Machine Learning Homework 1

Ian Sinclair
ENCE 3631-1

March 29, 2022
Dr. Zhihui Zhu

Problem 1:

Suppose that we have some number m of coins. Each coin has the same probability of landing on heads, denoted p. Suppose that we pick up each of the m coins in turn and for each coin do n independent coin tosses. Note that the probability of obtaining exactly k heads out of n tosses for any given coin is given by the binomial distribution:

$$\mathbb{P}[k|n,p] = \binom{n}{k} p^k (1-p)^{n-k}$$

For each series of coin tosses, we will record the results via the empirical estimates given by

$$\hat{p}_i = \frac{\text{number of times coin } i \text{ lands on heads}}{n}$$

A) Assume that $n = 10$. If all the coins have $p = 0.05$, compute a formula for the exact probability that at least one coin will have $\hat{p}_i = 0$ (This may be easier to calculate by instead computing the probability that this does not occur.) Give a table containing the values of this probability for the cases of $m = 1$, $m = 1000$, and $m = 1000000$.

Consider we desire the probability that at least one trial out of m trials will have no coin flips that are heads. Thus, we can also approach the statement by its equivalent negational probability,

$$\mathbb{P}[(\hat{p}_i = 0) \text{ for at least 1 of } m \text{ trials}] = 1 - \mathbb{P}[\text{All of } m \text{ trials have at least 1 heads}].$$

Now the probability that any particular trial has at least 1 heads is equivalent to the negational probability that a trial has no heads or only tails.

$$\mathbb{P}[\hat{p}_i \neq 0] = \mathbb{P}[\text{There is at least one flip that is heads}] = 1 - \mathbb{P}[\text{All of } n \text{ flips are tails}]$$

Now, because each coin flip is independent,

$$1 - \mathbb{P}[\text{All of } n \text{ flips are tails}] = 1 - (1-p)^n = 1 - (0.95)^{10}$$

NOTE: This result can also be evaluated using the binomial distribution for 0 heads over n tosses,

$$1 - \mathbb{P}[\text{All of } n \text{ flips are tails}] = 1 - \binom{n}{0} p^0 (1-p)^{n-0} = 1 - (1-p)^n = 1 - (0.95)^{10}$$

Now, because each of the m trials are independent, repeating these flips for each coins reveals,

$$\boxed{\mathbb{P}[(\hat{p}_i = 0) \text{ for at least 1 of } m \text{ trials}] = 1 - (1-(1-p)^n)^m = 1 - \left(1 - (0.95)^{10}\right)^m.}$$

m	1	100	1000000
$\mathbb{P}[\hat{p}_i = 0]$	0.5987	$\approx 1^-$	$\approx 1^-$

B) Now assume that $n = 100$, $m = 2$, and that $p = 0.5$ for both coins. Apply the Hoeffding inequality together with the union bound and then plot/sketch (with X-axis being ϵ) for

$$\mathbb{P}\Big[\max_i |\hat{p}_i - p| > \epsilon\Big]$$

Note: First, by convention, allow $\epsilon > 0$, then take $\max_i |\hat{p}_i - p|$ to be the 'riskiest' (worst case) empirical estimate $\hat{p}_i$ for all of i. Now, because this is effectively a variable we cannot apply Hoeffding's inequality directly.
Therefore, instead consider for a particular empirical estimate $\hat{p}_j$, for fixed j, Hoeffding's inequality implies,

$$\mathbb{P}\Big[|\hat{p}_j - p| > \epsilon\Big] \leq 2^{-2\epsilon^2 n}$$

For n number of coin flips in trial j.
Fortunately, using union bounds for both coins reveals,

$$\mathbb{P}\Big[\max_i |\hat{p}_i - p| > \epsilon\Big] \leq \mathbb{P}\Big[|\hat{p}_1 - p| > \epsilon \cup |\hat{p}_2 - p| > \epsilon\Big] \leq \mathbb{P}\Big[|\hat{p}_1 - p| > \epsilon\Big] + \mathbb{P}\Big[|\hat{p}_2 - p| > \epsilon\Big]$$

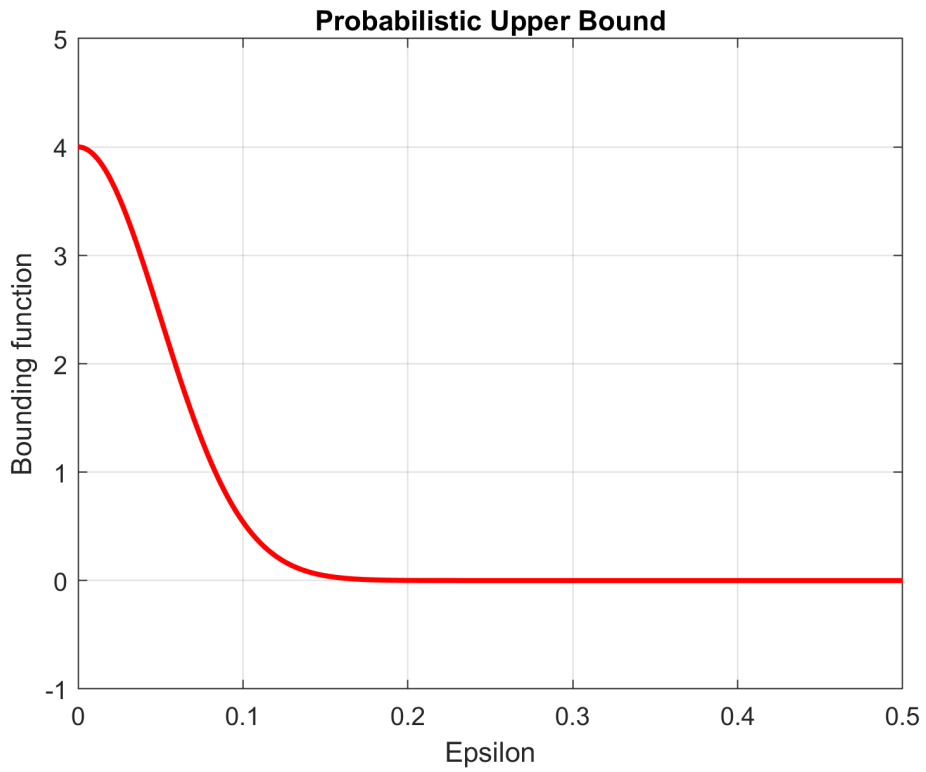
Now because n is consistent between trials, we can apply Hoeffding's inequality to each term,

$$\mathbb{P}\Big[|\hat{p}_1 - p| > \epsilon\Big] + \mathbb{P}\Big[|\hat{p}_2 - p| > \epsilon\Big] \leq 2e^{-2\epsilon^2 n} + 2e^{-2\epsilon^2 n} = 4e^{-2\epsilon^2 n}.$$

And so, taking $n = 100$, we get an upper bound for the desired probability,

$$\mathbb{P}\Big[\max_i |\hat{p}_i - p| > \epsilon\Big] \leq 4e^{-200\epsilon^2}.$$

Which results in the normal distribution,



Problem 2:

Consider a binary classification problem involving a single (scalar) feature x and suppose that $X|Y=0$ and $X|Y=1$ are continuous random variables with densities given by,

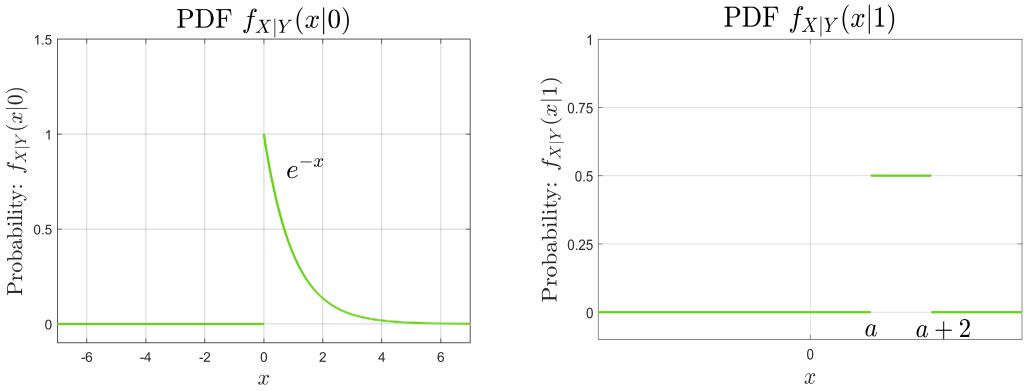
$$f_{X|Y}(x|0) = \begin{cases} e^{-x}, & x \geq 0, \\ 0, & otherwise. \end{cases}$$

$$f_{X|Y}(x|1) = \begin{cases} 1/2, & x \in [a, a+2], \\ 0, & otherwise. \end{cases}$$

respectively, where $a \geq 0$ is given. Furthermore, suppose the $\mathbb{P}[Y=0] = \mathbb{P}[Y=1] = \frac{1}{2}$.

A) Sketch $f_{X|Y}(x|0)$ and $f_{X|Y}(x|1)$.

Note: Take k as a particular discrete classification in the Y space, $k \in \{0,1\}$. Then $f_{X|Y}(x|k)$ is a probability density function (pdf) related to k. Which implies the $\int_{-\infty}^{\infty} f_{X|Y}(x|k)dx = 1$. And so, it is possible to define the probability distribution for each discrete value of k, by the following plots.



B) Suppose $X = 1/2$; what is the mostly likely value for Y according to the Bayes rule? What about $X = 1$? (Hint: the solution may depend on the value a).

Again let k be a particular classification in Y, st. $k \in \{0,1\}$. Then by Bayes Theorem, the conditional probability that $Y = k$, given $X = x$ for some continuous feature x is,

$$P_{Y|X}(k|x) = P[Y=k|X=x] = \frac{P[Y=k]f_{X|Y}(x|k)}{f_X(x)}.$$

i) $X = \frac{1}{2}$

Additionally, $f_X(x)$ is not dependent on k so it is not relevant to probability that x is assigned to k. And so given, $X = \frac{1}{2}$, we get the probability,

$$P_{Y|X}(k|\frac{1}{2}) = P[Y=k|X=\frac{1}{2}] = \frac{P[Y=k]f_{X|Y}(\frac{1}{2}|k)}{f_X(\frac{1}{2})}.$$

Now, it becomes possible to define a MAP for each possible discrete value of k.
Starting with $k=0$, such that,

$$P_{Y|X}(0|\frac{1}{2}) = P[Y=0|X=\frac{1}{2}] = \frac{P[Y=0]f_{X|Y}(\frac{1}{2}|0)}{f_X(\frac{1}{2})}$$

Where, $P[Y=0] = \frac{1}{2}$ is given,
and,

$$f_{X|Y}(\frac{1}{2}|0) = e^{-\frac{1}{2}} = 0.6065.$$

Thus,

$$P_{Y|X}(0|\frac{1}{2}) = \frac{\frac{1}{2}(0.6065)}{f_X(x)} = \frac{0.3033}{f_X(x)}.$$

Additionally, the probability that x will be classified as 1, will be,

$$P_{Y|X}(1|\frac{1}{2}) = P[Y=1|X=\frac{1}{2}] = \frac{P[Y=1]f_{X|Y}(\frac{1}{2}|1)}{f_X(\frac{1}{2})}$$

Similarly, $P[Y=1] = \frac{1}{2}$ and by the pdf of $f_{X|Y}(x|1)$,

$$f_{X|Y}(\frac{1}{2}|1) = \begin{cases} \frac{1}{2}, & \frac{1}{2} \in [a, a+2], \\ 0 & \frac{1}{2} \notin [a, a+2]. \end{cases}$$

As a result,

$$P_{Y|X}(1|\frac{1}{2}) = \begin{cases} \frac{\frac{1}{4}}{f_X(x)}, & \frac{1}{2} \in [a, a+2], \\ 0, & \frac{1}{2} \notin [a, a+2]. \end{cases}$$

Finally, we consider a Bayesian classifier under the follow definition,

$$h^*(x) = \arg\max_k P_{Y|X}(k|x).$$

In which, by our MAP, it is possible to find this maximum.
Result, note, $P_{Y|X}(1|\frac{1}{2})$ depends on a; however, $P_{Y|X}(0|\frac{1}{2}) > P_{Y|X}(1|\frac{1}{2})$ for all values of a. Therefore,

$$\max_k P_{Y|X}(k|\frac{1}{2}) = P_{Y|X}(0|\frac{1}{2}) = \frac{0.3033}{f_X(x)}.$$

and,

$$\arg\left(P_{Y|X}(0|\frac{1}{2})\right) = 0.$$

Therefore, for $x = \frac{1}{2}$ the most likely classification of x is 0.

ii) $X = 1$

By a symmetric argument it is possible to find the most likely classification for $X = 1$. Therefore, we take the probability that x is assigned to a particular value k; by,

$$P_{Y|X}(k|1) = P[Y=k|X=1] = \frac{P[Y=k]f_{X|Y}(1|k)}{f_X(1)}.$$

and approaching a MAP of all discrete values of Y, by first taking $k = 0$.

$$P_{Y|X}(0|1) = P[Y=0|X=1] = \frac{P[Y=0]f_{X|Y}(1|0)}{f_X(1)}$$

Where $P[Y=0] = \frac{1}{2}$ and $f_{X|Y}(1|0)$ is given by the pdf,

$$f_{X|Y}(1|0) = e^{-1} = 0.3679$$

And so,

$$P_{Y|X}(0|1) = \frac{\frac{1}{2}(0.3678)}{f_X(x)} = \frac{0.1844}{f_X(x)}.$$

Next taking $k = 1$, note $P[Y=1] = \frac{1}{2}$, and $f_{X|Y}(1|1)$ is defined by the pdf,

$$f_{X|Y}(1|1) = \begin{cases} \frac{1}{2}, & 1 \in [a, a+2], \\ 0 & 1 \notin [a, a+2]. \end{cases}$$

Thus, the assignment probability,

$$P_{Y|X}(1|1) = \begin{cases} \frac{\frac{1}{4}}{f_X(x)}, & 1 \in [a, a+2], \\ 0, & 1 \notin [a, a+2]. \end{cases}$$

Now again by the Bayesian classifier,

$$h^*(x) = \arg\max_k P_{X|Y}(k|x)$$

it is possible to find the relationship between elements in the MAP by the location of a.
Such that, because $P_{X|Y}(0|1) > P_{X|Y}(1|1)$ for $1 \notin [a, a+2]$, and $P_{X|Y}(0|1) < P_{X|Y}(1|1)$ for $1 \in [a, a+2]$, we get the max probability model,

$$\max_k P_{X|Y}(k|1) = \begin{cases} P_{X|Y}(1|1), & 1 \in [a, a+2], \\ P_{X|Y}(0|1), & \textit{otherwise}. \end{cases} = \begin{cases} \frac{0.25}{f_X(x)}, & 1 \in [a, a+2], \\ \frac{0.1844}{f_X(x)}, & \textit{otherwise}. \end{cases}$$

Therefore, taking the argument of the peicewise function,

$$\arg\max_k P_{X|Y}(k|1) = \begin{cases} \arg P_{X|Y}(1|1), & 1 \in [a, a+2], \\ \arg P_{X|Y}(0|1), & \textit{otherwise}. \end{cases} = \begin{cases} 1, & 1 \in [a, a+2], \\ 0, & \textit{otherwise}. \end{cases}$$

And therefore, for $X = 1$, if $1 \in [a, a+2]$ the most likely value for Y is 1, otherwise, the most likely value is 0.

Problem 3:

In this problem I would like you to design a k-nearest neighbor classifier for several different values of n. In particular, I would like you to consider $n = 100, 500, 1000, 5000$. For each of these values of n, experiment with different choices of k and decide what the "best" choice of k is for each of these values of n (either based on the visual results, or using some quantitative method of your own devising). Provide a table showing your choices of k, and include a plot of the resulting classifier for each value of n.

Considered is a k-nearest neighbor classifier that classifies features based on the greatest mean number of alike classifications for its k closest neighbors. For instance, given a binary Y space, (red, green), if we allow $k = 5$ and the closest neighbors to a particular feature include 2 green points and 3 red points, then the feature will be classified as red.
To that effect, choosing the right value for k is critical to prevent the classifier from being influenced by to many or to few neighboring data points. And so we fix n random data points, and employ a visual method to find a value k that creates the best boundary.
Specifically, we are interested in a boundary that excludes outliers and preserves the general expected distribution of data points, (minimizes errors).
As an example, here are two classifiers with either to low or to high values of k.

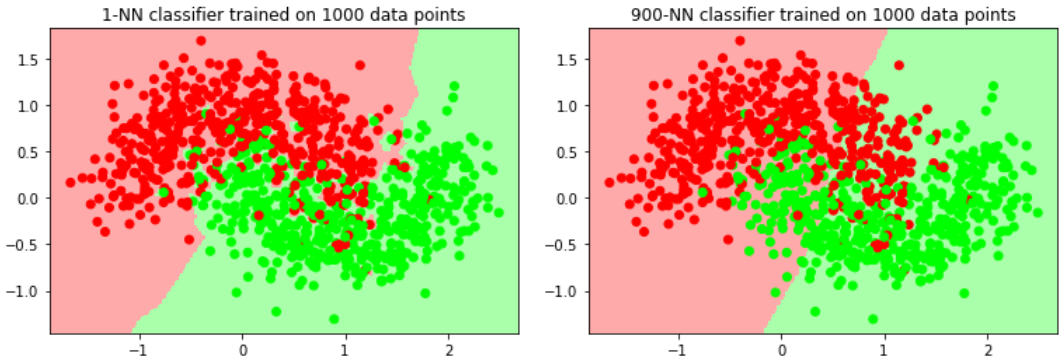


Figure 1: **Bad Examples** of choices for k.

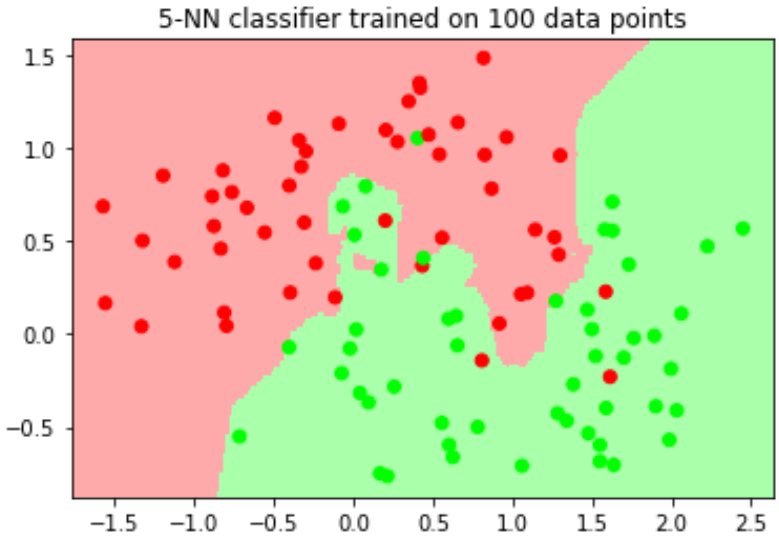
Both figures display poor choices for k; specifically, the figure on the left has many small boundary regions inside the section dominated by the opposing color, which makes the classifier less accurate. Additionally, the figure on the right mis-classifies many features because it is taking too large of a neighborhood.

Therefore, using a visual method we tweak the values of k to roughly optimize the classifier for $n = 100, 500, 1000,$ and 5000.

n	100	500	1000	5000
k	5	25	35	65

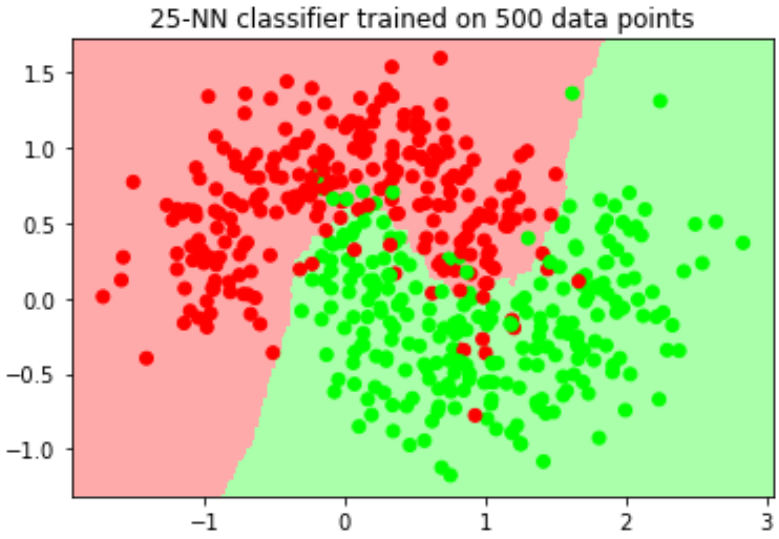
Additionally, note the plots for each classifier.

$$\boxed{n = 100 \qquad k = 5}$$



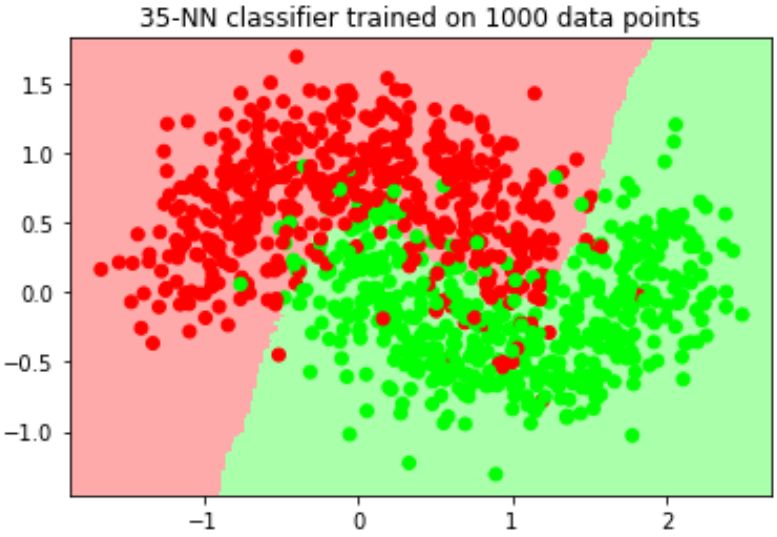
Note: This boundary attempts to fit all points in the correct color; however, there are still a few outliers and a few error around the boundary edge. Still, it correctly classifies most of the points and increasing the value for k will cause more errors at the boundary edge, while decreasing k will cause the regions to split. And so this value of k seems to be a logical compromise.

$$\boxed{n = 500 \qquad k = 25}$$



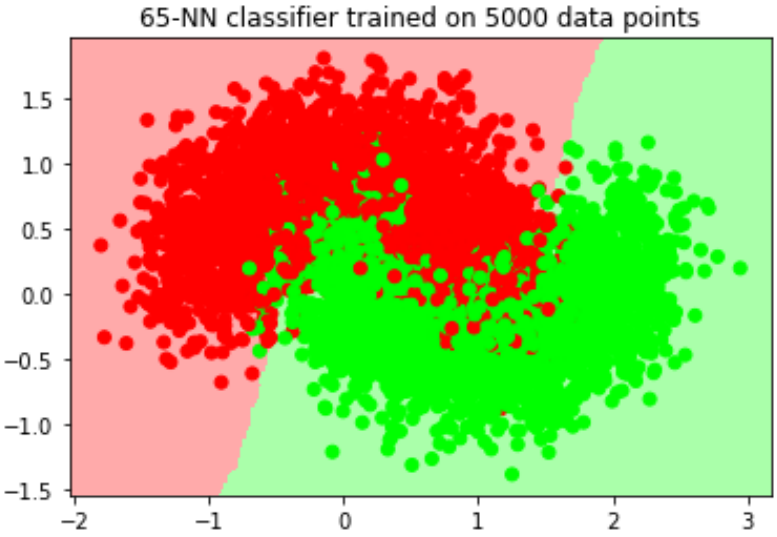
Note: Similarly to the $n = 100$ figure, this value for k is not perfect; however, the boundary edge seems to match the natural distribution of points. Therefore, $k = 25$ seems to be a logical choice.

$$\boxed{n = 1000 \qquad k = 35}$$



Note: This boundary edge is reasonably consistent with a visual approximation of the distribution of the data points.

$$\boxed{n = 5000 \qquad k = 65}$$



Note: Again, this boundary edge seems reasonably consistent with a visual approximation of the distribution of the data points.

Appendix

```
% Problem 1:
t = -0.5:0.001:0.5;
p = 4*exp(-200.*t.*t);
plot(t,p);
ylim([-1,5]);
title('Probabilistic Upper Bound');
xlabel('Epsilon');
ylabel('Bounding function');

% Problem 2:
t1 = 0:0.001:20;
t2 = -20:0.001:0;
t3 = -10:0.001:2;
t4 = 2:0.001:4;
t5 = 4:0.001:10;

f1 = exp(-t1);
f2 = 0 * ones(1, length(t2));
f3 = 0 * ones(1, length(t3));
f4 = 0.5 * ones(1, length(t4));
f5 = 0 * ones(1, length(t5));

hold on
plot(t3,f3);
plot(t4,f4);
plot(t5,f5);
title('PDF $$f_{X|Y}(x|1)$$','interpreter','latex');
xlabel('$$x$$','interpreter','latex');
ylabel('Probability: $$f_{X|Y}(x|1)$$','interpreter','latex');
ylim([-0.1,1]);
xlim([-7,7]);
```

```
## Problem 3:
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.colors import ListedColormap
from sklearn import neighbors, datasets

np.random.seed(2020) # Set random seed so results are repeatable

n = 5000 # number of training points
k = 65 # number of neighbors to consider

## Generate a simple 2D dataset
X, y = datasets.make_moons(n,'True',0.3)

## Create instance of KNN classifier
classifier = neighbors.KNeighborsClassifier(k,'uniform')
classifier.fit(X, y)

## Plot the decision boundary.
# Begin by creating the mesh [x_min, x_max]x[y_min, y_max].
h = .02  # step size in the mesh
x_delta = (X[:, 0].max() - X[:, 0].min())*0.05 # add 5% white space to border
y_delta = (X[:, 1].max() - X[:, 1].min())*0.05
x_min, x_max = X[:, 0].min() - x_delta, X[:, 0].max() + x_delta
y_min, y_max = X[:, 1].min() - y_delta, X[:, 1].max() + y_delta
```

```
xx, yy = np.meshgrid(np.arange(x_min, x_max, h), np.arange(y_min, y_max, h))
Z = classifier.predict(np.c_[xx.ravel(), yy.ravel()])

# Create color maps
cmap_light = ListedColormap(['#FFAAAA', '#AAFFAA'])
cmap_bold = ListedColormap(['#FF0000', '#00FF00'])

# Put the result into a color plot
Z = Z.reshape(xx.shape)
plt.figure()
plt.pcolormesh(xx, yy, Z, cmap=cmap_light)

## Plot the training points
plt.scatter(X[:, 0], X[:, 1], c=y, cmap=cmap_bold)
plt.xlim(xx.min(), xx.max())
plt.ylim(yy.min(), yy.max())
plt.title("%i-NN classifier trained on %i data points" % (k,n))

## Show the plot
plt.show()
```