

# 第2章

## 確率統計の基礎と Octaveの使い方

佐々木稔

# 確率関数と確率密度関数

- 確率変数

- 事象 (パターン  $\mathbf{x}$ , カテゴリ  $y$ ) 内で取りうる値

- 離散型確率変数  $y$

- $p(y)$  : 確率関数 (生起確率)

- $$p(y) \geq 0 \text{ for } y = 1, \dots, c \quad \sum_{y=1}^c p(y) = 1$$

- 連続型確率変数  $\mathbf{x}$

- $p(\mathbf{x})$  : 確率密度関数

- 離散型確率変数の場合は生起確率

- $$p(\mathbf{x}) \geq 0 \text{ for any } \mathbf{x} \in D \quad \int_D p(\mathbf{x}) d\mathbf{x} = 1$$

# 同時確率

- 同時確率  $p(\mathbf{x}, y)$ 
  - $\mathbf{x}$  と  $y$  が同時に生起する確率
- 周辺確率
  - 同時確率で一方の確率変数を積分したもの

$$\sum_{y=1}^c p(\mathbf{x}, y) = p(\mathbf{x}), \quad \int_D p(\mathbf{x}, y) d\mathbf{x} = p(y)$$

# 条件付き確率

- $y$  の条件付き確率  $p(y|\mathbf{x})$ 
  - $\mathbf{x}$  が与えられたもとで  $y$  が生起する確率
- $\mathbf{x}$  の条件付き確率  $p(\mathbf{x}|y)$ 
  - $y$  が与えられたもとで  $\mathbf{x}$  が生起する確率
- 条件付き確率の特徴

$$p(y|\mathbf{x})p(\mathbf{x}) = p(\mathbf{x}, y) = p(\mathbf{x}|y)p(y)$$

- ベイズの定理

$$p(y|\mathbf{x}) = \frac{p(\mathbf{x}|y)p(y)}{\sum_{y'=1}^c p(\mathbf{x}|y')p(y')}$$

# 期待値、分散共分散行列

- 期待値

$$E_{\mathbf{x}}[\mathbf{x}] = \int \mathbf{x}p(\mathbf{x})d\mathbf{x}$$

- 分散共分散行列

$$V_{\mathbf{x}}[\mathbf{x}] = E_{\mathbf{x}} \left[ (\mathbf{x} - E_{\mathbf{x}}[\mathbf{x}])(\mathbf{x} - E_{\mathbf{x}}[\mathbf{x}])^T \right]$$

$$V_{\mathbf{x}}[\mathbf{x}]_{i,i} = E_{\mathbf{x}} \left[ (\mathbf{x}^{(i)} - E_{\mathbf{x}}[\mathbf{x}^{(i)}])^2 \right]$$

$$V_{\mathbf{x}}[\mathbf{x}]_{i,j} = E_{\mathbf{x}} \left[ (\mathbf{x}^{(i)} - E_{\mathbf{x}}[\mathbf{x}^{(i)}])(\mathbf{x}^{(j)} - E_{\mathbf{x}}[\mathbf{x}^{(j)}]) \right]$$

# 相関係数

- 相関係数

- 分散共分散行列を正規化した量

$$\rho = \frac{E_{\mathbf{x}} \left[ \left( \mathbf{x}^{(i)} - E_{\mathbf{x}} \left[ \mathbf{x}^{(i)} \right] \right) \left( \mathbf{x}^{(j)} - E_{\mathbf{x}} \left[ \mathbf{x}^{(j)} \right] \right) \right]}{\sqrt{V_{\mathbf{x}} \left[ \mathbf{x}^{(i)} \right]} \sqrt{V_{\mathbf{x}} \left[ \mathbf{x}^{(j)} \right]}}$$

- 相関係数が正の値をとる場合

- 2つの確率変数の増減関係は同傾向

- 相関係数が負の値をとる場合

- 2つの確率変数の増減関係は逆傾向

# 独立と無相関

- 2つの確率変数が独立
  - 確率変数  $x$  と  $y$  の同時確率が以下の関係

$$p(x, y) = p(x)p(y)$$

- 独立ならば無相関
- 無相関でも独立とは限らない

# ガウス分布(正規分布)

- 1次元の確率密度関数

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

- $d$ 次元の確率密度関数

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2} \det(\boldsymbol{\Sigma})^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

# Octaveの使い方

- Octaveとは
  - 数値計算が手軽にできるフリーのツール
  - MATLABと互換性あり
- 起動方法
  - Linux環境の場合
    - コマンドラインから 'octave' を入力
  - Windows環境の場合
    - Octaveのショートカット、もしくはスタートメニューから Octave を選択

# 基本的な計算

```
Octave-3.2.4
GNU Octave, version 3.2.4
Copyright (C) 2009 John W. Eaton and others.
This is free software; see the source code for copying conditions.
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE.  For details, type `warranty'.

Octave was configured for "i686-pc-mingw32".

Additional information about Octave is available at http://www.octave.org.

Please contribute if you find this software useful.
For more information, visit http://www.octave.org/help-wanted.html

Report bugs to <bug@octave.org> (but first, please read
http://www.octave.org/bugs.html to learn how to write a helpful report).

For information about changes from previous versions, type `news'.

octave-3.2.4.exe:1> (1+2)*(3-4)*5^2
ans = -75
octave-3.2.4.exe:2> exp(10)*exp(5)*sin(0.5*pi)
ans = 3.2690e+006
octave-3.2.4.exe:3> exp(10)*log(5)*sin(0.5*pi)
ans = 3.5450e+004
octave-3.2.4.exe:4>
```

# ベクトル、行列の演算(1)

```
octave-3.2.4.exe:9> A=[1 2;3 4]
```

```
A =
```

```
1 2
```

```
3 4
```

```
octave-3.2.4.exe:10> b=[3;4]
```

```
b =
```

```
3
```

```
4
```

```
octave-3.2.4.exe:11> C=A*b
```

```
C =
```

```
11
```

```
25
```

```
octave-3.2.4.exe:12> C'
```

```
ans =
```

```
11 25
```

```
octave-3.2.4.exe:13> A(2,1)
```

```
ans = 3
```

```
octave-3.2.4.exe:14> A(2,1)=5
```

```
A =
```

```
1 2
```

```
5 4
```

```
octave-3.2.4.exe:15> A(:,1)
```

```
ans =
```

```
1
```

```
5
```

```
octave-3.2.4.exe:16> D=3*A
```

```
D =
```

```
3 6
```

```
15 12
```

# ベクトル、行列の演算(2)

```
octave-3.2.4.exe:17> A.*D
```

```
ans =
```

```
3 12
```

```
75 48
```

```
octave-3.2.4.exe:18> A./D
```

```
ans =
```

```
0.33333 0.33333
```

```
0.33333 0.33333
```

```
octave-3.2.4.exe:19> E = 10:5:30
```

```
E =
```

```
10 15 20 25 30
```

```
octave-3.2.4.exe:20> F = 1:5
```

```
F =
```

```
1 2 3 4 5
```

```
octave-3.2.4.exe:21> cos(2/3*pi)
```

```
ans = -0.50000
```

```
octave-3.2.4.exe:22> sin([1 2 3 4])
```

```
ans =
```

```
0.84147 0.90930 0.14112 -0.75680
```

```
octave-3.2.4.exe:23> max([1 3 5 4 2])
```

```
ans = 5
```

```
octave-3.2.4.exe:24> A=[1 2;3 4]; inv(A)
```

```
ans =
```

```
-2.00000 1.00000
```

```
1.50000 -0.50000
```

# ベクトル、行列の演算(3)

```
octave-3.2.4.exe:25> [eigvec eigval] =  
    eig(A)  
eigvec =  
  -0.82456 -0.41597  
   0.56577 -0.90938  
eigval =  
Diagonal Matrix  
  -0.37228    0  
    0  5.37228  
octave-3.2.4.exe:26> mean([1 2 3 4])  
ans = 2.5000  
octave-3.2.4.exe:27> std([1 2 3 4])  
ans = 1.2910  
octave-3.2.4.exe:28> var([1 2 3 4])  
ans = 1.6667
```

```
octave-3.2.4.exe:29> randn(2,4)  
ans =  
  
  1.13107 -0.85983  0.89008  1.82015  
  0.78744  0.27756 -0.79847  0.32866  
  
octave-3.2.4.exe:30> [sorted index] =  
    sort([7 3 6 1 2])  
sorted =  
  
  1  2  3  6  7  
  
index =  
  
  4  5  2  3  1
```

# グラフ描画

- plot関数
  - 折れ線グラフ
- hist関数
  - ヒストグラム
- surf関数 (mesh関数)
  - 塗りつぶしあり(なし) 3次元グラフ
- contour関数
  - 2次元の等高線

# プログラミング(1)

- スクリプトの基本
  - テキストファイルに複数のコマンド記述
- 例 (myscript.m)
  - A=[1 2; 3 4]
  - B=[5; 6]
  - C=A\*b;
- コマンドライン上で “myscript” として実行

# プログラミング(2)

- 繰り返し(for文)

```
n = 1000000;
```

```
a = 0;
```

```
for i = 1:n
```

```
    a=a+i;
```

```
end
```

- for文は処理が遅いので使わない

```
a = sum(1:1000000);
```

# プログラミング(3)

- 条件分岐 ( if 文)

```
x = -10:0.1:10;
for i=1:length(x)
    if x(i) > 0
        y(i) = x(i);
    else
        y(i) = -x(i);
    end
end
Plot(x,y);
```

- 別の記述

```
x = -10:0.1:10;
y=zeros(size(x));
y(x>0) = x(x>0);
y(x=<0) = -x(x=<0);
plot(x,y);
```

# プログラミング(4)

- 関数

- 関数の定義

- スクリプトの先頭行で function と宣言

- 関数ファイルの名前

- “関数名.m” という名前で保存

- 実行方法

- 他のスクリプトから関数を呼び出すことで実行可能