ISYE 6420 Bayesian Statistics

Homework 4

October 23, 2021

1.

a)

Based on the observed prior for the interval $[-m,\ m]$ we can derive the following:

$$\pi(\Theta) = \frac{cos^2\left(\frac{\pi(\Theta)}{2m}\right)}{m}$$

Therefore our Bickel-Levit prior where $m = 2$:

$$\pi(\Theta) = \frac{cos^2\left(\frac{\pi(\Theta)}{4}\right)}{2}$$

Our likelihood for the sample is observed with a normal distribution of the following:

$$f(y|\theta) \propto \sqrt{\tau}\exp\left\{-\frac{\tau}{2}(y-\theta)^2\right\},$$

Therefore our posterior is proportional to the following:

$$\propto \frac{cos^2\left(\frac{\pi(\Theta)}{4}\right)}{2} * \prod_{i=1}^{n} \frac{1}{2}\, exp\left(-\frac{1}{8}(y-\Theta)^2\right)$$

From this we can derive the proposal value for the metropolis algorithm by looking at our uniform distribution between the range of $[-2\ ,\ 2]$:

$$q\left(\Theta|\Theta'\right) = \frac{1}{b-a}$$
$$= \frac{1}{4}$$

Now we can derive γ which we know from the lectures is:

$$\gamma = \frac{\pi(\Theta') * q\left(\Theta|\Theta'\right)}{\pi(\Theta) * q\left(\Theta'|\Theta\right)}$$

$$\gamma = \frac{\frac{cos^2\left(\frac{\pi(\Theta)'}{4}\right)}{2} * \prod_{i=1}^{n} \frac{1}{2} exp\left(-\frac{1}{8}(y_i - \Theta')^2 * \frac{1}{4}\right)}{\frac{cos^2\left(\frac{\pi(\Theta)}{4}\right)}{2} * \prod_{i=1}^{n} \frac{1}{2} exp\left(-\frac{1}{8}(y_i - \Theta)^2 * \frac{1}{4}\right)}$$

$$\gamma = \left(\frac{cos^2\left(\frac{\pi(\Theta)'}{4}\right)}{cos^2\left(\frac{\pi(\Theta)}{4}\right)}\right)^2 * \prod_{i=1}^{n} \frac{1}{2} exp\left(-\frac{1}{8}\left((y_i - \Theta')^2 - (y_i - \Theta)^2\right)\right)$$

$$\gamma = \left(\frac{cos^2\left(\frac{\pi(\Theta)'}{4}\right)}{cos^2\left(\frac{\pi(\Theta)}{4}\right)}\right)^2 * \prod_{i=1}^{n} \frac{1}{2} exp\left(-\frac{1}{8}((\Theta - \Theta') - (2y_i - \Theta - \Theta'))\right)$$

So from this we know:

$$p = 1 \ \wedge \left(\frac{cos^2\left(\frac{\pi(\Theta)'}{4}\right)}{cos^2\left(\frac{\pi(\Theta)}{4}\right)}\right)^2 * \prod_{i=1}^{n} \frac{1}{2} exp\left(-\frac{1}{8}((\Theta - \Theta') - (2y_i - \Theta - \Theta'))\right)$$

$or\ 1 \ \wedge \gamma$

I then used python to run the metropolis algorithm for 10000 observations to produce the following:

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

data = np.array([- 2.0,- 3.0,4.0,- 7.0,0.0,4.0])
theta = 0
thetas = []
thetas.append(theta)
```

```
#Start metropolis experiment and generate 10,000 random
observations
for i in np.arange(1,10500+1).reshape(-1):
    theta_proposal = round(- 2 + (4 * round(np.random.rand(1)
[0],4)),4)

    proposed = round(np.prod(np.exp((- 1 / 8) * (theta -
theta_proposal) * (np.multiply(data,2) - theta_proposal -
theta))),4)

    tau = round(((np.cos((math.pi * theta_proposal) / 4) /
np.cos((math.pi * theta) / 4)) ** 2) * proposed,4)

    tau_min = round(np.amin(tau),4)
    if (round(np.random.rand(1)[0],4) < tau_min):
        theta = theta_proposal
    thetas.append(theta)

#Drop the first 500 observations and create the final array
thetas = np.array(thetas[500:len(thetas)])
n, bins, patches = plt.hist(x=thetas, bins='auto',
color='#0504aa',alpha=0.7, rwidth=0.85)
print(f"Bayes estimator is the following: {np.mean(thetas)}")
#Produce the 97.5 to generate the 95% equitable set
print(f"Our 95 percent equitable set is: {np.percentile(thetas,
[2.5,97.5])}")
```

This produces the following histogram plot:

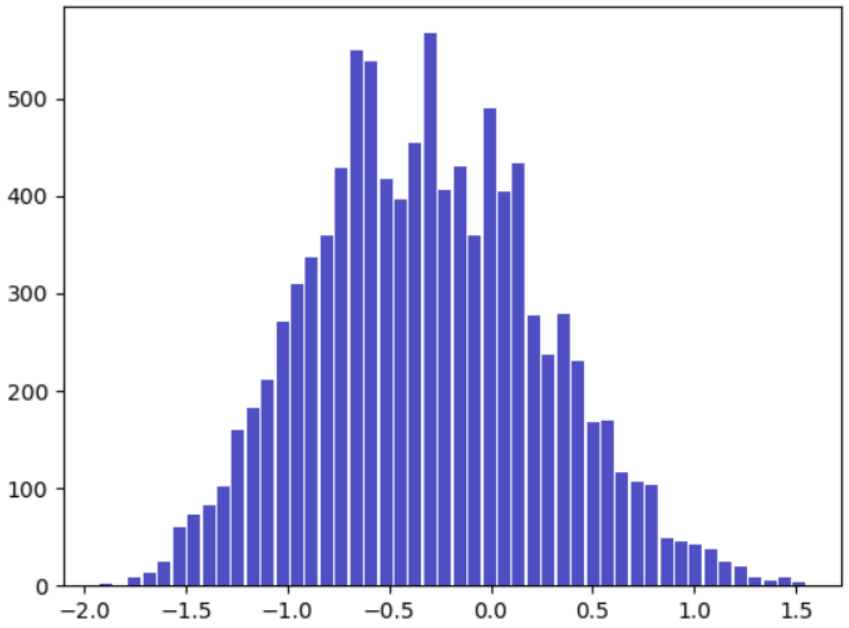

Figure 1: Histogram plot

b)

From the print output we get the answer for b as:

```
Bayes estimator is the following: -0.3077036696330367
Our 95 percent equitable set is: [-1.3644  0.8168]
```

2.

For this question we have the following known densities:

$$\tau_1 \sim Ga(a_1,\ b_1)$$

$$\Theta_1 \sim N\left(\Theta_{10},\ \frac{1}{\tau_{10}}\right)$$

We can determine that the joint distribution is proporitional to the following:

$$\propto\ \tau_1^{\frac{n}{2}}\ exp\left(\frac{\tau_1}{2}\sum_{i=1}^{n}(y_{1i}-\Theta_1)^2\right)\ exp(-\left(\frac{(\Theta_{10}-\Theta_1)^2}{2}\right)\ \tau_1^{a_1-1}exp\left(\tau_1-b_1\right)$$

Then use this to derive you conditional probability of Θ_1 as follows, where we replace μ with Θ_1

To find the full conditional for μ we select the terms from $f(y, \mu, \tau)$ that contain μ and normalize. Indeed,

$$\begin{aligned} \pi(\mu|\tau, y) &= \frac{\pi(\mu, \tau|y)}{\pi(\tau|y)} \\ &= \frac{\pi(\mu, \tau, y)}{\pi(\tau, y)} \propto \pi(\mu, \tau, y). \end{aligned}$$

Therefore we see that $\pi\left(\Theta_1 |\, \tau_1, y_{1i}\right)$ is proportional to the following and prove that it resembles a normal density:

$$\pi\left(\Theta_1 |\, \tau_1, y_{1i}\right) \propto exp(-\frac{1}{2}\left(\tau_{10} + n_1\tau_1\right)\left(\Theta_1 - \left(\frac{\tau_1 \sum_{i=1}^{n} y_{1i} + \Theta_{10}\tau_{10}}{\tau_{10} + n_1\tau_{10}}\right)^2\right)$$

The low protein diet resembles the same densities so I won't cover their derivation. The below below code will loop for 10000 iterations and determine τ_1, τ_2, Θ_1, Θ_2:

```
import numpy as np
from scipy.special import gamma
from scipy.stats import norm

np.random.RandomState(1)
y1 =
np.array([134.0,146.0,104.0,119.0,124.0,161.0,107.0,83.0,113.0,129.0,97.0,123.0
n1 = len(y1)
y2 = np.array([70.0,118.0,101.0,85.0,107.0,132.0,94.0])
n2 = len(y2)

#Set the number of iterations
n = 10000
#Set the empty lists to hold the updated values for theta and tau
for each variable (i.e 1,2)
thetas1 = []
```

```
taus1 = []
thetas2 = []
taus2 = []

#Generate the sum variables used for the below loop
sumy1 = sum(y1)
sumy2 = sum(y2)

#Set the initial values required
theta10 = 110
tau10 = 1 / 100
theta20 = 110
tau20 = 1 / 100
a1 = 0.01
b1 = 4
a2 = 0.01
b2 = 4
# start, initial values
theta1 = 110
tau1 = 1 / 100

theta2 = 110
tau2 = 1 / 100

for i in np.arange(1,n+1).reshape(-1):
    #Determine the value of updated value
    updatedTheta1 = np.sqrt(1 / (tau10 + 1 * tau1)) *
norm.ppf(np.random.rand(1))[0] + (tau1 * sumy1 + tau10 * theta10)
/ (tau10 + n1 * tau1)
    p1 = (b1 + 1 / 2) * sum((y1 - updatedTheta1) ** 2)
    updatedTau1 = np.random.gamma(a1 + n1 / 2,1 / p1)
    thetas1.append(updatedTheta1)
    taus1.append(updatedTau1)
    theta1 = updatedTheta1
    tau1 = updatedTau1
    updatedTheta2 = np.sqrt(1 / (tau20 + 1 * tau2)) *
norm.ppf(np.random.rand(1))[0] + (tau2 * sumy2 + tau20 * theta20)
/ (tau20 + n2 * tau2)
```

```
    p2 = b2 + 1 / 2 * sum((y2 - updatedTheta2) ** 2)
    updatedTau2 = np.random.gamma(a2 + n2 / 2,1 / p2)
    thetas2.append(updatedTheta2)
    taus2.append(updatedTau2)
    theta2 = updatedTheta2
    tau2 = updatedTau2


thetas1 = np.array(thetas1[500:len(thetas1)+1])
thetas2 = np.array(thetas2[500:len(thetas2)+1])
taus1 = np.array(taus1[500:len(taus1)+1])
taus2 = np.array(taus2[500:len(taus2)+1])
deltaThetas = thetas1 - thetas2


#Produce the 97.5 to generate the 95% equitable set
print(f"Bayes estimator for deltaThetas is the following:
{np.mean(deltaThetas)}")
print(f"Our 95 percent equitable set is:
{np.percentile(deltaThetas, [2.5,97.5])}")
print(f"The proportion of thetas1 > thetas2:
{np.count_nonzero(deltaThetas>0)/len(deltaThetas)}")
print(f"Bayes estimator for thetas1 is the following:
{np.mean(thetas1)}")
print(f"Bayes estimator for thetas2 is the following:
{np.mean(thetas2)}")
print(f"Bayes estimator for taus1 is the following:
{np.mean(taus1)}")
print(f"Bayes estimator for taus2 is the following:
{np.mean(taus2)}")
```

a)

Bayes estimator for θ_1 *is the following*: 111.91972903950226
Bayes estimator for θ_2 *is the following*: 105.0008964187764
Bayes estimator for τ_1 *is the following*: 0.00020819969979743016
Bayes estimator for τ_2 *is the following*: 0.002284944085825368

Bayes estimator for $\theta_1 - \theta_2$ *is the following*: 6.974751654551244

b)

$H_0 : \theta_1 > \theta_2 = 6645$

*The proportion of thetas*1 > *thetas*2: 0.6994736842105264

c)

Our 95 *percent equitable set is*: [−19.30683723 33.27821368] and we observe that it does contain 0.