

Bayesian Data Analysis - Assignment 4

October 8, 2017

Generalized linear model: Bioassay + grid sampling

prior:

$$\alpha \sim N(0, 2^2) \quad \beta \sim N(10, 10^2) \quad \text{corr}(\alpha, \beta) = 0.5$$

$$(\alpha, \beta) \sim N(\mu, \Sigma)$$

$$\mu = \begin{pmatrix} \mu_\alpha \\ \mu_\beta \end{pmatrix} \quad \Sigma = \begin{pmatrix} \sigma_\alpha^2 & \rho\sigma_\alpha\sigma_\beta \\ \rho\sigma_\alpha\sigma_\beta & \sigma_\beta^2 \end{pmatrix}$$

Binomial model:

$$y_i \mid \theta_i \sim \text{Bin}(\theta_i, n_i)$$

Link function:

$$\text{logit}(\theta_i) = \alpha + \beta x_i$$

Likelihood:

$$p(y_i \mid \alpha, \beta, n_i, x_i) \propto \theta_i^{y_i} [1 - \theta_i]^{n_i - y_i}$$
$$p(y_i \mid \alpha, \beta, n_i, x_i) \propto [\text{logit}^{-1}(\alpha + \beta x_i)]^{y_i} [1 - \text{logit}^{-1}(\alpha + \beta x_i)]^{n_i - y_i}$$

posterior:

$$p(\alpha, \beta \mid y, n, x) \propto p(\alpha, \beta) \prod_{i=1}^n p(y_i \mid \alpha, \beta, n_i, x_i)$$

a)

```
import numpy as np
from scipy.stats import multivariate_normal
from scipy.special import expit
import matplotlib.pyplot as plt
import plot_tools

# seed a random state
rng = np.random.RandomState(0)

# data
x = np.array([-0.86, -0.30, -0.05, 0.73])
n = np.array([5, 5, 5, 5])
y = np.array([0, 1, 3, 5])

# multivariate normal distribution prior
mean1 = 0
mean2 = 10
var1 = 2**2
var2 = 10**2
covab = 0.5*2*10
mean = [mean1, mean2]
cov = [[var1, covab], [covab, var2]]
prior_dist = multivariate_normal(mean, cov)

# compute the posterior density in grid
A = np.linspace(-4, 8, 100)
B = np.linspace(-8, 35, 100)
grid_x, grid_y = np.meshgrid(A, B)
pos = np.empty(grid_x.shape + (2,))
pos[:, :, 0] = grid_x
pos[:, :, 1] = grid_y
prior = prior_dist.pdf(pos)
ilogit_abx = 1 / (np.exp(-(A[:, None] +
B[:, None, None] * x)) + 1)
likelihood = np.prod(ilogit_abx**y *
(1 - ilogit_abx)**(n - y), axis=2)
p = prior*likelihood

# sample from the grid
nsamp = 1000
samp_indices = np.unravel_index(
rng.choice(p.size, size=nsamp,
p=p.ravel()/np.sum(p)), p.shape)
samp_A = A[samp_indices[1]]
samp_B = B[samp_indices[0]]
# add random jitter
```

```

samp_A += (rng.rand(nsamp) - 0.5) * (A[1]-A[0])
samp_B += (rng.rand(nsamp) - 0.5) * (B[1]-B[0])
# samples of LD50 conditional beta > 0
bpi = samp_B > 0
samp_ld50 = -samp_A[bpi]/samp_B[bpi]

# posterior contour plot
plt.contourf(grid_x, grid_y, p)
plt.xlabel(r'$\alpha$')
plt.ylabel(r'$\beta$')
plt.title("posterior contour plot")

# plot the samples
plt.figure()
plt.scatter(samp_A, samp_B, 10, linewidth=0)
plt.xlim([-4,8])
plt.ylim([-8,35])
plt.xlabel(r'$\alpha$')
plt.ylabel(r'$\beta$')
plt.title("samples")

# plot the histogram of LD50
plt.figure()
plt.hist(samp_ld50, np.arange(-0.5, 0.51, 0.02))
plt.xlim([-0.5, 0.5])
plt.xlabel(r'LD50 =  $-\alpha/\beta$ ')
plt.ylabel("frequency")
plt.title("histogram of LD50")
plt.yticks(())
plt.show()

```

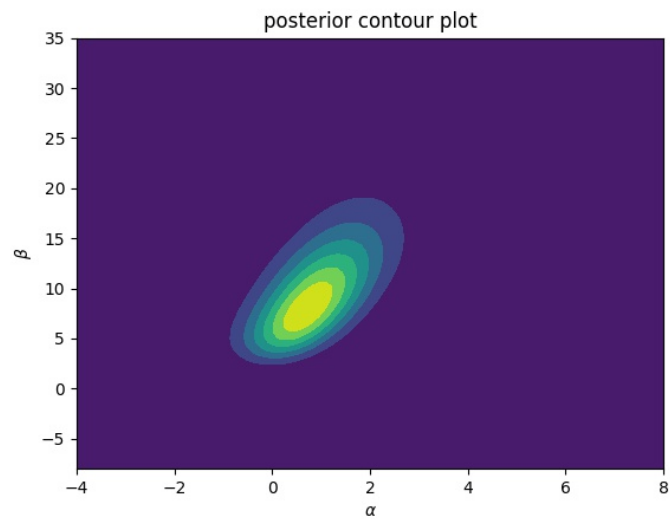


Figure 1: posterior contour plot

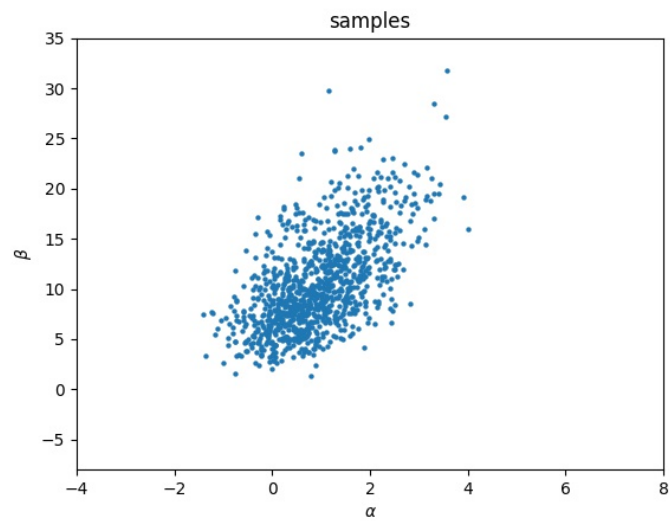


Figure 2: scatterplot of 1000 samples

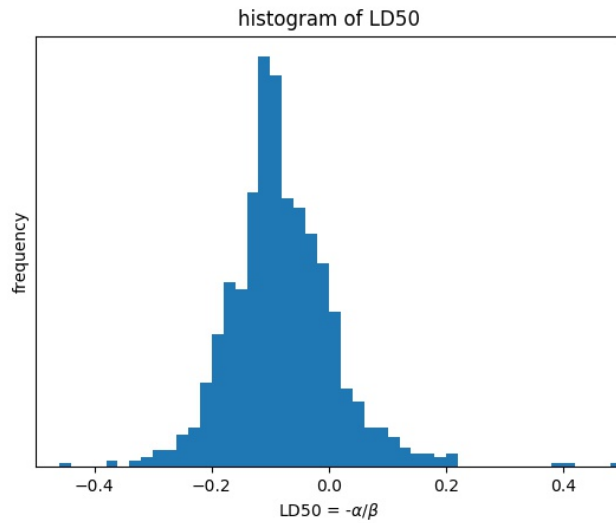


Figure 3: histogram of LD50

b)

```
# create figure
fig, axes = plt.subplots(1,4,figsize=(30,4))

# prior
ax = axes[0]
ax.contourf(grid_x, grid_y, prior)
ax.set_xlabel(r'$\alpha$')
ax.set_ylabel(r'$\beta$')
ax.set_title("prior")

# likelihood
ax = axes[1]
ax.contourf(grid_x, grid_y, likelihood)
ax.set_xlabel(r'$\alpha$')
ax.set_title('likelihood')

# posterior contour plot
ax = axes[2]
ax.contourf(grid_x, grid_y, p)
ax.set_xlabel(r'$\alpha$')
ax.set_title("posterior")

# plot the samples
ax = axes[3]
```

```

ax.scatter(samp_A, samp_B, 10, linewidth=0)
ax.set_xlim([-4,8])
ax.set_ylim([-8,35])
ax.set_xlabel(r'$\alpha$')
ax.set_title('samples')
plt.show()

```

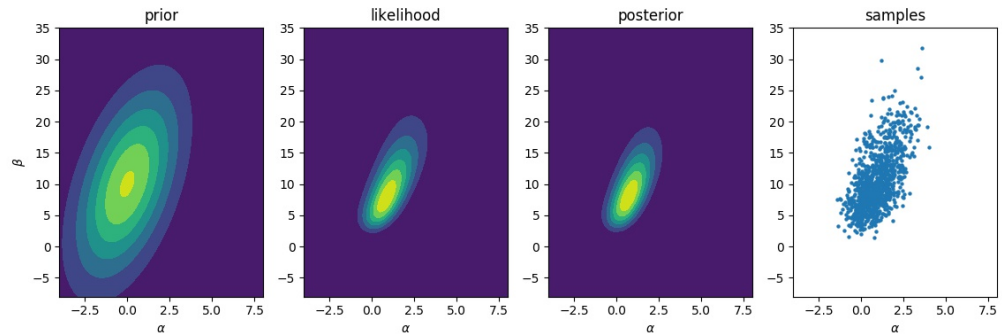


Figure 4: plots of prior, likelihood, posterior and samples

From the Figure 4, we can see that the contour plot and the scatter plot look like a compromise between the prior distribution and the likelihood.

c)

```

prop = len(samp_B[samp_B>0])/len(samp_B)
print(len(samp_B[samp_B>0]),len(samp_B),prop)

```

Output: 1000 1.0

The probability that $\beta > 0$ among samples is 1.

Considering the result comes from samples, more exactly, we can get

$$p(\beta > 0 \mid x, n, y) > 0.999$$

The probability that the drug is harmful is nearly 1.