DSC 140A - Homework 07

Due: Wednesday, February 28

Write your solutions to the following problems by either typing them up or handwriting them on another piece of paper. Unless otherwise noted by the problem's instructions, show your work or provide some justification for your answer. Homeworks are due via Gradescope at 11:59 PM.

Problem 1.

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: The get the σ_{MLE}, we need to maximize the log-likelihood $\tilde{L}(\sigma | x_1, \ldots, x_n)$ with respect to σ by taking the derivative and setting it to zero. We have:

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

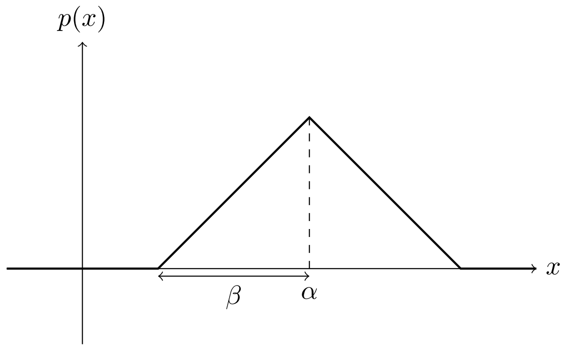
Setting this to zero, we have:

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

Which is the MLE of σ.

Problem 2.

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: The formula for the area of a triangle is $\frac{1}{2} \times \text{base} \times \text{height}$. And the $p(\alpha)$ is equivalent to the height of the triangle, and all pdfs should have an area of 1. So this gives us,

$$\begin{aligned} \frac{1}{2} \times 2\beta \times p(\alpha) = 1 \\ = \beta \times p(\alpha) = 1 \\ \implies p(\alpha) = \frac{1}{\beta} \end{aligned}$$

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

Solution: Looking at the graph, we can see that the density function is a linear function of x with a slope of $\frac{1}{\beta}$ and an intercept of $p(\alpha)$. So first we get the slope of the line,

$$\frac{p(\alpha) - 0}{\alpha - (\alpha - \beta)} = \frac{p(\alpha)}{\beta} = \frac{1}{\beta^2}$$

And the intercept is $p(\alpha)$. So the density function is,

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

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?

Solution: Since the data is symmetric, we can get an estimate for β easily by using the points 2 and 8. As that would give us the maximum width of the triangle. In addition, we also know that the width of the base of the triangle is 2β. So we can get the estimate for β by,

$$\begin{aligned} 2\beta &= 8 - 2 \\ \implies \beta &= 3 \end{aligned}$$

Hence, the maximum likelihood estimate for β is 3.

Problem 3.

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 last homework, you used a histogram estimator to apply the Bayes classification rule. This week, 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: Using the method of maximum likelihood, we can estimate the class-conditional, by getting the mean and variance of the data for each class. The mean and variance for each class is given by

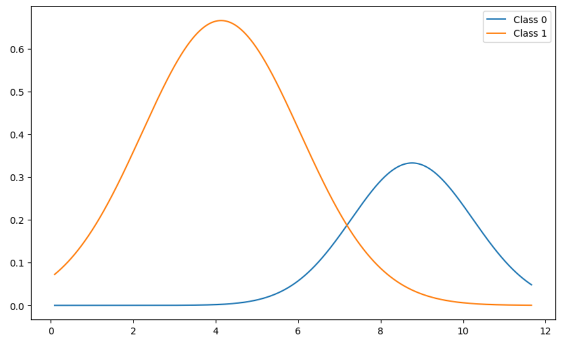
$$\text{Class 0: } \mu_0 = 8.764004,\ \sigma_0^2 = 2.1693085$$
$$\text{Class 1: } \mu_1 = 4.131046,\ \sigma_1^2 = 3.6730895124839993$$

All Code is below.

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: The plot of the class-conditional pdf using the estimated parameters,



All code is below.

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: Using the estimated class-conditional densities and the estimated class probabilities, we can calculate the posterior probabilities for each class, and the class with the higher posterior probability is the predicted class. The posterior probabilities are given by

$$p(Y = 0 \mid X = 6.271) = 0.07957047$$
$$p(Y = 1 \mid X = 6.271) = 0.35742232$$

Thus, using the Bayes classifier, the predicted class for $x = 6.271$ is 1.

All code is below.

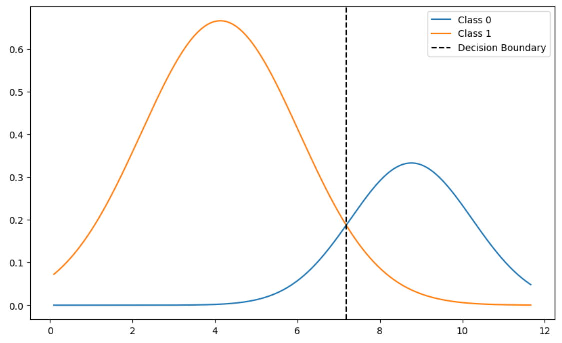
d) The `scipy.optimize.fsolve` function can be used to find the *roots* of a function f; that is, the places where function $f(x) = 0$. Use `fsolve` to find the decision boundary for the classifier you trained above. Show your code.

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: Finally, using fsolve from scipy.optimize, we can find the decision boundary for the classifier. The decision boundary is given by solving the solving the function $p(Y = 0 \mid X = x)p(Y = 0) - p(Y = 1 \mid X = x)p(Y = 1) = 0$. The decision boundary is given by

$$x_{\text{boundary}} = 7.1838603191$$

The plot is shown below,



All code is below.

Code

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

data = np.loadtxt('data.csv', delimiter=',')
X = data[:, :-1]
y = data[:, -1]

def Gaussian(x, mu, sigma):
    return np.exp(-np.linalg.norm(x - mu)**2 / (2 * sigma**2))

# get the mean and var when class is 0
```

```
X0 = X[y == 0]
mu0 = np.mean(X0, axis=0)
sigma0 = np.std(X0, axis=0)

# get the mean and var when class is 1
X1 = X[y == 1]
mu1 = np.mean(X1, axis=0)
sigma1 = np.std(X1, axis=0)

# a
print(f"The mean of class 0 is {mu0[0]} and the variance is {sigma0[0]**2}")
print(f"The mean of class 1 is {mu1[0]} and the variance is {sigma1[0]**2}")

# Calculate the probability of each class
p0 = len(X0) / len(X)
p1 = len(X1) / len(X)

# Generate random x values to plot the pdfs
x = np.linspace(X.min(), X.max(), 1000)

# Calculate estimate pdfs for each class
pdf0 = np.array([Gaussian(x[i], mu0, sigma0) for i in range(len(x))])
pdf1 = np.array([Gaussian(x[i], mu1, sigma1) for i in range(len(x))])

# Scale the pdfs by the probability of each class
scaled_pdf0 = pdf0 * p0
scaled_pdf1 = pdf1 * p1


# b
# Plot the scaled pdfs
plt.figure(figsize=(10, 6))
plt.plot(x, scaled_pdf0, label='Class 0')
plt.plot(x, scaled_pdf1, label='Class 1')
plt.legend()
plt.show()



# c
x_new = 6.271

# Calculate the probability of each class given the new x value
p0_new = Gaussian(x_new, mu0, sigma0) * p0
p1_new = Gaussian(x_new, mu1, sigma1) * p1

print(f"The probability of class 0 given x_new is {p0_new}")
print(f"The probability of class 1 given x_new is {p1_new}")

# Using the bayes classifier to predict the class of the new x value
if p0_new > p1_new:
    print(f"The class of x_new is 0")
else:
    print(f"The class of x_new is 1")
```

```python
from scipy.optimize import fsolve

def f(x):
    return Gaussian(x, mu0, sigma0) * p0 - Gaussian(x, mu1, sigma1) * p1

x_star = fsolve(f, 7)
# d
print(f"The value of x_star is {x_star[0]} which is the decision boundary")

#d
# Plot the scaled pdfs with the decision boundary
plt.figure(figsize=(10, 6))
plt.plot(x, scaled_pdf0, label='Class 0')
plt.plot(x, scaled_pdf1, label='Class 1')
plt.axvline(x_star, color='black', linestyle='--', label='Decision Boundary')
plt.legend()
plt.show()
```