DSC 140A - Homework 06

Due: Wednesday, May 15

Instructions: Write your solutions to the following problems either by typing them or handwriting them on another piece of paper. Show your work or provide justification unless otherwise noted. If you write code to solve a problem, include the code by copy/pasting or as a screenshot. You may use `numpy`, `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.

Problem 1.

In this problem, you will demonstrate that kernel ridge regression is equivalent to ridge regression performed in feature space by showing that they make the same predictions on a concrete data set.

We'll use the toy data set:

i	$\vec{x}^{(i)}$	y_i
1	$(0, 2)^T$	1
2	$(1, 0)^T$	1
3	$(0, -2)^T$	1
4	$(-1, 0)^T$	1
5	$(0, 0)^T$	-1

We will define

$$\vec{\phi}(\vec{x}) = (1, x_1^2, x_2^2, \sqrt{2}\,x_1, \sqrt{2}\,x_2, \sqrt{2}\,x_1 x_2)^T.$$

It turns out that $\kappa(\vec{x}, \vec{x}') = (1 + \vec{x} \cdot \vec{x}')^2$ is a kernel for $\vec{\phi}$.

You can write code to perform any and all calculations in this problem. If you do, please show your code.

a) Learn a prediction rule $H_1(\vec{x})$ by performing ridge regression in the 6-dimensional feature space and report the optimal parameter vector $\vec{w}$. Use a regularization parameter of $\lambda = 2$.

Solution:

To perform ridge regression in feature space, we compute $\vec{w}^* = (\Phi^T\Phi + n\lambda I)^{-1}\Phi^T\vec{y}$, where Φ is the design matrix whose ith row is $\vec{\phi}(\vec{x})$.

The below code does exactly that:

```
import numpy as np

def phi(x):
    x_1, x_2 = x
    c = np.sqrt(2)
    return np.array([
        1,
        x_1**2,
        x_2**2,
        c * x_1,
        c * x_2,
        c * x_1 * x_2
    ])
```

```
X = np.array([
    [0, 2],
    [1, 0],
    [0, -2],
    [-1, 0],
    [0, 0]
])

y = np.array([1, 1, 1, 1, -1])

n = len(X)

ell = 2
Phi = np.array([phi(x) for x in X])
w = np.linalg.inv((Phi.T @ Phi + n * ell * np.eye(Phi.shape[1]))) @ Phi.T @ y
```

We find:

$$\vec{w} \approx (0.086, 0.152, 0.174, 0, 0, 0)^T$$

b) Given a new point, $\vec{x} = (1, 1)^T$, what is the prediction made by your ridge regressor? That is, what is $H_1(\vec{x})$? Show your calculations / code.

Solution: To predict $H_1(\vec{x})$, we need to map $\vec{x}$ to feature space with $\vec{\phi}(\vec{x})$ and dot it with $\vec{w}$. That is, we need to compute $\vec{p}hi(\vec{x}, \vec{w})$.

To do so in code, we first define `x = np.array([1, 1])`, and then compute `phi(x) @ w`. Our answer is approximately 0.413.

c) Compute the kernel matrix, K, for this data.

Solution: We define a python function for computing the kernel and apply it to every pair of training points:

```
def k(x, z):
    return (1 + x @ z)**2

K = np.zeros((n, n))
for i in range(n):
    for j in range(i, n):
        K[i,j] = K[j,i] = k(X[i], X[j])his is done with the code

>>> K
array([[25.,  1.,  9.,  1.,  1.],
       [ 1.,  4.,  1.,  0.,  1.],
       [ 9.,  1., 25.,  1.,  1.],
       [ 1.,  0.,  1.,  4.,  1.],
       [ 1.,  1.,  1.,  1.,  1.]])
```

d) Learn a prediction function $H_2(\vec{x})$ by solving the kernel ridge regression dual problem; that is, by finding the optimal vector $\vec{\alpha}$. Recall that there is an exact solution: $\vec{\alpha} = (K + n\lambda I)^{-1}\vec{y}$. Report the $\vec{\alpha}$ that you find.

Note: the lecture slides had originally stated the solution without the n in $n\lambda I$. This was a typo and has

been fixed.

Solution: `alpha = np.linalg.inv(K + n * ell * np.eye(n)) @ y`

We find $\vec{\alpha} \approx (0.022, 0.077, 0.022, 0.077, -0.11)^T$

e) Let $\vec{x} = (1, 1)^T$, as before. What does your kernel ridge regressor predict for this point? That is, what is $H_2(\vec{x})$?

Hint: $H_2(\vec{x})$ should be the same as $H_1(\vec{x})$.

Solution: To compute $H_2(\vec{x})$, we use the dual solution $\vec{\alpha}$. From lecture, we have:

$$H_2(\vec{x}) = \sum_{i=1}^{n} \alpha_i k(\vec{x}^{(i)}, x)$$

In code: `[k(x_i, x) for x_i in X] @ alpha`

We get 0.41, which is the same as $H_1(\vec{x})$, as expected.

Problem 2.

Let X be a continuous random variable, and let Y be a binary random variable. Suppose the class conditional densities are know to be:

$$p_X(x \mid Y = 0) = \begin{cases} 1/5, & \text{if } 0 \leq x \leq 2 \\ 1/3, & \text{if } 2 < x \leq 3 \\ 1/15, & \text{if } 3 < x \leq 7 \end{cases}$$

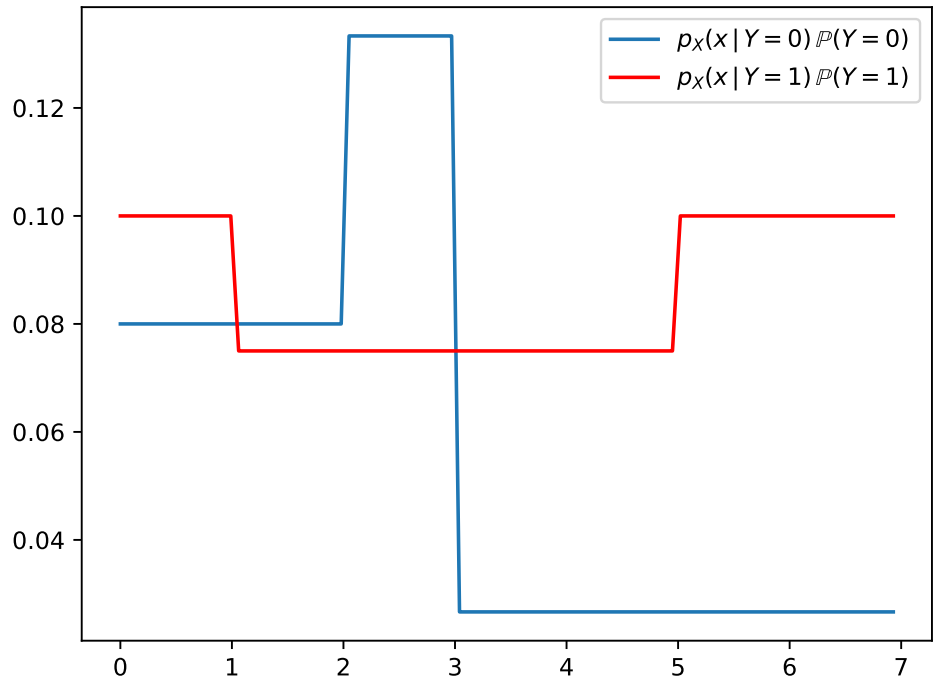
$$p_X(x \mid Y = 1) = \begin{cases} 1/6, & \text{if } 0 \leq x \leq 1 \\ 1/8, & \text{if } 1 < x \leq 5 \\ 1/6, & \text{if } 5 < x \leq 7 \end{cases}$$

Suppose also that $\mathbb{P}(Y = 0) = 0.4$ and $\mathbb{P}(Y = 1) = 0.6$.

For what values of $x \in [0, 7]$ will the Bayes classifier predict $y = 1$?

Solution: The Bayes classifier will predict $y = 1$ for all $x \in [0, 1] \cup (3, 7]$.

The Bayes classifier tells us to predict whichever is larger: $p_X(x \mid Y = 1)\mathbb{P}(Y = 1)$ or $p_X(x \mid Y = 0)\mathbb{P}(Y = 0)$. To determine this, we can draw both plots:



We see that the red curve is larger than the blue curve on the intervals $[0, 1]$ and $(3, 7]$, and this is exactly where the Bayes classifier will predict $y = 1$.

Problem 3.

Let X be a continuous random variable, and let Y be a random class label (1 or 0). Recall that the Gaussian probability density function (pdf) is given by:

$$\mathcal{N}(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Suppose $p_X(x \mid Y = 1)$ is a Gaussian pdf 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:

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

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:

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

Likewise, the probability of the second case is:

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

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:

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

All together, then, the Bayes error is 0.308.