=randperm(52);      % 乱数で置換を生成
  h(1,1:5)=n(1,idx(1:5)); % 5枚カードをひくのをシミュレーション

  % ストレートの判定
  h=sort(h);
  if (h(1,1)==h(1,2)-1 & h(1,2)==h(1,3)-1 & h(1,3)==h(1,4)-1 & h(1,4)==h(1,5)-1)
    count=count+1;
  end
end
```

まとめ

- 任意の確率密度に従う乱数の生成
  - 逆関数法
  - 棄却法
- 乱数を用いたシミュレーション