DSC 140A - Homework 07

Due: Wednesday, February 26

Instructions: Write your solutions to the following problems either by typing them or handwriting them on another piece of paper or on an iPad/tablet. Show your work or provide justification unless otherwise noted; submissions that don't show work might lose credit. If you write code to solve a problem, include the code by copy/pasting or as a screenshot. You may use `numpy`, `pandas`, `matplotlib` (or another plotting library), and any standard library module, but no other third-party libraries unless specified. Submit homeworks via Gradescope by 11:59 PM.

A LaTeX template is provided at `http://dsc140a.com`, next to where you found this homework. Using it is totally optional, but encouraged if you plan to go to grad school. See this video for a quick introduction to LaTeX.

Problem 1.

The file linked below contains a data set of 150 samples. The first column contains a single continuous feature, X, assumed to have been drawn from an unknown probability density. The second column contains the binary class label Y.

`https://f000.backblazeb2.com/file/jeldridge-data/011-univariate_density_estimation/data.csv`

In this problem, use a histogram estimator with bins $[0, 1.5), [1.5, 3), \ldots, [13.5, 15)$ to estimate the requested probabilities. In each part, show your code and provide your reasoning.

Hint: you may find `np.histogram` useful.

a) Estimate $\mathbb{P}(Y = 1 \mid X = 6.271)$ directly.

Solution: To estimate $\mathbb{P}(Y = 1 \mid X = 6.271)$ directly, we filter the data to include only the samples that are within the same histogram bin as $x = 6.271$, and then find the fraction of those samples that have $y = 1$.

In code:

```
data = np.loadtxt('data.csv', delimiter=',')

# histogram bin containing 6.271
bin_lower, bin_upper = (6, 7.5)

# split the two columns of data into x and y arrays
x, y = data.T

# filter the data to only include the x values in the bin
y[(bin_lower <= x) & (x < bin_upper)].mean()
```

This gives 0.714, approximately.

b) Estimate $\mathbb{P}(Y = 1 \mid X = 6.271)$ by estimating all of: 1) the marginal density $p_X(x)$, 2) the class-conditional density $p_X(x \mid Y = 1)$, and 3) the class prior $\mathbb{P}(Y = 1)$, and then applying Bayes' rule.

Solution:

```
# filter the data into two subsets:
# the data from class 1 and the data from class 0
samples_1 = data[data[:, 1] == 1][:, 0]
samples_0 = data[data[:, 1] == 0][:, 0]

bins = np.arange(0, 15.1, 1.5)

# create class-conditional density histograms
p_1 = np.histogram(samples_1, bins=bins, density=True)[0]
p_0 = np.histogram(samples_0, bins=bins, density=True)[0]

# create a histogram for the marginal density
p = np.histogram(x, bins=bins, density=True)[0]

# our estimate of P(Y = 1)
p_y1 = y.mean()

# the point x = 6.271 falls within the fifth bin (bin at index 4)
# of the histogram. To compute p(x | Y = 1) * P(Y = 1) / p(x), we write:
p_1[4] * p_y1 / p[4]
```

This yields the same answer as part (a): 0.714.

c) For what values of $x \in [0, 15]$ will the Bayes classifier predict $y = 1$?

Solution: The Bayes classifier predicts $y = 1$ whenever $p_X(x \,|\, Y = 1)\mathbb{P}(Y = 1) > p_X(x \,|\, Y = 0)\mathbb{P}(Y = 0)$. Because we are using a histogram estiamtor, the density estimates are piecewise constant, and the prediction can only change at the bin boundaries. Therefore, it's enough to check the prediction within each bin.

We can do this in one line of code: `p_1 * y.mean() > p_0 * (1 - y.mean())`

The result is:

```
array([ True, True, True, True, True, False, False, False, False, False])
```

This tells us that the prediction is for $y = 1$ in the first 5 bins, and for $y = 0$ in the remaining bins. The first five bins cover the interval $[0, 7.5)$, and so the Bayes classifier will predict $y = 1$ for x in this interval.

Problem 2.

The Rayleigh distribution has pdf:

$$p(x) = \frac{x}{\sigma^2} e^{-x^2/(2\sigma^2)},$$

where σ is a parameter.

Suppose a data set of points $x_1, \ldots, x_n$ is drawn from a Rayleigh distribution with unknown parameter σ. It was shown in discussion section that the log-likelihood of σ given this data is:

$$\tilde{L}(\sigma | x_1, \ldots, x_n) = n \ln \frac{1}{\sigma^2} + \sum_{i=1}^{n} \ln x_i - \frac{1}{2\sigma^2} \sum_{i=1}^{n} x_i^2$$

Show that the maximum likelihood estimate of σ is:

$$\sigma_{\text{MLE}} = \sqrt{\frac{1}{2n}\sum_{i=1}^n x_i^2}.$$

Solution: We start by taking the derivative of the log likelihood with respect to σ:

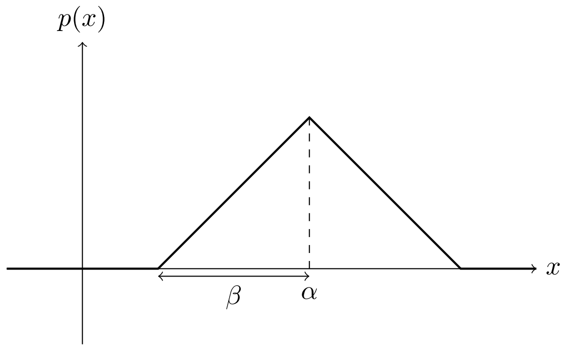
$$\begin{aligned}\frac{d}{d\sigma}\tilde{L}(\sigma|x_1,\ldots,x_n) &= \frac{d}{d\sigma}\left[n \ln \frac{1}{\sigma^2} + \sum_{i=1}^n \ln x_i - \frac{1}{2\sigma^2}\sum_{i=1}^n x_i^2\right] \\ &= n\sigma^2(-2)\sigma^{-3} + \frac{2}{2\sigma^3}\sum_{i=1}^n x_i^2 \\ &= -\frac{2n}{\sigma} + \frac{2}{2\sigma^3}\sum_{i=1}^n x_i^2\end{aligned}$$

Setting to zero and solving for σ:

$$\begin{aligned}-\frac{2n}{\sigma} + \frac{2}{2\sigma^3}\sum_{i=1}^n x_i^2 &= 0 \\ \implies \frac{2}{2\sigma^3}\sum_{i=1}^n x_i^2 &= \frac{2n}{\sigma} \\ \implies \frac{1}{\sigma^2}\sum_{i=1}^n x_i^2 &= 2n \\ \implies \sigma &= \sqrt{\frac{1}{2n}\sum_{i=1}^n x_i^2}\end{aligned}$$

Problem 3.

Justin's triangle is a parametric density function p that looks like a triangle. It has two parameters, α and β, which control the location and width of the triangle, respectively. A plot of the pdf is shown below:



a) Show that $p(\alpha) = \frac{1}{\beta}$.

Solution: Let h be the (unknown) height of the triangle. The full triangle is made up of two smaller right triangles, each with base β and height h. The area of each of these small triangles is $\frac{1}{2}\beta h$. The total area of the full triangle is therefore βh. Since the area of the full triangle must be 1 for this to be a valid density, we solve $\beta h = 1$ for h to get $h = \frac{1}{\beta}$.

b) Write down the formula for the density function $p(x)$.

Solution:

$$p(x) = \begin{cases} 0 & \text{if } x < \alpha - \beta \\ \frac{1}{\beta^2}(x - \alpha + \beta) & \text{if } \alpha - \beta \le x < \alpha \\ -\frac{1}{\beta^2}(x - \alpha - \beta) & \text{if } \alpha \le x < \alpha + \beta \\ 0 & \text{if } x > \alpha + \beta \end{cases}$$

c) Let $\mathcal{X} = \{2, 3, 5, 7, 8\}$ be a data set of 5 points. Note that this data set is symmetric around the middle point, 5.

It can be shown that, in this case, the maximum likelihood estimate for α is 5.

What is the maximum likelihood estimate for β?

Hint: you might get to a point where you need to solve a quadratic equation. You can use the quadratic formula to find the roots, but be careful to check which root is valid for this problem.

Solution: We first write down the likelihood function with respect to β:

$$\begin{aligned} \mathcal{L}(\beta) &= \prod_{i=1}^{5} p(x_i) \\ &= p(x_1) \cdot p(x_2) \cdot p(x_3) \cdot p(x_4) \cdot p(x_5) \end{aligned}$$

Assume that $x_1, x_2, \ldots, x_5$ are in the order that they appear in $\mathcal{X}$. That is, $x_1 = 2$, $x_2 = 3$, and so on. The x_1 and x_2 fall into the second case of the piecewise function, and the x_4 and x_5 fall into the third case. $x_3 = 5$, and so we already know that $p(x_3) = \frac{1}{\beta}$ from the first part. Using our formula for p from part (b), we can write down the likelihood function as:

$$\begin{aligned} &= \underbrace{\frac{1}{\beta^2}(2 - 5 + \beta)}_{p(x_1)} \cdot \underbrace{\frac{1}{\beta^2}(3 - 5 + \beta)}_{p(x_2)} \cdot \frac{1}{\beta} \cdot \underbrace{-\frac{1}{\beta^2}(7 - 5 - \beta)}_{p(x_4)} \cdot \underbrace{-\frac{1}{\beta^2}(8 - 5 - \beta)}_{p(x_5)} \\ &= \frac{1}{\beta^2}(\beta - 3) \cdot \frac{1}{\beta^2}(\beta - 2) \cdot \frac{1}{\beta} \cdot \frac{1}{\beta^2}(\beta - 2) \cdot \frac{1}{\beta^2}(\beta - 3) \end{aligned}$$

Now, writing the log-likehood function to make the calculus easier:

$$\begin{aligned} \tilde{L}(\beta) &= -2\ln\beta + \ln(\beta - 3) - 2\ln\beta + \ln(\beta - 2) - \ln\beta - 2\ln\beta + \ln(\beta - 2) - 2\ln\beta + \ln(\beta - 3) \\ &= -9\ln\beta + 2\ln(\beta - 3) + 2\ln(\beta - 2) \end{aligned}$$

Differentiating with respect to β:

$$\frac{d\tilde{L}}{d\beta} = -\frac{9}{\beta} + \frac{2}{\beta - 3} + \frac{2}{\beta - 2}$$

Setting this to zero and solving for β:

$$\begin{aligned}
\frac{d\tilde{L}}{d\beta} = 0 &\implies -\frac{9}{\beta} + \frac{2}{\beta-3} + \frac{2}{\beta-2} = 0 \\
&\implies \frac{9}{\beta} = \frac{2}{\beta-3} + \frac{2}{\beta-2} \\
&\implies 9(\beta-3)(\beta-2) = 2\beta(\beta-2) + 2\beta(\beta-3) \\
&\implies 9(\beta^2 - 5\beta + 6) = 2\beta^2 - 4\beta + 2\beta^2 - 6\beta \\
&\implies 9\beta^2 - 45\beta + 54 = 4\beta^2 - 10\beta \\
&\implies 5\beta^2 - 35\beta + 54 = 0
\end{aligned}$$

Making use of the quadratic formula, we find:

$$\begin{aligned}
&\implies \beta = \frac{35 \pm \sqrt{35^2 - 4 \cdot 5 \cdot 54}}{2 \cdot 5} \\
&\implies \beta = \frac{35 \pm \sqrt{145}}{6} \\
&\implies \beta = \frac{35 + \sqrt{145}}{6} \approx 4.704 \text{ or } \beta = \frac{35 - \sqrt{145}}{6} \approx 2.296
\end{aligned}$$

The solution $\beta = 2.296$ is not valid because it would make the density zero at x_1 and x_5, and therefore the likelihood would be zero. That makes $\beta = 4.704$ the maximum likelihood estimate for β.

Problem 4.

The file linked below contains a data set of 150 samples. The first column contains a single continuous feature, X, assumed to have been drawn from an unknown probability density. The second column contains the binary class label Y.

`https://f000.backblazeb2.com/file/jeldridge-data/011-univariate_density_estimation/data.csv`

In the first problem on this homework, you used a histogram estimator to apply the Bayes classification rule. In this problem, you will instead estimate the class-conditional densities by fitting Gaussians using the method of maximum likelihood. In each part, show your code and provide your reasoning.

Note: you should *not* use `sklearn`, `scipy`, or any other library to fit the Gaussian densities. You should instead calculate the maximum likelihood estimates directly (using `numpy` or `pandas` to do this is fine).

a) Estimate the class-conditional densities $p_X(x \mid Y = 0)$ and $p_X(x \mid Y = 1)$ with Gaussians using the method of maximum likelihood and report the estimated parameters.

Solution: To estimate the class-conditional densities, we separate the data with label 0 and the data with label 1 and estimate the parameters μ and σ separately for each subset using the MLE formulas. The MLE for the mean is the sample mean, and the MLE for the variance is the sample variance. The code that does this is:

```
data = np.loadtxt('data.csv', delimiter=',')
y = data[:, 1]
X_1 = data[y == 1, 0]
X_0 = data[y == 0, 0]

# "fit" a Gaussian distribution to the data
mu_1 = np.mean(X_1)
sigma_1 = np.std(X_1)

mu_0 = np.mean(X_0)
sigma_0 = np.std(X_0)
```

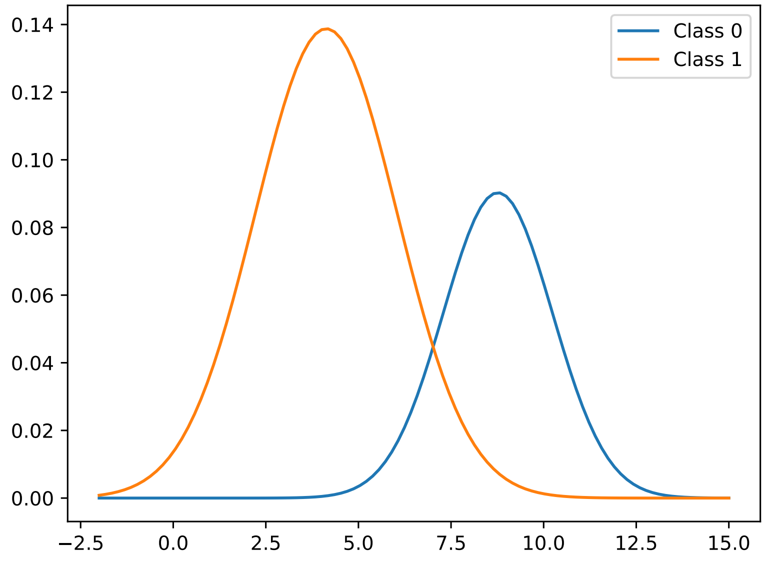
The parameter estimates we get are:

Parameter	Value
μ_1	4.13
σ_1	1.91
μ_0	8.76
σ_0	1.47

b) Let $\tilde{p}_X(x \mid Y = 0)$ and $\tilde{p}_X(x \mid Y = 1)$ be the estimated class-conditional densities from the previous part, and let $\tilde{\mathbb{P}}(Y = 1)$ be the estimate for $\mathbb{P}(Y = 1)$.

Plot $\tilde{p}_X(x \mid Y = 0) \cdot \tilde{\mathbb{P}}(Y = 0)$ and $\tilde{p}_X(x \mid Y = 1) \cdot \tilde{\mathbb{P}}(Y = 1)$ on the same axis. Label your plot so that the grader can tell which Gaussian corresponds to which class. Remember to show your code.

Solution:



```
def make_gaussian(mu, sigma):
    """Returns a function that evaluates a Gaussian density."""
    def f(x):
        return 1/np.sqrt(2 * np.pi * sigma**2) * np.exp(-(x - mu)**2 / (2 * sigma**2))
    return f
```

```python
g1 = make_gaussian(mu_1, sigma_1)
g0 = make_gaussian(mu_0, sigma_0)

# make functions which compute p(x|Y) P(Y)

def s1(x):
    return g1(x) * np.mean(y)

def s0(x):
    return g0(x) * (1 - np.mean(y))

xx = np.linspace(-2, 15, 100)

plt.plot(xx, g0(xx), label='Class 0')
plt.plot(xx, g1(xx), label='Class 1')
plt.legend()

plt.savefig('../include-solution/part_b.pdf')
```

c) Suppose a new point is observed at $x = 6.271$. What class does the Bayes classifier predict for x when the estimated densities and probabilities are used? Remember to show your code, and determine the predicted class through calculation (and not by visual inspection of the plot from the previous part).

Solution: To determine the prediction of the Bayes classifier, we calculate:

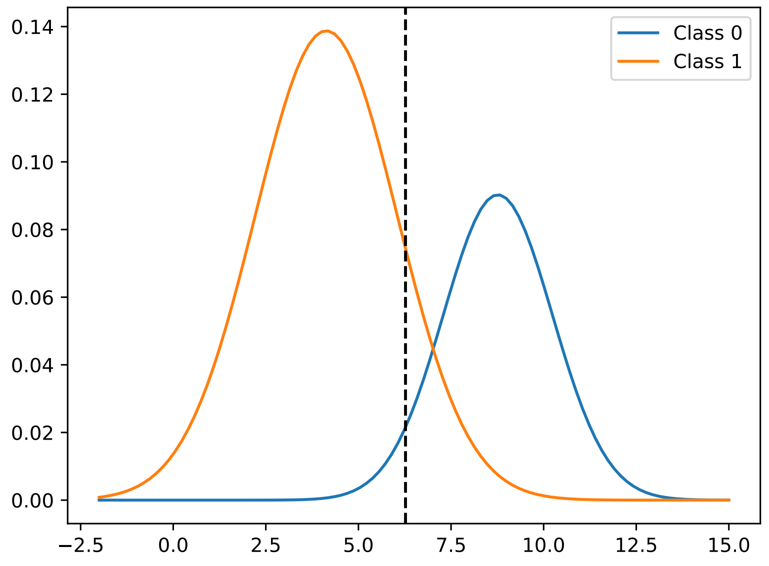
$$p(x \mid Y = 0) \cdot \mathbb{P}(Y = 0) \quad \text{and} \quad p(x \mid Y = 1) \cdot \mathbb{P}(Y = 1)$$

In our code above, we already created functions `s0` and `s1` that calculate these values. We can use these functions to calculate the predictions for $x = 6.271$:

```python
print(s0(6.271))
print(s1(6.271))
```

For Class 0, we get 0.022, while for Class 1 we get 0.074. Thus, the Bayes classifier predicts that $x = 6.271$ belongs to Class 1.

Note that we could have guessed this from the plot from part (b). If we draw a dashed line at $x = 6.271$, we can see that $p(6.271 \mid Y = 1) \cdot \mathbb{P}(Y = 1)$ is larger than $p(6.271 \mid Y = 0) \cdot \mathbb{P}(Y = 0)$:



d) The `scipy.optimize.fsolve` function can be used to find the *roots* of a function f; that is, the places where function $f(x) = 0$. Use `fsolve` to find the decision boundary for the classifier you trained above. Show your code.

Note that there may actually be multiple decision boundaries at the extremes, but there is one clear decision boundary in the middle of the data, as should be evident in your plot; find that one.

Solution: We use `scipy.optimize.fsolve` to find the decision boundary. It searches for the *roots* of a function, f. To use it to find where the decision boundary is, we define the function f:

$$f(x) = p(x \mid Y = 1) \cdot \mathbb{P}(Y = 1) - p(x \mid Y = 0) \cdot \mathbb{P}(Y = 0)$$

This will be zero precisely when $p(x \mid Y = 1) \cdot \mathbb{P}(Y = 1) = p(x \mid Y = 0) \cdot \mathbb{P}(Y = 0)$, which is exactly the decision boundary.

In code:

```python
import scipy.optimize

def f(x):
    return s1(x) - s0(x)

scipy.optimize.fsolve(f, x0=5)
```

This gives 7.01. Referencing our plots above, this looks to be where the orange and blue curves intersect, and is therefore reasonable.