DS-GA 1002 - Homework 4

Eric Niblock

October 2nd, 2020

1. **(Bayesian coin flip) Let us try out another prior for the Bayesian coin flip problem in the notes. We now model the parameter of the Bernouilli as being uniform between 1/2 and 1.**

   (a) **Briefly justify the model and compute the probability that the result of the coin flip is heads or tails under this model.**

   Working under the assumption that the uncle is cheating, and has somehow rigged the coin to land on heads, it makes sense to assume that parameter θ would be distributed such that $\theta \geq 0.5$ because θ is the probability of getting heads. So, we have,

$$f_{\tilde{\theta}}(\theta) = 2 \qquad\qquad for\ \theta \in [0.5, 1] \tag{1}$$

   Furthermore, we model the result of the coin flip $\tilde{r}$ as a Bernoulli random variable with parameter θ, such that $\tilde{r} = 0$ implies tails and $\tilde{r} = 1$ implies heads. Then,

$$p_{\tilde{r}}(0) = \int_{-\infty}^{\infty} f_{\tilde{\theta}}(u) p_{\tilde{r}|\tilde{\theta}}(0|u) du = \int_{1/2}^{1} 2(1-u) du = 0.25 \tag{2}$$

$$p_{\tilde{r}}(1) = \int_{-\infty}^{\infty} f_{\tilde{\theta}}(u) p_{\tilde{r}|\tilde{\theta}}(1|u) du = \int_{1/2}^{1} 2u\, du = 0.75 \tag{3}$$

   (b) **After the coin flip we update the distribution of the bias of the coin (i.e. the parameter of the Bernoulli that represents the coin flip) by conditioning it on the outcome. Compute the distribution if the outcome is tails and if the outcome is heads. Sketch any distributions you compute and explain why the drawing makes sense.**

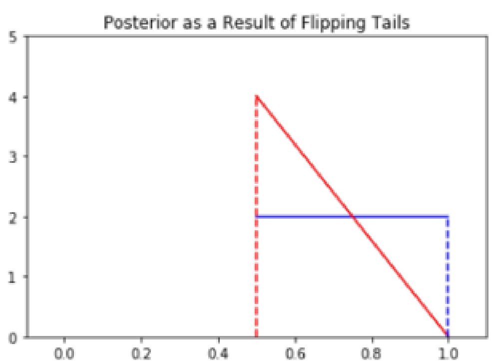
   We need to calculate the posterior probability given this new information. So, we first handle the scenario when the outcome is tails,

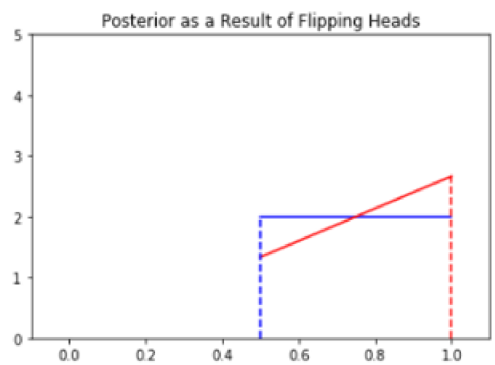
$$f_{\tilde{\theta}|\tilde{r}}(\theta|0) = \frac{f_{\tilde{\theta}}(\theta)\, p_{\tilde{r}|\tilde{\theta}}(0|\theta)}{p_{\tilde{r}}(0)} = \frac{2(1-\theta)}{0.25} = 8(1-\theta) \tag{4}$$

Where this is only valid for $\theta \in [0.5, 1]$. Now, we handle the situation when the outcome is heads,

$$f_{\tilde{\theta}|\tilde{r}}(\theta|1) = \frac{f_{\tilde{\theta}}(\theta)\, p_{\tilde{r}|\tilde{\theta}}(1|\theta)}{p_{\tilde{r}}(1)} = \frac{2(\theta)}{0.75} = \frac{8}{3}\theta \tag{5}$$

Where this is only valid for $\theta \in [0.5, 1]$.





These drawings make sense because we expect the posterior distributions (given in red) to shift from the prior distributions (given in blue) in the direction of the new information we inputted. Getting a tails shifted the distribution of θ towards 0 as expected, and getting a heads further shifted the distribution towards 1 as expected. A limitation of this model is that the resulting posteriors are always in the same domain as the priors. Both posterior pdfs still have a total enclosed area of one, as expected.

(c) You observe 100 coin flips and they all turn out to be tails (i.e. 0). Do you think you should reconsider your prior? If so, why?

Yes, the prior should be changed, because the prior only acts over the range $[0.5, 1]$. Even when we account for the 100 coin flips being tails, the calculated posterior will still only yield a pdf that is valid between $[0.5, 1]$ when in reality the actual value of parameter θ would likely be less than 0.5.

2. (Halloween parade) The city of New York hires you to estimate whether it will rain during the Halloween parade. Checking past data you determine that the chance of rain is 20%. You model this using a random variable $\tilde{r}$ with pmf

$$p_{\tilde{r}}(1) = 0.2 \qquad p_{\tilde{r}}(0) = 0.8$$

where $\tilde{r} = 1$ means that it rains and $\tilde{r} = 0$ that it doesn't. Your first idea is to be lazy and just use the forecast of a certain website. Analyzing data from previous forecasts, you model this with a random variable $\tilde{w}$ that satisfies

$$p(\tilde{w} = 1|\tilde{r} = 1) = 0.8 \qquad p(\tilde{w} = 0|\tilde{r} = 0) = 0.75$$

(a) What is the probability that the website is wrong?

The probability of $\tilde{w}$ being correct is given by,

$$P(\tilde{w}\ correct) = P(\tilde{w} = 1|\tilde{r} = 1)p_{\tilde{r}}(1) + (\tilde{w} = 0|\tilde{r} = 0)p_{\tilde{r}}(0) \tag{6}$$

$$P(\tilde{w}\ correct) = (0.8)(0.2) + (0.75)(0.8) = 0.76 \tag{7}$$

So then its obvious that,

$$P(\tilde{w}\ incorrect) = 1 - P(\tilde{w}\ correct) = 0.24 \tag{8}$$

(b) Unsatisfied with the accuracy of the website, you look at the data used for the forecast (they are available online). Surprisingly the relative humidity of the air is not used, so you decide to incorporate it in your prediction in the form of a random variable $\tilde{h}$. Is it more reasonable to assume that $\tilde{h}$ and $\tilde{w}$ are independent, or that they are conditionally independent given $\tilde{r}$? Explain why.

It makes more sense to consider $\tilde{h}$ and $\tilde{w}$ conditionally independent given $\tilde{r}$. This translates to: if we already know its raining, does knowing that the website predicted the rain convey any more information about the humidity? Additionally, would knowing the humidity convey any more information about the website's prediction? The answer to these questions should be assumed no, because the website does not take into account humidity data. It is our hope that $\tilde{h}$ and $\tilde{w}$ are actually *dependent*, since we hope that knowing information about the humidity will be beneficial in calculating the website's prediction.

(c) You assume that $\tilde{h}$ and $\tilde{w}$ are conditionally independent given $\tilde{r}$. More research establishes that conditioned on $\tilde{r} = 1$, $\tilde{h}$ is uniformly distributed between 0.5 and 0.7, whereas conditioned on $\tilde{r} = 0$, $\tilde{h}$ is uniformly distributed between 0.1 and 0.6. Compute the conditional pmf of $\tilde{r}$ given $\tilde{w}$ and $\tilde{h}$. Use the distribution to determine whether you would predict rain for any possible value of $\tilde{w}$ and $\tilde{h}$.

Given that $\tilde{w}$ and $\tilde{h}$ are conditionally independent given $\tilde{r}$, we are entitled to write,

$$p_{\tilde{r}|\tilde{h},\tilde{w}}(r|h,w) = \frac{p_{\tilde{r}}(r)p_{\tilde{w}|\tilde{r}}(w|r)f_{\tilde{h}|\tilde{r}}(h|r)}{\sum_{r=0}^{r=1} p_{\tilde{r}}(r)p_{\tilde{w}|\tilde{r}}(w|r)f_{\tilde{h}|\tilde{r}}(h|r)} \tag{9}$$

From this, we can calculate the various components of $p_{\tilde{r}|\tilde{h},\tilde{w}}$ based off of the fact that,

$$f_{\tilde{h}|\tilde{r}}(h \in [0.1, 0.6]|r = 0) = 2 \qquad f_{\tilde{h}|\tilde{r}}(h \in [0.5, 0.7]|r = 1) = 5 \tag{10}$$

And, furthermore,

$$p(\tilde{w} = 1|\tilde{r} = 1) = 0.8 \qquad p(\tilde{w} = 0|\tilde{r} = 0) = 0.75 \tag{11}$$

$$p(\tilde{w} = 0|\tilde{r} = 1) = 0.2 \qquad p(\tilde{w} = 1|\tilde{r} = 0) = 0.25 \tag{12}$$

Then we have the following four cases,

$$\begin{aligned} p_{\tilde{r}|\tilde{h},\tilde{w}}(0|[0.1, 0.5], 0) &= \frac{p_{\tilde{r}}(0)p_{\tilde{w}|\tilde{r}}(0|0)f_{\tilde{h}|\tilde{r}}([0.1, 0.5]|0)}{\sum_{r=0}^{r=1} p_{\tilde{r}}(r)p_{\tilde{w}|\tilde{r}}(0|r)f_{\tilde{h}|\tilde{r}}([0.1, 0.5]|r)} \\ &= \frac{(0.8)(0.75)(2)}{(0.8)(0.75)(2) + (0.2)(0.2)(0)} = 1 \end{aligned} \tag{13}$$

$$p_{\tilde{r}|\tilde{h},\tilde{w}}(1|[0.1, 0.5], 0) = 0 \tag{14}$$

$$p_{\tilde{r}|\tilde{h},\tilde{w}}(0|[0.1, 0.5], 1) = 1 \tag{15}$$

$$p_{\tilde{r}|\tilde{h},\tilde{w}}(1|[0.1,0.5],1) = 0 \tag{16}$$

$$\begin{aligned} p_{\tilde{r}|\tilde{h},\tilde{w}}(0|[0.5,0.6],0) &= \frac{p_{\tilde{r}}(0)p_{\tilde{w}|\tilde{r}}(0|0)f_{\tilde{h}|\tilde{r}}([0.5,0.6]|0)}{\sum_{r=0}^{r=1} p_{\tilde{r}}(r)p_{\tilde{w}|\tilde{r}}(0|r)f_{\tilde{h}|\tilde{r}}([0.5,0.6]|r)} \\ &= \frac{(0.8)(0.75)(2)}{(0.8)(0.75)(2)+(0.2)(0.2)(5)} = \frac{6}{7} \end{aligned} \tag{17}$$

$$\begin{aligned} p_{\tilde{r}|\tilde{h},\tilde{w}}(1|[0.5,0.6],0) &= \frac{p_{\tilde{r}}(1)p_{\tilde{w}|\tilde{r}}(0|1)f_{\tilde{h}|\tilde{r}}([0.5,0.6]|1)}{\sum_{r=0}^{r=1} p_{\tilde{r}}(r)p_{\tilde{w}|\tilde{r}}(0|r)f_{\tilde{h}|\tilde{r}}([0.5,0.6]|r)} \\ &= \frac{(0.2)(0.2)(5)}{(0.8)(0.75)(2)+(0.2)(0.2)(5)} = \frac{1}{7} \end{aligned} \tag{18}$$

$$\begin{aligned} p_{\tilde{r}|\tilde{h},\tilde{w}}(0|[0.5,0.6],1) &= \frac{p_{\tilde{r}}(0)p_{\tilde{w}|\tilde{r}}(1|0)f_{\tilde{h}|\tilde{r}}([0.5,0.6]|0)}{\sum_{r=0}^{r=1} p_{\tilde{r}}(r)p_{\tilde{w}|\tilde{r}}(1|r)f_{\tilde{h}|\tilde{r}}([0.5,0.6]|r)} \\ &= \frac{(0.8)(0.25)(2)}{(0.8)(0.25)(2)+(0.2)(0.8)(5)} = \frac{1}{3} \end{aligned} \tag{19}$$

$$\begin{aligned} p_{\tilde{r}|\tilde{h},\tilde{w}}(1|[0.5,0.6],1) &= \frac{p_{\tilde{r}}(1)p_{\tilde{w}|\tilde{r}}(1|1)f_{\tilde{h}|\tilde{r}}([0.5,0.6]|1)}{\sum_{r=0}^{r=1} p_{\tilde{r}}(r)p_{\tilde{w}|\tilde{r}}(1|r)f_{\tilde{h}|\tilde{r}}([0.5,0.6]|r)} \\ &= \frac{(0.2)(0.8)(5)}{(0.8)(0.25)(2)+(0.2)(0.8)(5)} = \frac{2}{3} \end{aligned} \tag{20}$$

$$\begin{aligned} p_{\tilde{r}|\tilde{h},\tilde{w}}(0|[0.6,0.7],0) &= \frac{p_{\tilde{r}}(0)p_{\tilde{w}|\tilde{r}}(0|0)f_{\tilde{h}|\tilde{r}}([0.6,0.7]|0)}{\sum_{r=0}^{r=1} p_{\tilde{r}}(r)p_{\tilde{w}|\tilde{r}}(0|r)f_{\tilde{h}|\tilde{r}}([0.6,0.7]|r)} \\ &= \frac{(0.8)(0.75)(0)}{(0.8)(0.75)(0)+(0.2)(0.2)(5)} = 0 \end{aligned} \tag{21}$$

$$p_{\tilde{r}|\tilde{h},\tilde{w}}(1|[0.6,0.7],0) = 1 \tag{22}$$

$$p_{\tilde{r}|\tilde{h},\tilde{w}}(0|[0.6,0.7],1) = 0 \tag{23}$$

$$p_{\tilde{r}|\tilde{h},\tilde{w}}(1|[0.6,0.7],1) = 1 \tag{24}$$

This completely describes the conditional pmf $p_{\tilde{r}|\tilde{h},\tilde{w}}(r|h,w)$. Any other combination of $r|h,w$ would yield a probability of zero.

(d) What is the probability that you make a mistake?

We have complete certainty about the results unless $h \in [0.5, 0.6]$. This means that any mistake would come from within this range. In fact, a mistake occurs when $r \neq w$, so, we have,

$$\begin{aligned}
P(mistake) &= P_{\tilde{r},\tilde{h},\tilde{w}}(1, [0.5, 0.6], 0) + P_{\tilde{r},\tilde{h},\tilde{w}}(0, [0.5, 0.6], 1) \\
&= \int_{0.5}^{0.6} f_{\tilde{h}|\tilde{r}}(h|1) p_{\tilde{w}|\tilde{r}}(0|1) p_{\tilde{r}}(1) + f_{\tilde{h}|\tilde{r}}(h|0) p_{\tilde{w}|\tilde{r}}(1|0) p_{\tilde{r}}(0)\, dr \\
&= \int_{0.5}^{0.6} (5)(0.2)(0.2) + (2)(0.25)(0.8) dr \\
&= 0.06
\end{aligned} \tag{25}$$

3. (Markov chain) In this problem we consider the Markov chain corresponding to the state diagram in Figure 1. Derive an expression for the state vector of the Markov chain at an arbitrary time i assuming that we always start at state A.

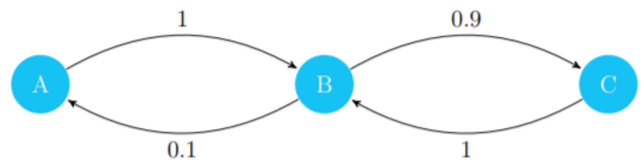


Figure 1: State diagram of a Markov chain with periodic states A and C.

First, we must generate the transition matrix, which is as follows,

$$T_{\tilde{x}} = \begin{bmatrix} 0 & 0.1 & 0 \\ 1 & 0 & 1 \\ 0 & 0.9 & 0 \end{bmatrix} \tag{26}$$

Where, as an example, $T_{1,2} = 0.1$ is the probability of starting in state B and transitioning to state A. Furthermore, we know that,

$$p_{\tilde{x}[i]} = T_{\tilde{x}}^{i} p_{\tilde{x}[1]} \tag{27}$$

Where the initial state vector is trivial since we always start in state A. So, we have,

$$p_{\tilde{x}[i]} = \begin{bmatrix} 0 & 0.1 & 0 \\ 1 & 0 & 1 \\ 0 & 0.9 & 0 \end{bmatrix}^{i} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{28}$$

With $i \in \{0, 1, 2, ...\}$. Now, when i is odd, we have,

$$p_{\tilde{x}[i,odd]} = \begin{bmatrix} 0 & 0.1 & 0 \\ 1 & 0 & 1 \\ 0 & 0.9 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \tag{29}$$

And when i is even, we have,

$$p_{\tilde{x}[i,even]} = \begin{bmatrix} 0.1 & 0 & 0.1 \\ 0 & 1 & 0 \\ 0.9 & 0 & 0.9 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.1 \\ 0 \\ 0.9 \end{bmatrix} \tag{30}$$

So, the system oscillates between two different state vectors, always returning to state B when i is odd, and oscillating between A and C when i is even.

4. **(Heart-disease detection) A hospital is interested in developing a system for automatic heart-disease detection. Your task is to use the data in the *heart_disease_data.npz* to detect heart disease in patients. You model heart disease as a random variable $\tilde{h}$ that indicates whether the patient suffers from heart disease or not:**

$$\tilde{h} = \begin{cases} 0, & \textbf{if patient does not suffer from heart disease} \\ 1, & \textbf{if patient does suffer from heart disease} \end{cases} \tag{31}$$

The available data contain the patient's sex, the type of chest pain experienced by the patient and the cholesterol of the patient. We model these quantities as the random variables $\tilde{s}$, $\tilde{c}$ and $\tilde{x}$ respectively, where

$$\tilde{s} = \begin{cases} 0, & \textbf{if patient is female} \\ 1, & \textbf{if patient is male} \end{cases} \tag{32}$$

$$\tilde{c} = \begin{cases} 0, & \textbf{if the pain is typical angina} \\ 1, & \textbf{if the pain is atypical angina} \\ 2, & \textbf{for other types of chest pain} \\ 3, & \textbf{if there is no chest pain} \end{cases} \tag{33}$$

and $\tilde{x}$ is a continuous random variable.

(a) **Derive the MAP estimate of $\tilde{h}$ given $\tilde{s}$ and $\tilde{c}$ as a function of the pmf of $\tilde{h}(p_{\tilde{h}})$ and the conditional pmfs $p_{\tilde{s}|\tilde{h}}$ and $p_{\tilde{c}|\tilde{h}}$. The MAP estimate is defined as the mode of the posterior distribution. Assume that if we know whether a patient is suffering from heart disease, the sex of the patient and the type of chest pain experienced by the patient are conditionally independent.**

We know that by Bayes rule, we have,

$$p_{\tilde{h}|\tilde{s},\tilde{c}}(h|s,c) = \frac{p_{\tilde{h}}(h)p_{\tilde{s},\tilde{c}|\tilde{h}}(s,c|h)}{p_{\tilde{s},\tilde{c}}(s,c)} \tag{34}$$

Now, since the denominator has no dependence on h, we only need to worry about the numerator. We want to find the value of h that maximizes the numerator given values s and c. Furthermore, we can simplify the numerator through conditional independence assumptions. The result becomes,

$$MAP_{h_{s,c}} = \begin{cases} 0, & \text{if } p_{\tilde{h}}(1)p_{\tilde{s}|\tilde{h}}(s|1)p_{\tilde{c}|\tilde{h}}(c|1) < p_{\tilde{h}}(0)p_{\tilde{s}|\tilde{h}}(s|0)p_{\tilde{c}|\tilde{h}}(c|0) \\ 1, & \text{otherwise} \end{cases} \quad (35)$$

(b) **Complete the corresponding part of the script *hw4q4.py* to estimate the necessary probability mass functions from the data. The training data consists of 218 patients and is provided in the arrays *data["heart disease"]*, *data["sex"]* and *data["chest pain"]*. Apply the MAP decision rule you derived in part (a) to predict whether a group of 50 other patients, whose information is stored in the vectors *data["sex test"]* and *data["chest pain test"]*, suffer from heart disease. Calculate the error rate (i.e. the proportion of predictions that are incorrect) by comparing your results to data["heart disease test"], which indicates whether the patients suffer from heart disease or not.**

Observe the following code, which creates the MAP estimation and tests it up against the test data. The resulting error is shown.

```python
### --- QUESTION (B) --- ###
# Estimate the pmf of H
P_H0 = count_x_eq_val(data['heart_disease'], 0)
P_H1 = count_x_eq_val(data['heart_disease'], 1)

# Estimate the conditional pmf of S given H
P_S_H0 = np.zeros(2)
P_S_H1 = np.zeros(2)

H_0people = np.array([data['sex'][i] for i in ind_x_eq_val(data['heart_disease'], 0)])
H_1people = np.array([data['sex'][i] for i in ind_x_eq_val(data['heart_disease'], 1)])

for ind_S in range(2):
    P_S_H0[ind_S] = count_x_eq_val(H_0people, ind_S)
    P_S_H1[ind_S] = count_x_eq_val(H_1people, ind_S)

H_0people = np.array([data['chest_pain'][i] for i in ind_x_eq_val(data['heart_disease'], 0)])
H_1people = np.array([data['chest_pain'][i] for i in ind_x_eq_val(data['heart_disease'], 1)])

# Estimate the conditional pmf of C given H
P_C_H0 = np.zeros(4)
P_C_H1 = np.zeros(4)
for ind_C in range(4):
    P_C_H0[ind_C] = count_x_eq_val(H_0people, ind_C)
    P_C_H1[ind_C] = count_x_eq_val(H_1people, ind_C)

# Calculate the MAP estimate

map_matrix = np.zeros((2,4))
for i in range(2):
    for j in range(4):
        if P_H1*P_S_H1[i]*P_C_H1[j] < P_H0*P_S_H0[i]*P_C_H0[j]:
            map_matrix[i][j] = 0
        else:
            map_matrix[i][j] = 1

MAP_estimate_S_C = map_matrix

# Calculate the error rate - i.e. the proportion of all predictions that were incorrect

heart_disease_test = []
for p in range(len(data['heart_disease_test'])):
    i = int(data['sex_test'][p])
    j = int(data['chest_pain_test'][p])
    heart_disease_test.append(MAP_estimate_S_C[i][j])


error_rate_S_C = np.sum(np.array(heart_disease_test) == data['heart_disease_test'])/len(heart_disease_test)

print ("Probability of error " + str(round(1-error_rate_S_C,5)))
```

```
Probability of error 0.18
```

(c) Derive a MAP estimate of $\tilde{h}$ given $\tilde{s}$, $\tilde{c}$ and $\tilde{x}$ that only depends on the pmf of $\tilde{h}$, $p_{\tilde{h}}$, the conditional pmfs $p_{\tilde{s}|\tilde{h}}(s|h)$ and $p_{\tilde{c}|\tilde{h}}(c|h)$ and the conditional pdf $f_{\tilde{x}|\tilde{h}}(x|h)$, assuming that if we know whether a patient is suffering from heart disease, the sex, type of chest pain and cholesterol level of the patient are all independent.

Again, we have a similar expression for our map estimate, though we first apply Bayes rule,

$$p_{\tilde{h}|\tilde{s},\tilde{c},\tilde{x}}(h|s,c,x) = \frac{p_{\tilde{h}}(h)p_{\tilde{s},\tilde{c},\tilde{x}|\tilde{h}}(s,c,x|h)}{p_{\tilde{s},\tilde{c},\tilde{x}}(s,c,x)} \tag{36}$$

Again, due to conditional independence, we can simplify the expression. This time, we show the expression,

$$p_{\tilde{h}|\tilde{s},\tilde{c},\tilde{x}}(h|s,c,x) = \frac{p_{\tilde{h}}(h)p_{\tilde{s}|\tilde{h}}(s|h)p_{\tilde{c}|\tilde{h}}(c|h)f_{\tilde{x}|\tilde{h}}(x|h)}{p_{\tilde{s},\tilde{c},\tilde{x}}(s,c,x)} \tag{37}$$

Then, as before, we ignore the denominator, which yields the MAP estimation by finding the maximization of h given values for s, c, and x. The map rule then becomes,

$$MAP_{h_{s,c,x}} = \begin{cases} 0, & \text{if } p_{\tilde{h}}(1)p_{\tilde{s}|\tilde{h}}(s|1)p_{\tilde{c}|\tilde{h}}(c|1)f_{\tilde{x}|\tilde{h}}(x|1) < p_{\tilde{h}}(0)p_{\tilde{s}|\tilde{h}}(s|0)p_{\tilde{c}|\tilde{h}}(c|0)f_{\tilde{x}|\tilde{h}}(x|0) \\ 1, & \text{otherwise} \end{cases} \tag{38}$$

(d) You decide to model the cholesterol level of a patient conditioned on whether he or she suffers from heart disease as a Gaussian random variable. For both cases, complete the corresponding part of the script *hw4q4.py* to obtain the ML estimates of the conditional distributions from the data in cholesterol and compare the estimated pdf to the histogram of the data.

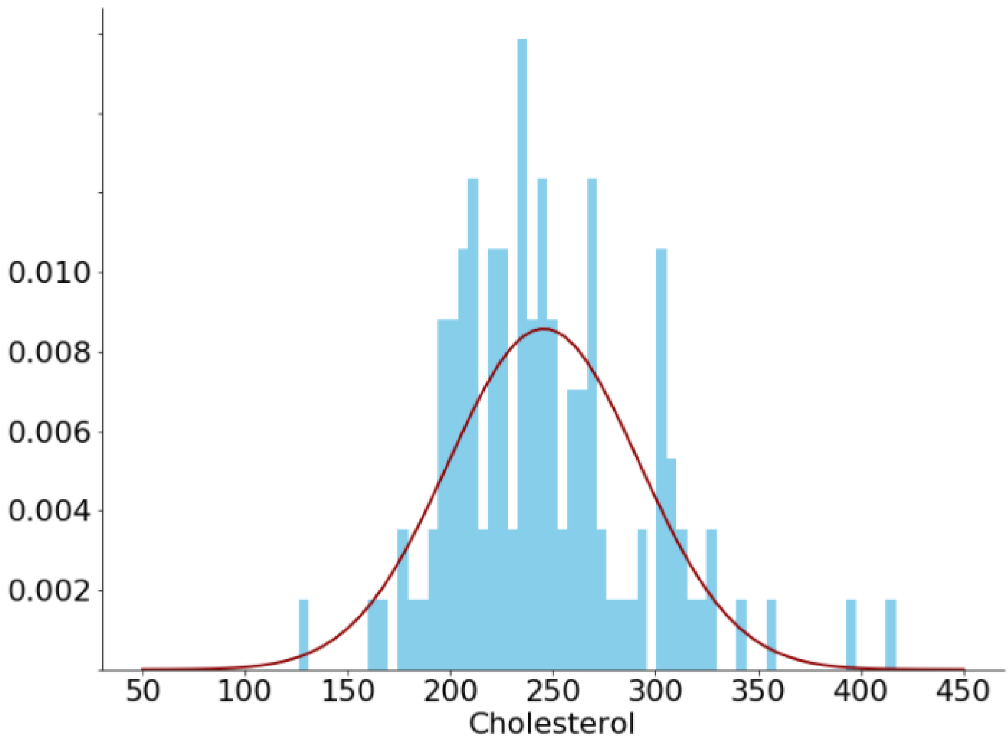
Below is the result of the computed ML estimates with accompanying code and resulting plots. The first plot is the histogram and pdf for cholesterol levels conditioned on not having heart disease. The second plot is the histogram and pdf for cholesterol levels conditioned on having heart disease.

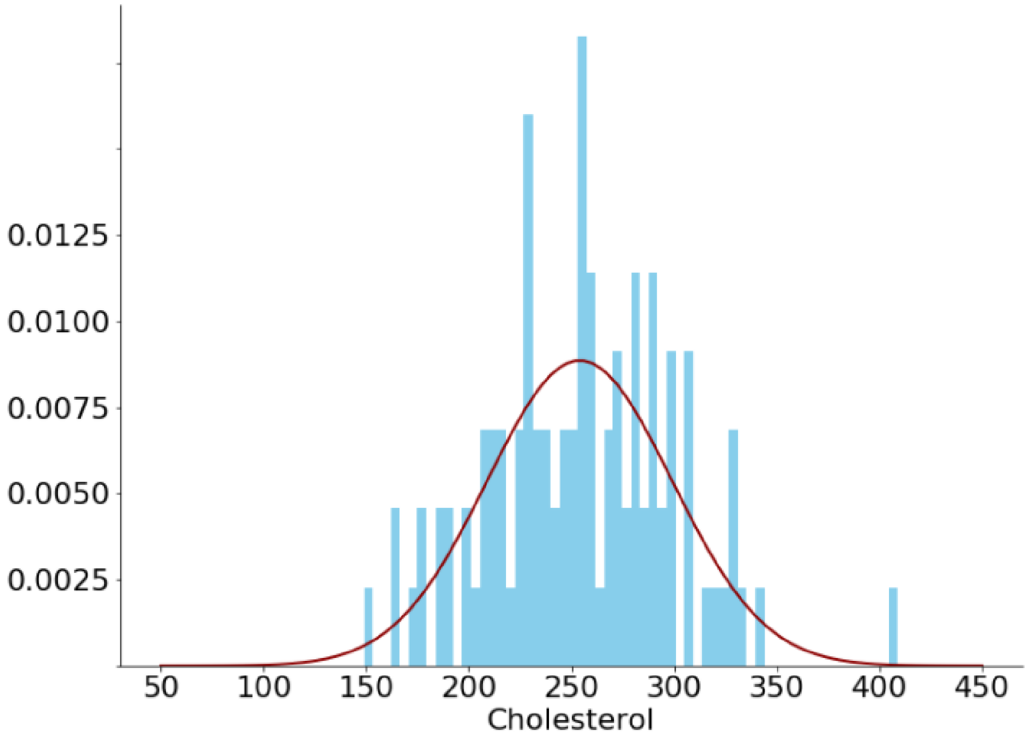
```python
### --- QUESTION (D) --- ###
## Estimate conditional pdf of X given H
mean_X_H = np.zeros(2)
std_X_H = np.zeros(2)
H_0people = np.array([data['cholesterol'][i] for i in ind_x_eq_val(data['heart_disease'], 0)])
H_1people = np.array([data['cholesterol'][i] for i in ind_x_eq_val(data['heart_disease'], 1)])
cond_H = [H_0people,H_1people]
mean_X_H[0]= np.mean(H_0people)
std_X_H[0] = np.std(H_0people)
mean_X_H[1]= np.mean(H_1people)
std_X_H[1] = np.std(H_1people)

n_plot = 100
for i in range(2):
    plt.figure(figsize=(12, 9))
    ax = plt.subplot(111)
    ax.spines["top"].set_visible(False)
    ax.spines["right"].set_visible(False)
    ax.get_xaxis().tick_bottom()
    ax.get_yaxis().tick_left()
    yticks = ax.yaxis.get_major_ticks()
    yticks[0].label1.set_visible(False)
    plt.xticks(fontsize=22)
    plt.yticks(fontsize=22)
    plt.xlabel("Cholesterol", fontsize=22)

    plt.hist(cond_H[i],
             60,normed=True,edgecolor = "none", color="skyblue")

    plt.plot(np.linspace(50, 450, n_plot),gaussian(np.linspace(50, 450, n_plot),
                     mean_X_H[i], std_X_H[i]), color="darkred", lw=2)
```





(e) **Complete the corresponding part of the script *hw4q4.py* to apply your MAP decision rule incorporating the cholesterol data and compute the new error rate (using the cholesterol rates of the 50 new patients, stored in *data["cholesterol test"]*. Do you trust this result?**

Below is the new MAP decision rule which incorporates the cholesterol data.

```
### --- QUESTION (E) --- ###
# Calculate the MAP estimate

map_matrix = np.zeros((2,4,451))
for i in range(2):
    for j in range(4):
        for k in range(50,451):
            if P_H1*P_S_H1[i]*P_C_H1[j]*gaussian(k, mean_X_H[1], std_X_H[1]) <\
            P_H0*P_S_H0[i]*P_C_H0[j]*gaussian(k, mean_X_H[0], std_X_H[0]):
                map_matrix[i][j][k] = 0
            else:
                map_matrix[i][j][k] = 1


MAP_estimate_S_C_X = map_matrix

# Calculate the error rate

heart_disease_test = []
for p in range(len(data['heart_disease_test'])):
    i = int(data['sex_test'][p])
    j = int(data['chest_pain_test'][p])
    k = int(data['cholesterol_test'][p])
    heart_disease_test.append(MAP_estimate_S_C_X[i][j][k])

error_rate_S_C_X = np.sum(np.array(heart_disease_test) == data['heart_disease_test'])/len(heart_disease_test)
print ("Probability of error using cholesterol " + str(round(1-error_rate_S_C_X,5)))
```

```
Probability of error using cholesterol 0.14
```

I do not trust this result because it seems we have made some conditional independence assumptions that might not necessarily be true. For example, $\tilde{x}$ and $\tilde{c}$ do not seem to be conditionally independent given $\tilde{h}$. I would expect that even if we know a patient has heart disease, the severity of their chest pain would provide more information about what we would estimate their cholesterol levels to be. The reverse of this would also make logical sense. Given a patient with heart disease, the severity of their chest pain could potentially be informed by their cholesterol level. Given that we used conditional independence assumptions in our derivation, the results do not appear to be trustworthy, even though our error decreased.

(f) We have made some conditional independence assumptions that do not necessarily hold. Another option would have been to estimate the joint distribution of all the random variables from the data. Is this a good idea?

This would not have been a good idea. The curse of dimensionality implies that the number of parameters we would have to estimate would become exponential if we were to attempt to calculate all of the joint pdfs. This process would be cumbersome and very inefficient. Furthermore, given the amount of data that we have, the joint pdfs would likely be based off of relatively few data points, since our data set is so small.