統計的機械学習
第八回　レポート ID: 01

37-196360　森田涼介

2019 年 6 月 10 日

ある薬の効果を調べるために 5 人の被験者に対して実験を行った。5 人のうち 4 人の効果が認められた。このとき，この薬の効果をベイズ推定によって分析する。

理論

効果のある確率を π とおき，観測データは独立にベルヌーイ分布に従うとする。また，事前分布は $a, b\ (>0)$ をパラメータとするベータ分布であると仮定する。これらを式で表すと，

$$p(\text{data}|\pi) = \text{Bernoulli}(\text{data}|\pi) \tag{1}$$

$$p(\pi) = \text{Beta}(\pi|a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\pi^{a-1}(1-\pi)^{b-1} \tag{2}$$

ここで，観測データのうち陽性のものの数を n_{pos}，陰性のものの数を n_{neg} とする。$p(\text{data})$ は定数であることに注意すると，π の事後分布の確率密度関数は，

$$\begin{aligned}
p(\pi|\text{data}) &= \frac{p(\text{data}|\pi)p(\pi)}{p(\text{data})} && (3)\\
&\propto p(\text{data}|\pi)p(\pi) && (4)\\
&= \pi^{n_{\text{pos}}} \cdot (1-\pi)^{n_{\text{neg}}} \cdot \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\pi^{a-1}(1-\pi)^{b-1} && (5)\\
&\propto \pi^{a+n_{\text{pos}}-1}(1-\pi)^{b+n_{\text{neg}}-1} && (6)
\end{aligned}$$

これを正規化すると，結局，事後分布は，

$$p(\pi|\text{data}) = \text{Beta}(\pi|a+n_{\text{pos}},\ b+n_{\text{neg}}) \tag{7}$$

また，効果のある確率 π が threshold 以上である確率は次式で計算できる。

$$p(\pi \geq \text{threshold}|\text{data}) = \int_{\text{threshold}}^{1} p(\pi|\text{data})\mathrm{d}\pi \tag{8}$$

結果

事前分布 Beta(1, 1), Beta(0.1, 0.1), Beta(5, 5) について，$\pi \geq 0.5$，及び $\pi \geq 0.8$ となる確率を，数値積分により求める。結果を表 1 にまとめる。また，ベータ分布の形状を図 1 に示す。

表 1: 結果

threshold	Beta(1, 1)	Beta(0.1, 0.1)	Beta(5, 5)
0.5	0.89	0.93	0.79
0.8	0.34	0.56	0.044

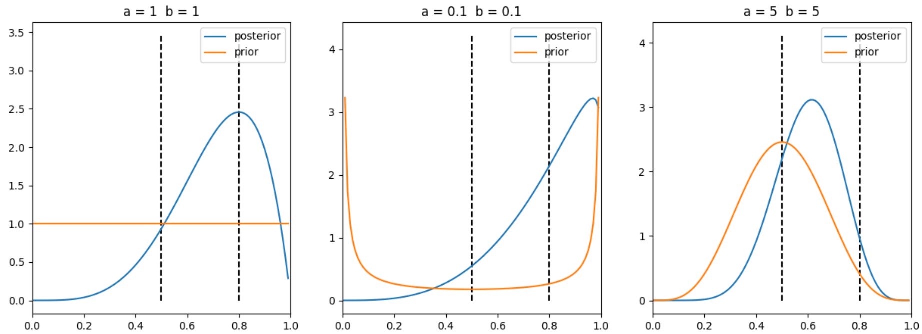

図 1: 各 a, b に対するベータ分布の形状

プログラム

Listings 1: assignment1.py

```
# -*- coding: utf-8 -*-

import math
import numpy as np
import matplotlib.pyplot as plt


def beta_distribution(pi, a, b):
    value = (
        (math.gamma(a + b) / (math.gamma(a) * math.gamma(b)))
        * pi**(a-1) * (1 - pi)**(b-1)
        )
    return value


def make_beta(a, b):
    def _beta(pi):
        return beta_distribution(pi, a, b)
    return _beta


def integrate(distribution, lower=0.0, upper=1.0, dx=0.01,):
```

```
    x = np.arange(lower, upper+dx, dx)
    dist = distribution(x)
    p = (dist*dx).sum()
    return p


def main():
    # settings
    n_pos, n_neg = 4, 1  # data
    ab_cands = [(1, 1), (0.1, 0.1), (5, 5)]  # prior dist
    thresholds = [0.5, 0.8]
    dx = 0.0001
    fig_path = '../figures/assignment1_result.png'

    # calc
    x_axis = np.arange(0, 1.0, 0.01)
    n_row = 1
    n_col = len(ab_cands)
    fig = plt.figure(figsize=(5*n_col, 5*n_row))
    for i, (a, b) in enumerate(ab_cands):
        distribution = make_beta(a+n_pos, b+n_neg)
        for threshold in thresholds:
            p = integrate(distribution, lower=threshold, dx=dx)
            print(f'a = {a}  b = {b}  p(pi > {threshold}) = {p}')

        # plot
        dist = distribution(x_axis)
        label = f'a = {a}  b = {b}'
        ax = fig.add_subplot(n_row, n_col, i+1)
        ax.set_title(label)
        ax.set_xlim(0, 1)
        for threshold in thresholds:
            ax.vlines(threshold, 0, dist.max()+1, linestyle='dashed')
        ax.plot(x_axis, dist, label='posterior')
        ax.plot(x_axis, beta_distribution(x_axis, a, b), label='prior')
        ax.legend()
    if fig_path:
        plt.savefig(str(fig_path))
    plt.show()



if __name__ == '__main__':
    main()
```