Bayesian Data Analysis - Assignment 2

September 24, 2017

Inference for binomial proportion

The language used is Python. The source code is attached in the appendix.

likelihood: $p(y \mid \pi) \propto \pi^a(1-\pi)^b$

$$p(y \mid \pi) = Bin(y \mid n, \pi) = \binom{n}{y} \pi^y (1-\pi)^{n-y}, \quad y = 0, 1, ..., n \tag{1}$$

prior for π: $\pi \sim Beta(\alpha, \beta) \quad (\alpha = 2, \beta = 10)$
prior density: $p(\pi) \propto \pi^{\alpha-1}(1-\pi)^{\beta-1}$

$$\begin{aligned} p(\pi) &= \frac{1}{B(\alpha,\beta)}\pi^{\alpha-1}(1-\pi)^{\beta-1} \\ B(\alpha,\beta) &= \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)} \end{aligned} \tag{2}$$

posterior:

$$\begin{aligned} p(\pi \mid y) &\propto \pi^y(1-\pi)^{n-y}\pi^{\alpha-1}(1-\pi)^{\beta-1} \\ &= \pi^{y+\alpha-1}(1-\pi)^{n-y+\beta-1} \\ &= Beta(\pi \mid \alpha+y, \beta+n-y) \end{aligned} \tag{3}$$

a)

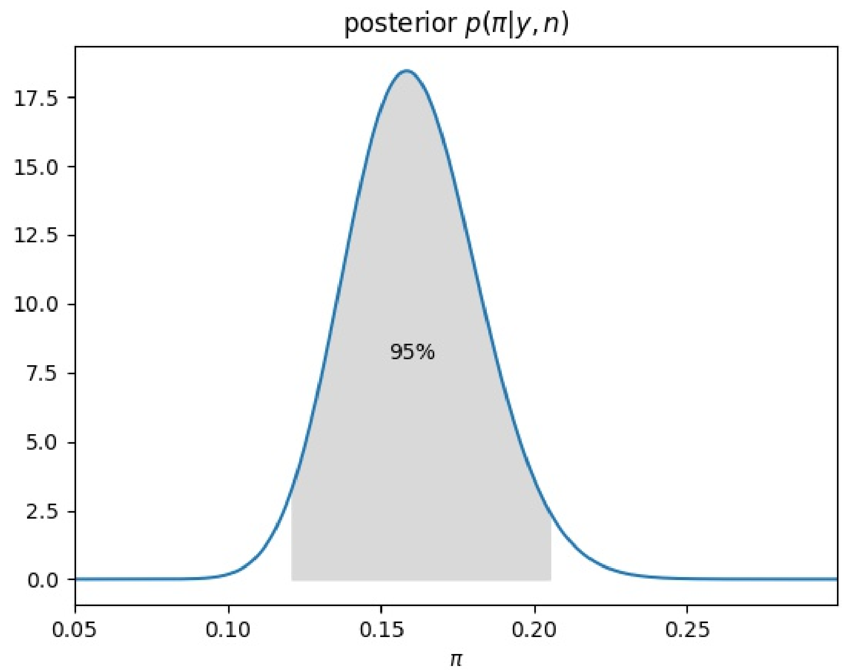

Figure 1: posterior density function

y = 44, n = 274, posterior: $\alpha = 46, \quad \beta = 240$

$$mean = 0.160839$$
$$median = 0.160048$$
$$variance = 0.000470$$

We can get $(\frac{y}{n} \approx 0.1606) < E[p(\pi \mid y, n)] < (E[\pi] = \frac{2}{2+10} \approx 0.1667)$

The central 95% interval: [0.120656, 0.205512]

b)

The probability that π is smaller than 0.2:

$$p(\pi < (\pi_0 = 0.2)) = 0.958614 \approx 95.86\%$$

c)

Assumptions:
1. trials are independent and trial probabilities do not vary from trial to trial. (independent and identically distributed)
2. trials are exchangeable.

d)

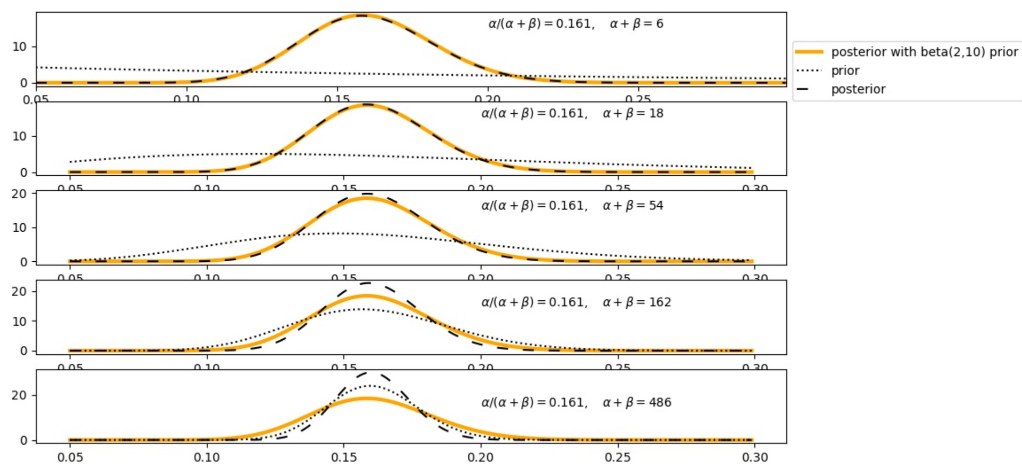

Figure 2: comparisons of different priors

Parameters of the prior distrubution		Summaries of the posterior distribution			
$\frac{\alpha}{\alpha+\beta}$	$\alpha+\beta$	mean	median	variance	central 95%interval
0.161	6	0.160593	0.159784	0.000480	[0.120037,0.205738]
0.161	18	0.160610	0.159834	0.000460	[0.120841,0.204778]
0.161	54	0.160652	0.159962	0.000410	[0.122993,0.202229]
0.161	162	0.160739	0.160220	0.000309	[0.127816,0.196608]
0.161	486	0.160850	0.160552	0.000177	[0.135610,0.187780]

Table 1: Summaries of the posterior distribution

Posterior inferences are not particularly sensitive to the prior distribution when $\alpha + \beta$ (prior observations) is relatively small. With $\alpha + \beta$ increasing, posterior inferences become relatively more sensitive to the prior, and the rows of the table use prior distributions that are increasingly concentrated around 0.161.

Appendix

Source code

```
import numpy as np
from scipy.stats import beta
import matplotlib.pyplot as plt


# Read data from the file
file = open("algae.txt", 'r')
y = 0  # y represents the amount of sites where algae present
n = 0  # N represents total observations
for line in file:
        for char in line:
                if char == '1':
                        y += 1
                        n += 1
                if char == '0':
                        n += 1

print("y = {}, n={}".format(y, n))

# Beta(2,10) prior for pi
a = 2
b = 10

# Posterior distribution Beta(a+y, b+n-y)
dist = beta(a+y, b+n-y)
x = np.arange(0.05, 0.3, 0.001)
```

```python
pd = dist.pdf(x)
plt.figure()
plt.plot(x, pd)
plt.autoscale(axis='x', tight=True)
plt.xlabel(r'$\pi$')
plt.title("posterior " r'$p(\pi|y,n)$')

# a)
mean = dist.mean()
median = dist.median()
variance = dist.var()
# mean = (a+y)/(a+b+n)
# variance = ((mean*(1-mean))/(a+b+n+1))
print("mean = {:.6f}\n"
      "median = {:.6f}\n"
      "variance = {:.6f}".format(mean, median, variance))

# find the points in y that are between 2.5% and 97.5% quantile
x_95_idx = (x > dist.ppf(0.025)) & (x < dist.ppf(0.975))
print("The central 95\% interval: [{:.6f},{:.6f}]"
      .format(dist.ppf(0.025), dist.ppf(0.975)))
plt.fill_between(x[x_95_idx], pd[x_95_idx], color='0.85')
plt.text(dist.median(), 8, "95%", horizontalalignment='center')

# b)
smaller_than_20percent = dist.cdf(0.2)
print("The probability that pi"
      " is smaller than 0.2: {:.6f}\n"
      .format(smaller_than_20percent))

# d)
# compare 5 cases
# prior distribution: Beta(0.161*n, (1-0.161)*n)
# for n = 6, 18, 54, 162, 486
prior_a = np.array([0.161*(6*3**i) for i in range(5)])
prior_b = np.array([(1-0.161)*(6*3**i) for i in range(5)])

# corresponding posteriors with y,n
posterior_a = prior_a + y
posterior_b = prior_b + n - y
```

```python
# calculate prior and posterior densities
prior_pd = beta.pdf(x, prior_a[:, np.newaxis], prior_b[:, np.newaxis])
posterior_pd = beta.pdf(x, posterior_a[:, np.newaxis],
posterior_b[:, np.newaxis])

# plot 5 subplots
fig, axes = plt.subplots(nrows=5, ncols=1, figsize=(8, 15))
for i, ax in enumerate(axes):
        post1 = ax.plot(x, pd, color='orange',
                                        linewidth=3, label="posterior with
        prior = ax.plot(x, prior_pd[i], 'k:', label="prior")
        post2 = ax.plot(x, posterior_pd[i], color='k',
                                        dashes=(6, 8), label="posterior")
        ax.annotate(r'$\alpha/(\alpha+\beta)=0.161, \quad '
                              r'\alpha+\beta={}$'.format(6*3**i), xy=(0.
        box = ax.get_position()
        ax.set_position([box.x0, box.y0, box.width * 0.8, box.height])

axes[0].autoscale(axis='x', tight=True)
plt.legend(loc='center left', bbox_to_anchor=(1, 5))


for i in range(5):
        print("prior: a={},b={}".format(prior_a[i], prior_b[i]))
        print("posterior: a={},b={}".
                    format(posterior_a[i], posterior_b[i]))
        dist = beta(posterior_a[i], posterior_b[i])
        mean = dist.mean()
        median = dist.median()
        variance =dist.var()
        print("mean ={:.6f}, median={:.6f}, "
                    "variance={:.6f}".format(mean, median, variance))
        x_95_idx = (x > dist.ppf(0.025)) & (x < dist.ppf(0.975))
        print("The central 95\% interval: "
                "[{:.6f},{:.6f}]\n".
                format(dist.ppf(0.025), dist.ppf(0.975)))

plt.show()
```