Name: Rishal Shah     UnityID: rshah27

Bayesian Parameter Estimation

## Q1:

**Q2:**

From equation (A), we can see that the posterior distribution is quadratic and a Gaussian distribution as it is proportional to a Normal Distribution of the form:

where,

**Q3:**

**Q4:**

From eq. (C) we can identify the weights:

**Q5:**

From equation 4, we can see that weight and , are **inversely proportional** to their variances.

**Q6:**

Sum of weights =

**Q7:**

Since the weights , here numerator = 1 and denominator is greater than 1. So is between 0 and 1.

also has numerator = 1 and denominator greater than 1. So is also between 0 and 1.

Therefore, both the weights are between 0 and 1.

**Q8:**

From the previous answers 4-7, we can infer that, since both weights have their value between 0 and 1, the maximum value of is (when both weights are 1) and 0 (when both weights are 0). But, since both weights are inversely dependent on n, the edge cases of 0 and 1 are not possible. Hence, will always lie between and .

**Q9:**

Since,

,

where, (Normal distribution)

and (Normal distribution)

**This is because if there are two** mutually independent normal random variables with means μ1, μ2 and variances , then the linear combination: follows the normal distribution:

Hence,

**Q10:** Code (Python3):

import numpy as np

import scipy.stats as st

from matplotlib import pyplot as plt

sample = 20

line = np.linspace(0, 10, sample)

m0 = 4

sd0 = 0.8

d\_prior = st.norm(m0, sd0).pdf(line)

mx = 6

sdx = 1.5

d\_sample = st.norm(mx, sdx).pdf(line)

x\_t = st.norm(mx,sdx).rvs(sample)

variance = 1/((1/sd0\*\*2)+(sample/sdx\*\*2))

mean = variance \* ((m0/sd0\*\*2)+(np.mean(x\_t)\*sample/sdx\*\*2))

print("Posterior distibution:\nMean =",round(mean,3))

print("Variance =",round(variance,3))

d\_posterior = st.norm(mean, np.sqrt(variance)).pdf(line)

plt.figure(figsize=(20,20))

plt.plot(line,d\_prior,"r-",label='Prior Distribution')

plt.plot(line,d\_sample,"g-",label='Sample Distribution')

plt.plot(line,d\_posterior,"b-",label='Posterior Distribution')

plt.legend(loc='upper right')

plt.title('Probability Density Plot')

plt.ylabel('Probability Density')

plt.xlabel('X')

plt.show()

