DSC 140A - Homework 07

Due: Wednesday, March 1

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: 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 2.

Let X be a continuous random variable, and let Y be a random class label (1 or 0).

Suppose $p_X(x \mid Y = 1)$ is Gaussian with $\mu = 2$ and $\sigma = 3$ and that $p_X(x \mid Y = 0)$ is also Gaussian with $\mu = 5$ and $\sigma = 3$. Suppose, too, that $\mathbb{P}(Y = 1) = \mathbb{P}(Y = 0) = \frac{1}{2}$.

Recall that the Bayes error is the probability that the Bayes classifier makes an incorrect prediction. What is the Bayes error for this distribution? Show your work.

Hint: you'll want some way to compute the area under a Gaussian. You can use the tables that appear in the back of your statistics book, or you can use something like `scipy.stats.norm.cdf`. We'll let you read the documentation to see how to use it, but it may be helpful to remember that if F is the cumulative density function for a distribution with density f, then $\int_a^b f(x)\,dx = F(b) - F(a)$.

Solution: Because the Gaussians have the same width and the classes are equally probable, the decision boundary is exactly halfway between their means, at $x = 3.5$. Any point to the left of 3.5 is predicted Class 0, and everything to the right is predicted Class 1.

The Bayes error is the probability that a point is misclassified. This can occur in two ways: either the point came from class $Y = 1$, but the prediction is for class 0, or the point came from class $Y = 0$ but was predicted to be class 1. The total probability of an error is the sum of the probabilities of either case occurring.

Consider the first case, where the point came from class $Y = 1$ but is predicted to be Class 0. This will occur when the point is to the left of 3.5. What is the probability that a point comes from $Y = 1$ and is to the left of 3.5? It is:

$$\begin{aligned}\mathbb{P}(\text{x} < 3.5 \text{ and Y} = 1) &= \mathbb{P}(\text{x} < 3.5 \mid Y = 1)\mathbb{P}(Y = 1)\\ &= \mathbb{P}(\text{x} < 3.5 \mid Y = 1) \times 0.5\end{aligned}$$

The probability that $x < 3.5$ given that it comes from Class 1 is computed as the area under the curve of the Gaussian for Class 1's distribution from $-\infty$ to 3.5. This can be computed with

```
scipy.stats.norm.cdf(3.5, 5, 3)
```

the result is 0.308. Therefore:

$$\begin{aligned}\mathbb{P}(\text{x} < 3.5 \text{ and Y} = 1) &= \mathbb{P}(\text{x} < 3.5 \mid Y = 1)\mathbb{P}(Y = 1)\\ &= 0.308 \times 0.5\\ &= 0.154\end{aligned}$$

Likewise, the probability of the second case is:

$$\begin{aligned}\mathbb{P}(\text{x} > 3.5 \text{ and Y} = 0) &= \mathbb{P}(\text{x} > 3.5 \mid Y = 0)\mathbb{P}(Y = 0)\\ &= \mathbb{P}(\text{x} > 3.5 \mid Y = 0) \times 0.5\end{aligned}$$

The probability that $x > 3.5$ when drawn from the Gaussian with mean $\mu = 2$ is:

$1 -$ `scipy.stats.norm.cdf(3.5, 2, 3)`

This is also 0.308, which could have been recognized from symmetry.

Therefore:

$$\begin{aligned}\mathbb{P}(\text{x} > 3.5 \text{ and Y} = 0) &= \mathbb{P}(\text{x} > 3.5 \mid Y = 0)\mathbb{P}(Y = 0)\\ &= .308 \times 0.5\end{aligned}$$

All together, then, the Bayes error is 0.308.

Problem 3.

The file at the link below contains a data set of 100 points from two classes (1 and -1).

`https://f000.backblazeb2.com/file/jeldridge-data/003-two_clusters/data.csv`

The first two columns contains features, and the last column contains the label of the point. Note that the labels are 1 and -1, not 1 and 0, and that there are no column headers.

In all parts of this problem you may use code to compute your answers. If you do, be sure to include your code.

a) Suppose two Gaussians with full covariance matrices are used to model the densities $p_X(x \mid Y = 1)$ and $p_X(x \mid Y = -1)$. What are the maximum likelihood estimates for the covariance matrices of each Gaussian?

(Allow each Gaussian to have its own covariance matrix; don't use the same covariance for both.)

Hint: the covariance matrix for the Gaussian fit to points from class 1 should have 12.29 in its top-left entry.

Solution: If you'd like to compute the covariances manually:

```
data = np.loadtxt('data.csv', delimiter=',')
X = data[:,:2]
y = data[:,-1]

X_1 = X[y == 1]
X_0 = X[y == -1]

y_1 = y[y == 1]
y_0 = y[y == -1] * 0

mu_1 = X_1.mean(axis=0)
Z_1 = (X_1 - mu_1)
n_1 = len(Z_1)

mu_0 = X_0.mean(axis=0)
Z_0 = (X_0 - mu_0)
n_0 = len(Z_0)

C_1 = 1 / n_1 * Z_1.T @ Z_1
C_0 = 1 / n_0 * Z_0.T @ Z_0

>>> C_1
array([[12.29584016,  0.28098224],
       [ 0.28098224, 16.06766736]])
>>> C_0
array([[10.91736224,  0.53015728],
       [ 0.53015728, 15.17320916]])
```

You could also have used numpy. If you did, you had to make sure to get to pass it an array whose *columns* are data points, instead of rows, and to use `bias = True`:

```
np.cov(X_1.T, bias=True)
```

b) Using the estimated Gaussians with the Bayes classification rule, what are the predicted labels of each of the following points?

- $(0, 0)^T$
- $(1, 1)^T$
- $(10, 5)^T$
- $(5, -5)^T$
- $(8, 5)^T$

Show your calculations.

Note: making predictions in this way (using Gaussians with unequal covariance matrices) is known as *Quadratic Discriminant Analysis*.

Solution: The predicted labels are -1, -1, 1, -1, -1.

The code below defines a function for making the predictions:

```
mvn = scipy.stats.multivariate_normal

py_1 = n_1 / (n_1 + n_0)
py_0 = n_0 / (n_1 + n_0)

def predict(x):
    if mvn.pdf(x, mean=mu_1, cov=C_1) * py_1 > mvn.pdf(x, mean=mu_0, cov=C_0) * py_0:
        return 1
    else:
        return -1
```