2017/5/4 Pythonで体験するベイズ推論輪読会LT

ベイズ統計の入口

@currypurin

自己紹介

- @currypurin
- 輪読会をやっています
- ベイズ統計はじめました
- 今の状況は、次ページ

これらなわかるベイズ統計学P232より

- MCMCの道は厳しそうですけど、PyMCか簡単にモデリングしてくれるらしい
- 今日は、左の道の「自然な共役分布」の話を少しだけ

目次

ベイズの定理の復習

コイン投げの問題をやってみる

まとめ

目次

分布に関するベイズの定理

$$\overbrace{\pi(\theta|D)}^{\text{事後分布}} = \frac{\overbrace{f(D|\theta)}^{\text{尤度}}\,\overbrace{\pi(\theta)}^{\text{事前分布}}}{f(D)}$$

ベイズ統計では、θ を確率変数として、

その分布を考える

事後分布は尤度と事前分布の積に比例

$$\pi(\theta|D) = kf(D|\theta)\pi(\theta)$$

（kは定数）

ベイズ統計の基本公式

$$\pi(\theta|D) \propto f(D|\theta)\pi(\theta)$$

目次

ベイズの定理の復習

コイン投げの問題をやってみる

まとめ

コイン投げ

- 表の出る確率がθである１枚のコインがある。このコインを投げたとき１回目→表、２回目→表、３回目→表、４回目→裏と出たとする。このとき「表のでる確率θ」の確率分布を求めよ。

・表が出やすそうなコイン

・分布に関するベイズの定理を用いて計算してみる

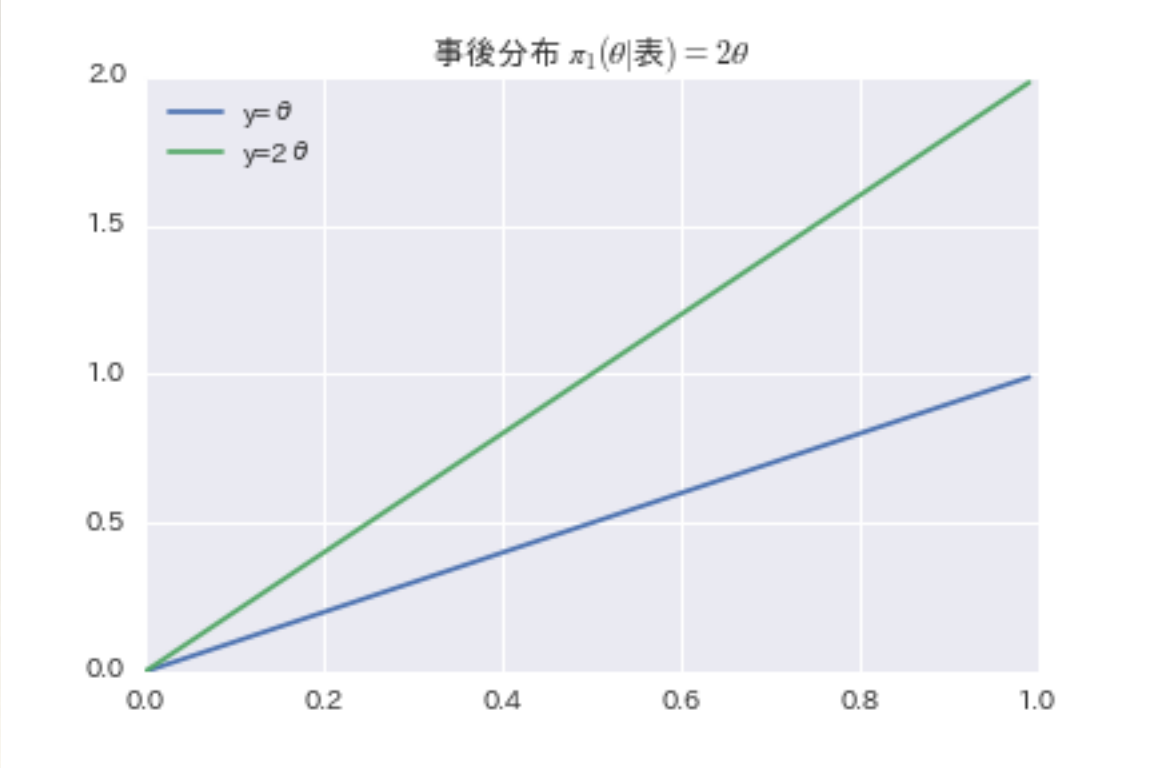
コイン投げ１回目(表)

$$\pi_1(\theta|\text{表}) = kf(\text{表}|\theta)\pi_0(\theta)$$

- 尤度　　　：f(表|θ)= θ
- 事前分布：$\pi_0(\theta)$=1
- $\pi_1(\theta|$表)=k×θ
- k　　　　：k=2

※kは積分したら1という確率分布の条件から求める。

$$\pi_1(\theta|\text{表}) = 2\theta$$

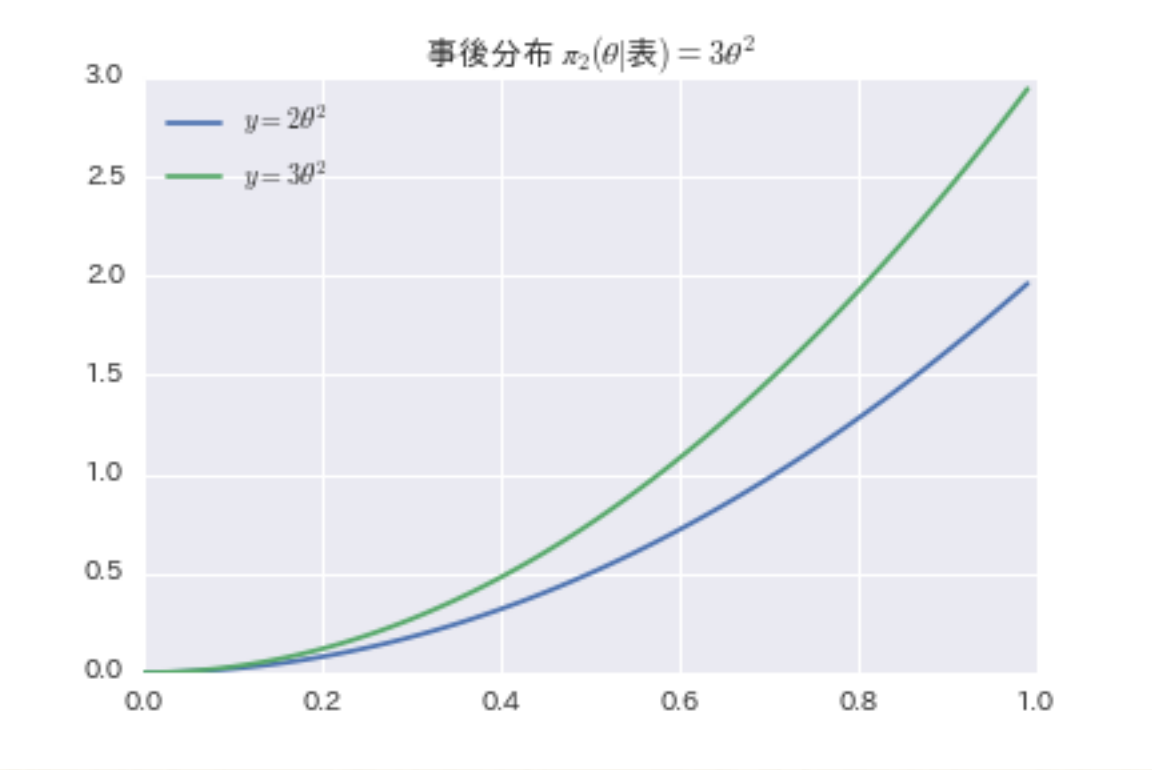


コイン投げ２回目(表→表)

$$\pi_2(\theta|\text{表}) = kf(\text{表}|\theta)\pi_1(\theta)$$

- 尤度　　　：f(表|θ)= θ
- 事前分布：$\pi_1(\theta)=2\theta$
- $\pi_2(\theta|\text{表})=k\times 2\theta$^2
- k　　　　：k=3/2

$$\pi_2(\theta|\text{表}) = 3\theta^2$$

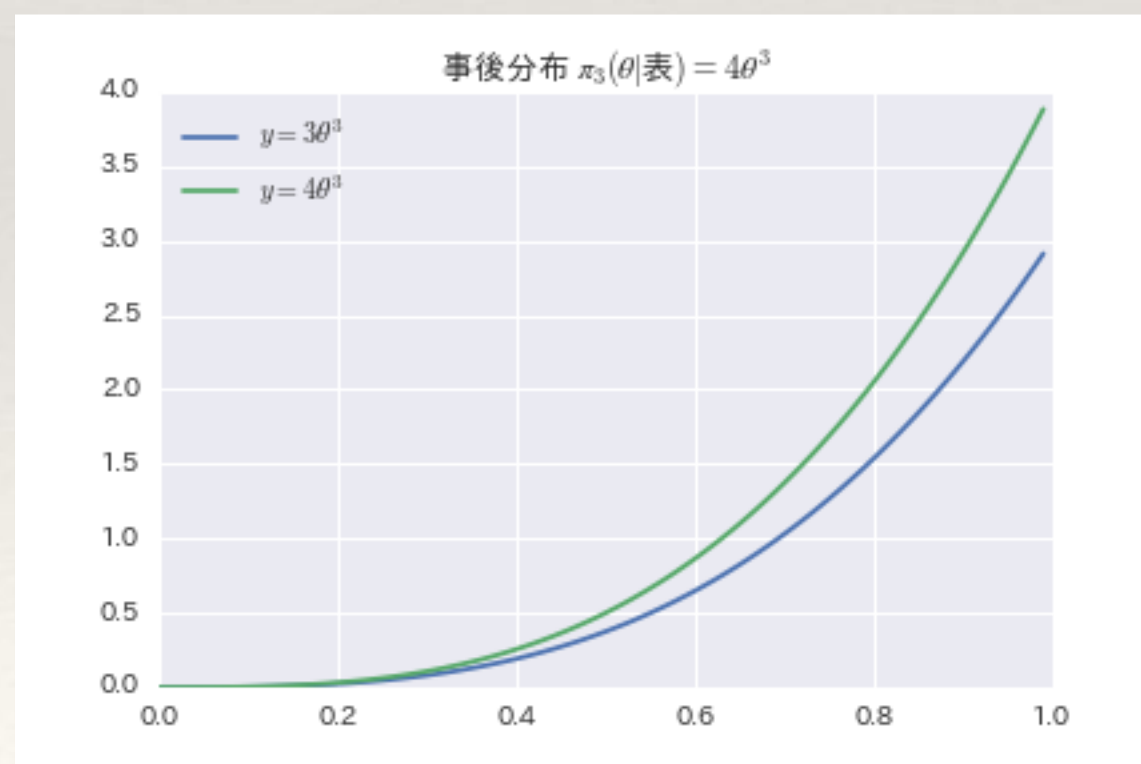


コイン投げ３回目(表表→表)

$$\pi_3(\theta|\text{表}) = kf(\text{表}|\theta)\pi_2(\theta)$$

- 尤度　　　：f(表|θ)= θ
- 事前分布：π₂(θ)=3θ^2
- π₃(θ|表)=k×3θ^3
- k　　　　：k=4/3

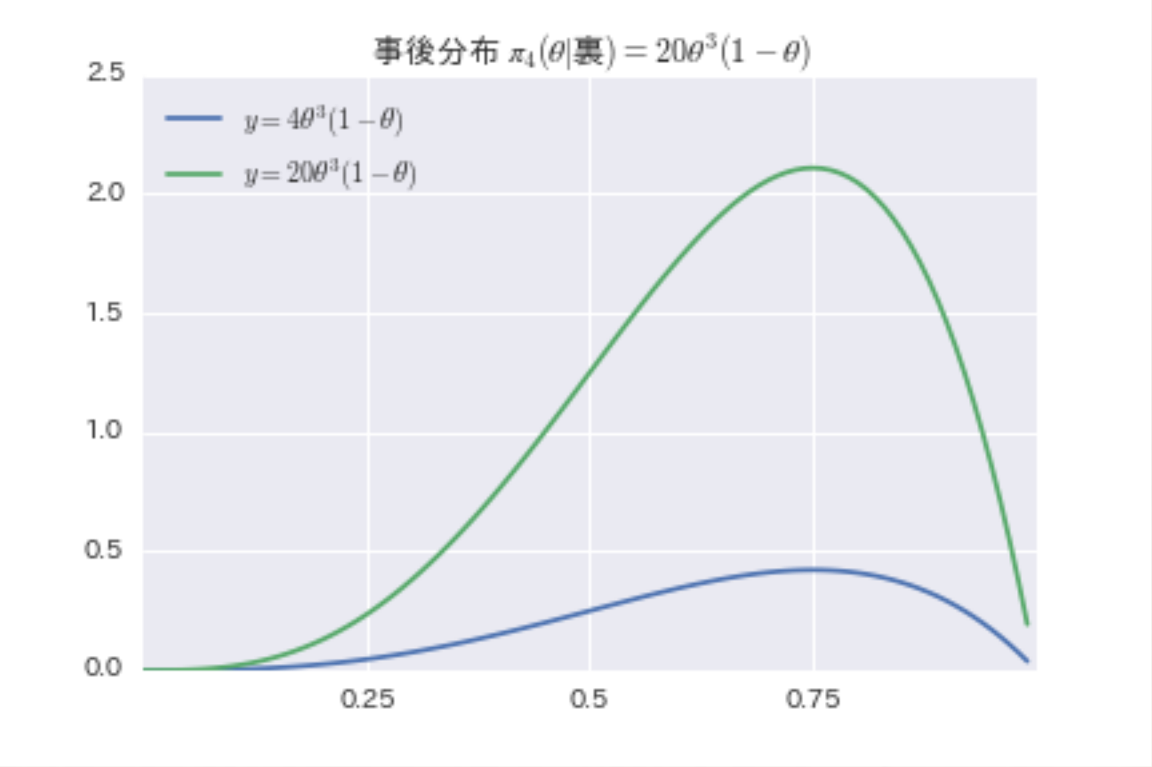
$$\pi_3(\theta|\text{表}) = 4\theta^3$$



コイン投げ４回目(表表表→裏)

$$\pi_4(\theta|\text{裏}) = kf(\text{裏}|\theta)\pi_3(\theta)$$

- 尤度　　　：f(裏|θ)= 1-θ
- 事前分布：π₃(θ)=4θ^3
- π₄(θ|表)=k×4θ^3×(1-θ)
- k　　　　：k=5



$$\pi_4(\theta|\text{裏}) = 20\theta^3(1-\theta)$$

コイン投げ(表表表裏)　　一気に求めることも可能

$$\pi(\theta|\text{表表表裏}) = kf(\text{表表表裏}|\theta)\pi_0(\theta)$$

- 尤度　　　：f(表表表裏| θ)= θ ^3 ×(1- θ)
- 事前分布：$\pi_0(\theta)=1$
- π (θ |表表表裏)=k× θ ^3×(1- θ)
- k　　　　　：k=20

$$\pi(\theta|\text{表表表裏}) = 20\theta^3(1-\theta)$$

一般に、kはどのように求めるか？

kの求め方

- １　積分を用いる
- ２　自然な共役分布を用いる

$$\int_0^1 \theta^3(1-\theta)d\theta$$ を計算

Pythonコード

```
θ = sym.symbols('θ')
f = (θ**3)*(1-θ)
sym.integrate(f,(θ,0,1)) #fを0,1の範囲で積分
```

```
1/20
```

よって、積分すると１という条件からk=20

（もちろん手計算してもOK）

（参考）Pythonでの積分

- 代数計算ライブラリsympy を使用して計算が可能

Pythonコード

```
In [47]: import sympy as sym
         z = sym.symbols("z") # zをsympyの変数にする
         f1 = z #  関数z1を定義
         f2 = 4/(1+z**2) #  関数f2を定義

In [45]: sym.integrate(f1,(z,0,1)) #  f1を0,1の範囲で積分
Out[45]: 1/2

In [48]: sym.integrate(f2,(z,0,1)) #  f4を0,1の範囲で積分
Out[48]: pi
```

自然な共役分布を用いた計算１

ベータ分布

$$Be(p,q) \equiv \frac{1}{B(p,q)} x^{p-1}(1-x)^{q-1}$$

$$(0 \leq x \leq 1)$$

ベータ関数

$$B(p,q) = \int_0^1 x^{p-1}(1-x)^{q-1}dx$$

p,qが整数の時

$$B(p,q) = \frac{(p-1)!(q-1)!}{(p+q-1)!}$$

自然な共役分布を用いた計算2

$$事後分布 : k\theta^3(1-\theta)$$

$$Be(p,q) = \frac{1}{B(p,q)}\theta^{p-1}(1-\theta)^{q-1}$$

- ベータ分布において、p=4,q=2とすると事後分布と比例する
- B(4,2)=20
- ∴k=20

目次

尤度と事前分布・事後分布の相性

- 二項分布（例：コイン投げ）に従う確率の分布は、ベータ分布を用いると簡単に扱うことができる。
- このように、事前分布に特定の分布を指定すると、事後分布がその特定のパターンと同じパターンに収まる関係を**自然な共役分布の関係**という。
- 複雑な確率分布などは、これが使えない。

（参考）有名な自然な共役分布の関係

尤度関数	事前分布、事後分布
二項分布	ベータ分布
正規分布	正規分布
正規分布	逆ガンマ分布
ポアソン分布	ガンマ分布

➡ 続きは、5/8の輪読会で。