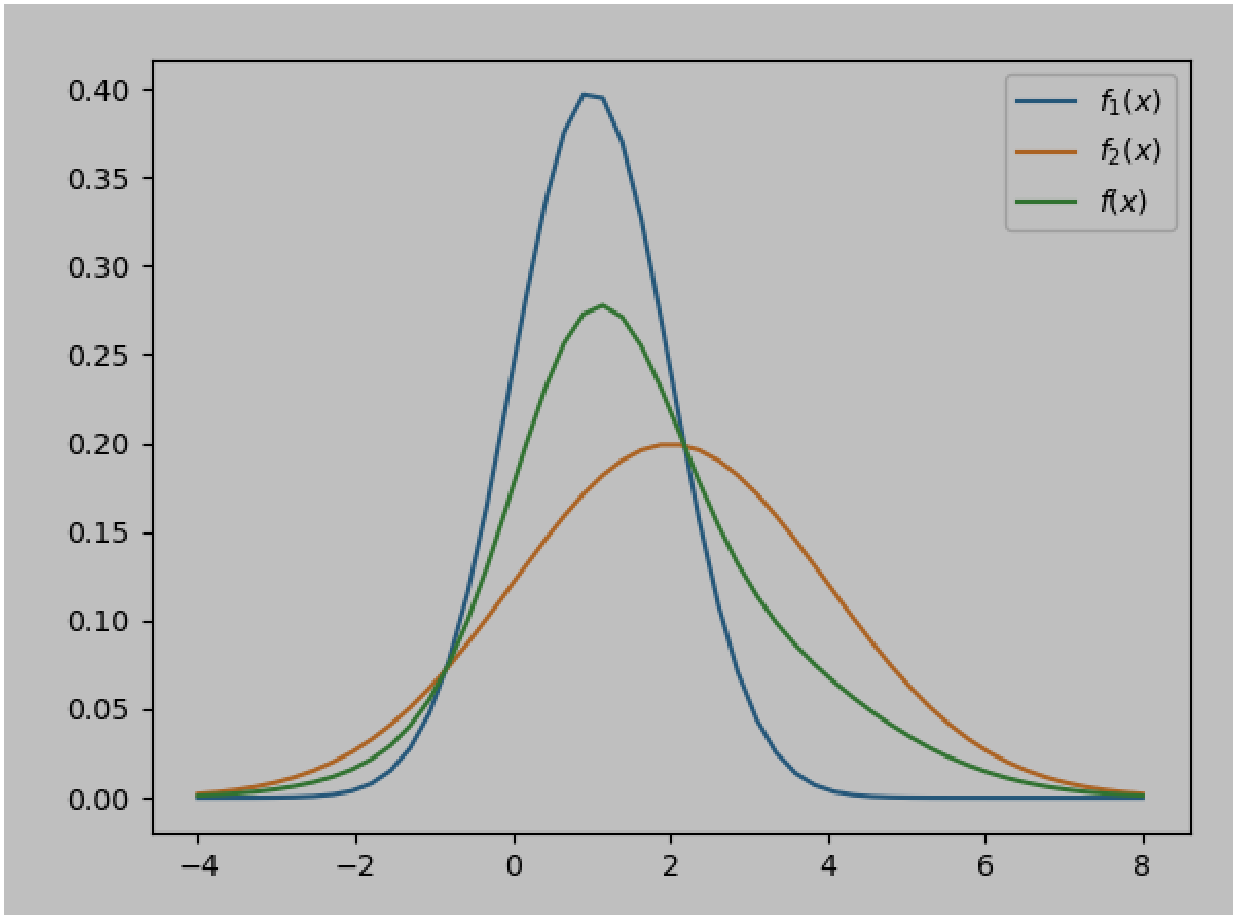
APMA 1740/2610 2025: Homework 4

1. **A simple two-category classification problem.** Let $Y \in \{1, 2\}$ be a random variable with $\mathbb{P}(Y = 1) = 0.45$. Given $Y = y$, let X be a $\mathrm{N}(y, y^2)$ random variable (i.e., normal with mean y and standard deviation y). Let $f(x)$ denote the pdf of X, and let $f_y(x) = f(x|Y = y)$ denote the conditional pdf of X given that $Y = y$. Let $r_y(x) = \mathbb{P}(Y = y|X = x)$ denote the conditional probability that $Y = y$ given $X = x$, defined in the usual way with Bayes' rule.

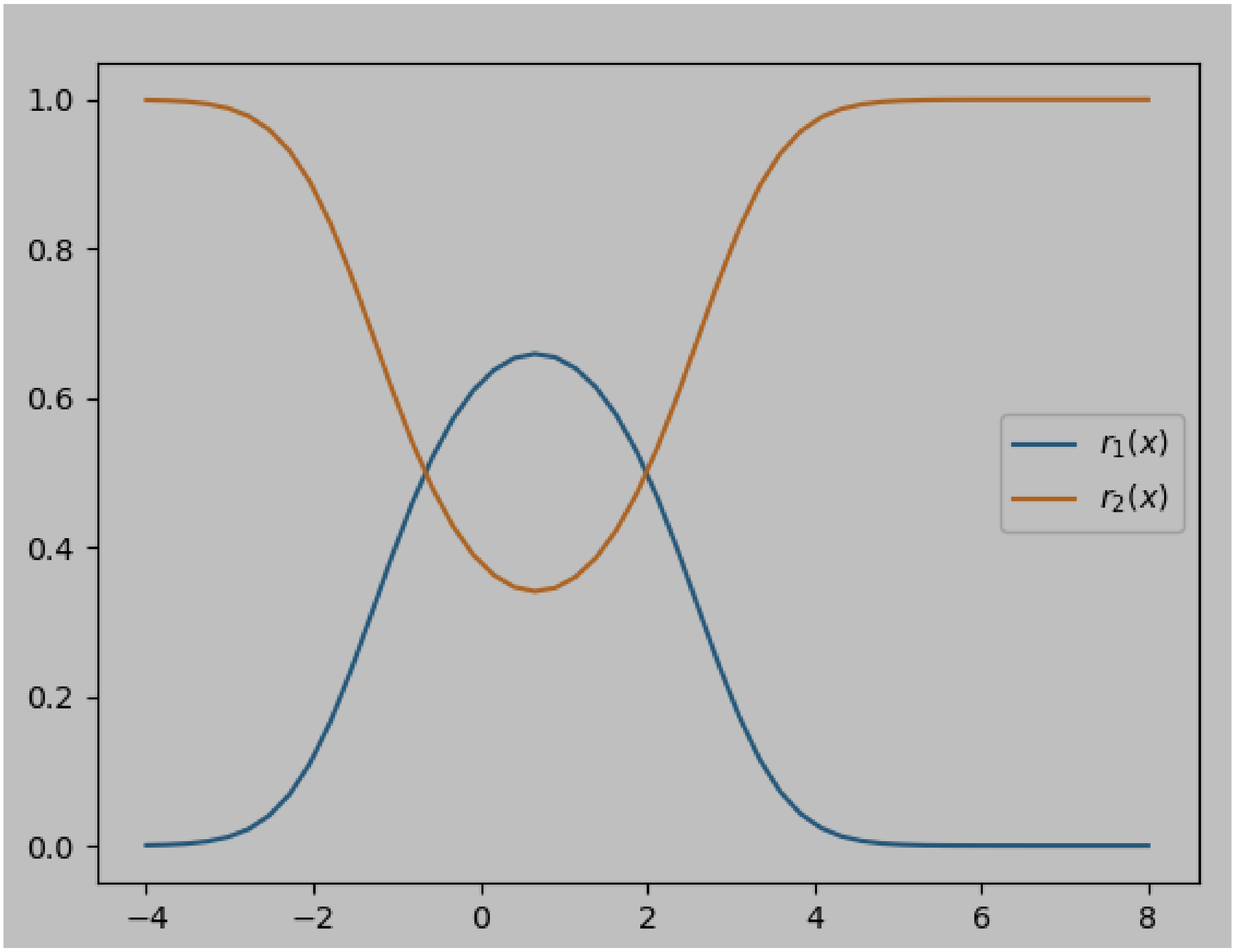
   (a) Find expressions for $f(x)$, $f_1(x)$, and $f_2(x)$. Plot all 3 on the same graph for $x \in (-4, 8)$.

$$\begin{aligned}
f_1(x) &= f(x \mid Y = 1) \\
&\sim \mathcal{N}(1, 1) \\
&= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x-1)^2}{2}\right) \\
f_2(x) &= f(x \mid Y = 2) \\
&=\sim \mathcal{N}(2, 4) \\
&= \frac{1}{2\sqrt{2\pi}} \exp\left(-\frac{(x-2)^2}{8}\right) \\
f(x) &= 0.45 f_1(x) + 0.55 f_2(x) \\
&= \frac{0.45}{\sqrt{2\pi}} \exp\left(-\frac{(x-1)^2}{2}\right) + \frac{0.55}{2\sqrt{2\pi}} \exp\left(-\frac{(x-2)^2}{8}\right)
\end{aligned}$$



   (b) Find expressions for $r_1(x)$ and $r_2(x)$. Plot both on the same graph for $x \in (-4, 8)$.

$$\begin{aligned} r_1(x) &= \mathbb{P}(Y = 1 | X = x) \\ &= \frac{\mathbb{P}(X = x | Y = 1)\mathbb{P}(Y = 1)}{\mathbb{P}(X = x)} \\ &= \frac{f_1(x)\mathbb{P}(Y = 1)}{f(x)} \\ &= 0.45\frac{f_1(x)}{f(x)} \\ r_2(x) &= 0.55\frac{f_2(x)}{f(x)} \end{aligned}$$



(c) Let $B \triangleq \{x : r_2(x) > r_1(x)\}$. Find the set B.

$$\begin{aligned}
B &= \{x : r_2(x) > r_1(x)\} \\
&= \{x : 0.55\frac{f_2(x)}{f(x)} > 0.45\frac{f_1(x)}{f(x)}\} \\
&= \{x : 0.55 f_2(x) > 0.45 f_1(x)\} \\
&= \{x : 0.55\frac{1}{2\sqrt{2\pi}}\exp\left(-\frac{(x-2)^2}{8}\right) > 0.45\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{(x-1)^2}{2}\right)\} \\
&= \left\{x : \exp\left(-\frac{(x-2)^2}{8}\right) > \frac{90}{55}\exp\left(-\frac{(x-1)^2}{2}\right)\right\} \\
&= \left\{x : -\frac{(x-2)^2}{8} > \log\left(\frac{18}{11}\right) - \frac{(x-1)^2}{2}\right\} \\
&= \left\{x : -\frac{x^2-4x+4}{8} > \log\left(\frac{18}{11}\right) - \frac{x^2-2x+1}{2}\right\} \\
&= \left\{x : -x^2+4x-4 > 8\log\left(\frac{18}{11}\right) - 4x^2+8x-4\right\} \\
&= \left\{x : 3x^2-4x > 8\log\left(\frac{18}{11}\right)\right\} \\
&\approx \boxed{\{x : x < -0.341\} \cup \{x : x > 1.674\}}
\end{aligned}$$

2. **Optimal and sub-optimal classifiers.** Problem 1 continued: Consider the family of classifiers

$$h_t(X) = 1 + \mathbb{1}_{X>t} = \begin{cases} 2 & \text{if } X > t \\ 1 & \text{if } X \leq t \end{cases}$$

for $t \in \mathbb{R}$.

(a) Calculate[1] the error probabilities $\mathbb{P}(h_{1.5}(X) \neq Y | Y = 1)$ and $\mathbb{P}(h_{1.5}(X) \neq Y | Y = 2)$.

$$\mathbb{P}(h_{1.5}(X) \neq Y \mid Y = 1) = \mathbb{P}(h_{1.5}(X) = 2 \mid Y = 1) = \mathbb{P}(X > 1.5 \mid Y = 1) = 1 - \int_{-\infty}^{1.5} f_1(x)\, dx \approx \boxed{0.3085}$$

$$\mathbb{P}(h_{1.5}(X) \neq Y \mid Y = 2) = \mathbb{P}(h_{1.5}(X) = 1 \mid Y = 2) = \mathbb{P}(X \leq 1.5 \mid Y = 2) = \int_{-\infty}^{1.5} f_2(x)\, dx \approx \boxed{0.4013}$$

(b) Calculate the classification error rate $\mathbb{P}(h_{1.5}(X) \neq Y)$.

$$\begin{aligned} \mathbb{P}(h_{1.5}(X) \neq Y) &= \mathbb{P}(h_{1.5}(X) \neq Y \mid Y = 1)\mathbb{P}(Y = 1) + \mathbb{P}(h_{1.5}(X) \neq Y \mid Y = 2)\mathbb{P}(Y = 2) \\ &= 0.45 \cdot \mathbb{P}(X > 1.5 \mid Y = 1) + 0.55 \cdot \mathbb{P}(X \leq 1.5 \mid Y = 2) \\ &\approx 0.45 \cdot 0.3085 + 0.55 \cdot 0.4013 \\ &\approx \boxed{0.3596} \end{aligned}$$

(c) (Do not submit.) Generate $n = 10^6$ iid pairs $(X_1, Y_1), (X_2, Y_2), \ldots, (X_n, Y_n)$ from this distribution and find the fraction of times that $h_{1.5}(X_k) \neq Y_k$. Report your results. If your answer is not very close to part (b), then something is wrong.

(d) Find another value of $t \neq 1.5$ so that the classifier h_t has smaller classification error rate than $h_{1.5}$. Compute the two error probabilities from part (a) and the classification error rate from part (b) for your improved classifier.

For $\boxed{t = 2}$,

$$\mathbb{P}(h_2(X) \neq Y \mid Y = 1) = 1 - \int_{-\infty}^{2} f_1(x)\, dx \approx \boxed{0.1587}$$

$$\mathbb{P}(h_2(X) \neq Y \mid Y = 2) = \int_{-\infty}^{2} f_2(x)\, dx \approx \boxed{0.5}$$

$$\mathbb{P}(h_2(X) \neq Y) \approx \boxed{0.3464} < 0.3596 = \mathbb{P}(h_{1.5}(X) \neq Y)$$

(e) Find the classifier h^* (not necessarily of the form h_t above) that has the smallest possible classification error rate.

Bayes' Rule tells us that

$$h^*(x) = \arg\max_{i=1:c} \mathbb{P}(Y = i \mid X = x)$$

For $Y \in \{1, 2\}$,

$$\begin{aligned} \mathbb{P}(Y = 1 \mid X = x) &= \frac{\mathbb{P}(Y = 1)\mathbb{P}(X = x \mid Y = 1)}{\mathbb{P}(X = x)} \\ &= \frac{0.45 f_1(x)}{f(x)} \\ \mathbb{P}(Y = 2 \mid X = x) &= \frac{\mathbb{P}(Y = 2)\mathbb{P}(X = x \mid Y = 2)}{\mathbb{P}(X = x)} \\ &= \frac{0.55 f_2(x)}{f(x)} \end{aligned}$$

[1] This refers to the distribution over (X, Y) defined in Problem 1. You will need the cumulative distribution function (cdf) of a normal random variable, which you can get in Matlab with `normcdf`.

Hence,

$$h^*(x) = \begin{cases} 2 & \text{if } 0.45 f_1(x) < 0.55 f_2(x) \\ 1 & \text{otherwise} \end{cases}$$

(f) Compute the two error probabilities from part (a) and the classification error rate from part (b) for h^*.

$$\begin{aligned} \mathbb{P}(h^*(X) \neq Y \mid Y = 1) &= \mathbb{P}(h^*(X) = 2 \mid Y = 1) \\ &= \mathbb{P}(0.45 f_1(X) < 0.55 f_2(X) \mid Y = 1) \\ &= \mathbb{P}(X \leq t \mid Y = 1) \qquad \text{for } 0.45 f_1(t) = 0.55 f_2(t) \\ \mathbb{P}(h^*(X) \neq Y \mid Y = 2) &= \mathbb{P}(h^*(X) = 1 \mid Y = 2) \\ &= \mathbb{P}(X > t \mid Y = 2) \qquad \text{for } 0.45 f_1(t) = 0.55 f_2(t) \end{aligned}$$

We saw in Problem 1.c that

$$0.45 f_1(t) = 0.55 f_2(t) \iff 3t^2 - 4t - 8\log(\frac{18}{11}) = 0 \iff t = \frac{4 \pm \sqrt{16 + 96 \log \frac{18}{11}}}{6}$$

But as we saw in Part d, we want

$$t = \frac{4 + \sqrt{16 + 96 \log \frac{18}{11}}}{6} \approx 1.992$$

so

$$\begin{aligned} \mathbb{P}(h^*(X) \neq Y \mid Y = 1) &= \int_{-\infty}^{t} f_1(x)\, dx \approx \boxed{0.1605} \\ \mathbb{P}(h^*(X) \neq Y \mid Y = 2) &= 1 - \int_{-\infty}^{t} f_2(x)\, dx \approx \boxed{0.4985} \\ \mathbb{P}(h^*(X) \neq Y) &= 0.45\mathbb{P}(h^*(X) \neq Y \mid Y = 1) + 0.55\mathbb{P}(h^*(X) \neq Y \mid Y = 2) \approx \boxed{0.3464} \end{aligned}$$

3. **Preprocessing cannot improve optimal classification performance.** Consider a pair of random variables (X, Y) where $Y \in \{1, \ldots, s\}$. Let L be the optimal classification error rate for predicting Y from X. Suppose we transform X into $\tilde{X} = g(X)$ for some function g, and predict Y using $\tilde{X}$ instead of X. In other words, we preprocess the data prior to classification. Let $\tilde{L}$ be the optimal classification error rate for predicting Y from $\tilde{X}$.

(a) Prove that $\tilde{L} \geq L$, i.e., preprocessing can only create more error if we are using the optimal classifier.[2]

Let $\tilde{h}$ be the optimal classifier for predicting Y from $\tilde{X}$ and h be the optimal classifier for predicting Y from X. By Bayes' Rule, we can write

$$\begin{aligned} L &= \mathbb{P}(h(X) \neq Y) = 1 - \mathbb{P}(h(X) = Y) \\ &= 1 - \int_x \mathbb{P}(Y = h(x) \mid X = x) f(x)\, dx \\ &= 1 - \int_x \max_c \mathbb{P}(Y = c \mid X = x) f(x)\, dx \\ &= 1 - \mathbb{E}_X[\max_c \mathbb{P}(Y = c \mid X)] \\ \tilde{L} &= 1 - \mathbb{E}_{\tilde{X}}[\max_c \mathbb{P}(Y = c \mid \tilde{X})] \end{aligned}$$

Hence, it suffices to show

$$\begin{aligned} \mathbb{E}_{\tilde{X}}[\max_c \mathbb{P}(Y = c \mid \tilde{X})] &\leq \mathbb{E}_X[\max_c \mathbb{P}(Y = c \mid X)] \\ \mathbb{E}_{g(X)}[\max_c \mathbb{P}(Y = c \mid g(X))] &\leq \mathbb{E}_X[\max_c \mathbb{P}(Y = c \mid X)] \end{aligned}$$

Notice,

$$\begin{aligned} \mathbb{P}(Y = c \mid g(X)) &= \mathbb{E}[\mathbb{1}_{Y=c} \mid g(X)] \\ \mathbb{P}(Y = c \mid X) &= \mathbb{E}[\mathbb{1}_{Y=c} \mid X] \end{aligned}$$

so

$$\begin{aligned} \mathbb{E}[\mathbb{P}(Y = c \mid X) \mid g(X)] &= \mathbb{E}[\mathbb{E}[\mathbb{1}_{Y=c} \mid X] \mid g(X)] \\ &= \mathbb{P}(Y = c \mid g(X)) \end{aligned}$$

But since $\max$ is a convex function, we can apply Jensen's inequality to get

$$\begin{aligned} \max_c \mathbb{P}(Y = c \mid g(X)) &= \max_c \mathbb{E}[\mathbb{P}(Y = c \mid X) \mid g(X)] \\ &\leq \mathbb{E}[\max_c \mathbb{P}(Y = c \mid X) \mid g(X)] \end{aligned}$$

Taking expected values,

$$\begin{aligned} \mathbb{E}[\max_c \mathbb{P}(Y = c \mid g(X))] &\leq \mathbb{E}[\mathbb{E}[\max_c \mathbb{P}(Y = c \mid X) \mid g(X)]] \\ &= \mathbb{E}[\max_c \mathbb{P}(Y = c \mid X]] \\ 1 - \tilde{L} &\leq 1 - L \end{aligned}$$

hence, $\tilde{L} \geq L$.

(b) Prove that $\tilde{L} = L$ if the function g is invertible.

Suppose g is invertible.

[2] This result is analogous to the data processing inequality in information theory (if you've seen that before), which says that the mutual information between Y and $\tilde{X}$ is less than or equal to the mutual information between Y and X. At best, preprocessing loses no information.

In part (a), we showed that for any $g : x \mapsto g(X)$,

$$1 - \tilde{L} = \mathbb{E}[\max_c \mathbb{P}(Y = c \mid g(X))] \leq \mathbb{E}[\max_c \mathbb{P}(Y = c \mid X)] = 1 - L$$

However, by assumption, $\exists g^{-1}$ such that $g^{-1}(g(X)) = X$ so we can apply the same argument to g^{-1} to get

$$\mathbb{E}[\max_c \mathbb{P}(Y = c) \mid g^{-1}(g(X))] \leq \mathbb{E}[\max_c \mathbb{P}(Y = c \mid g(X))]$$

but since $g^{-1}(g(X)) = X$, we have

$$\begin{cases} \mathbb{E}[\max_c \mathbb{P}(Y = c) \mid X] \leq \mathbb{E}[\max_c \mathbb{P}(Y = c \mid g(X))] \\ \mathbb{E}[\max_c \mathbb{P}(Y = c \mid g(X))] \leq \mathbb{E}[\max_c \mathbb{P}(Y = c \mid X)] \end{cases} \begin{aligned} &\implies \mathbb{E}[\max_c \mathbb{P}(Y = c) \mid X] = \mathbb{E}[\max_c \mathbb{P}(Y = c \mid X)] \\ &\implies 1 - \tilde{L} = 1 - L \\ &\implies \tilde{L} = L \quad \blacksquare \end{aligned}$$

(c) Suppose that $s = 2$ and that the class conditional distributions for X given Y have pdfs $f_1(x)$ and $f_2(x)$ over $\mathbb{R}^d$. Suppose that g is invertible and that both g and g^{-1} are continuously differentiable. Let $\tilde{f}_1(\tilde{x})$ and $\tilde{f}_2(\tilde{x})$ be the class conditional pdfs of $\tilde{X}$ given Y.[3] Show that the Neyman-Pearson classifiers for Y given X and for Y given $\tilde{X}$ produce the same ROC curves.

Let

$$h_t(X) = \begin{cases} 1 & \text{if } \frac{f_2(X)}{f_1(X)} < t \\ 2 & \text{otherwise} \end{cases}$$

$$\tilde{h}_t(\tilde{X}) = \begin{cases} 1 & \text{if } \frac{\tilde{f}_2(\tilde{X})}{\tilde{f}_1(\tilde{X})} < t \\ 2 & \text{otherwise} \end{cases}$$

Then for all t, we want

$$\frac{\mathbb{P}(\tilde{h}_t(\tilde{X}) = 2 \mid Y = 2)}{\mathbb{P}(\tilde{h}_t(\tilde{X}) = 2 \mid Y = 1)} = \frac{\mathbb{P}(h_t(X) = 2 \mid Y = 2)}{\mathbb{P}(h_t(X) = 2 \mid Y = 1)}$$

It suffices to show that, for all t, $h_t(X) = \tilde{h}_t(\tilde{X})$ or equivalently

$$\frac{f_2(X)}{f_1(X)} < t \iff \frac{\tilde{f}_2(\tilde{X})}{\tilde{f}_1(\tilde{X})} < t$$

However, by the change of variables formula, we know that

$$\frac{\tilde{f}_2(\tilde{X})}{\tilde{f}_1(\tilde{X})} = \frac{f_2(g^{-1}(\tilde{X}))\left|\det J^{g^{-1}}(\tilde{X})\right|}{f_1(g^{-1}(\tilde{X}))\left|\det J^{g^{-1}}(\tilde{X})\right|} = \frac{f_2(g^{-1}(\tilde{X}))}{f_1(g^{-1}(\tilde{X}))} = \frac{f_2(g^{-1}(g(X)))}{f_1(g^{-1}(g(X)))} = \frac{f_2(X)}{f_1(X)}$$

[3] **Change of variables/change of densities.** Let $g : A \subseteq \mathbb{R}^d \to B \subseteq \mathbb{R}^d$ be invertible and onto (i.e. $\forall \tilde{x} \in B, \exists x \in A$ such that $g(x) = \tilde{x}$) and let g^{-1} be the inverse of g. Suppose that both g and g^{-1} are continuously differentiable. If X is a random vector taking values in $\mathbb{R}^d$ and having density f_X, then what is the density of $\tilde{X} \doteq g(X)$? This is just a change of variables problem, as follows:

$$f_X(x)dx = \quad (\text{c.o.v. } x = g^{-1}(\tilde{x}))\ \ f_X\left(g^{-1}(\tilde{x})\right)\left|\frac{dx}{d\tilde{x}}\right| d\tilde{x} = f_X\left(g^{-1}(\tilde{x})\right)\left|\det J^{g^{-1}}(\tilde{x})\right| d\tilde{x}$$

where $J^{g^{-1}}$ is the $d \times d$ Jacobian matrix, with i, j entry $\frac{\partial g_i^{-1}(\tilde{x})}{\partial \tilde{x}_j}$, and g_i^{-1} is the i component of $g^{-1}(\tilde{x}) \in \mathbb{R}^d$. Therefore, for any $S \subseteq A$

$$\int_{x \in S} f_X(x)dx = \int_{\tilde{x} \in g(S)} f_X\left(g^{-1}(\tilde{x})\right)\left|\det J^{g^{-1}}(\tilde{x})\right| d\tilde{x} \implies f_{\tilde{X}}(\tilde{x}) = f_X\left(g^{-1}(\tilde{x})\right)\left|\det J^{g^{-1}}(\tilde{x})\right|$$

Or in the other direction, $f_X(x) = f_{\tilde{X}}\left(g(x)\right)|\det J^g(x)|$. Notice then that $|\det J^g(x)| = \dfrac{1}{\left|\det J^{g^{-1}}(g(x))\right|}$, which makes sense for an area element. Notice also that therefore the smoothness conditions on g and its inverse imply, among other things, that the Jacobian determinant is everywhere non-zero.

which gives us the strictly stronger result that for g invertible and $g, g^{-1} \in C^1$,

$$\frac{\tilde{f}_2(\tilde{X})}{\tilde{f}_1(\tilde{X})} = \frac{f_2(X)}{f_1(X)} \implies \tilde{h}_t(\tilde{X}) = h_t(X) \implies \frac{\mathbb{P}(\tilde{h}_t(\tilde{X}) = 2 \mid Y = 2)}{\mathbb{P}(\tilde{h}_t(\tilde{X}) = 2 \mid Y = 1)} = \frac{\mathbb{P}(h_t(X) = 2 \mid Y = 2)}{\mathbb{P}(h_t(X) = 2 \mid Y = 1)}$$

for all values of t. Hence, the Neyman-Pearson classifiers for Y given X and for Y given $\tilde{X}$ produce the same ROC curves.

Remark: Preprocessing, i.e., finding good features, is often crucial for real-world classification, even though you just showed that it can only hurt optimal performance. The reason is that in most practical situations we do not know the underlying distributions and, consequently, we cannot do optimal classification. Preprocessing can make it easier for a statistical procedure to learn a good classifier, even though the classifier may not be optimal.

4. (Do not submit.) **Constructing the ROC curve.** Consider a two-category classification problem for (X, Y) where $X \in \mathbb{R}^2$ and $Y \in \{1, 2\}$ with $\pi_1 \triangleq \mathbb{P}(Y = 1) = 0.4$ and $\pi_2 \triangleq \mathbb{P}(Y = 2) = 0.6$. We will describe the distribution of $X = (X_1, X_2)$ in terms of polar coordinates (R, U) that satisfy

$$(X_1, X_2) = (R \cos U, R \sin U)$$

or, equivalently,

$$(R, U) = (\sqrt{X_1^2 + X_2^2}, \text{atan2}(X_2, X_1)).$$

Conditional on $Y = 1$, R and U are independent with $R \sim \text{Gamma}(20, 0.1)$ and $U \sim \text{Uniform}(0, 2\pi)$. Conditional on $Y = 2$, R and U are independent with $R = \sqrt{2E}$ for $E \sim \text{Exponential}(1)$ and $U \sim \text{Uniform}(0, 2\pi)$.[4]

(a) Sample 1000 iid pairs from this distribution and make three scatter plots: (i) $X|Y = 1$, (ii) $X|Y = 2$, and (iii) X for any Y, but with different symbols for the two different classes. $X|Y = 1$ should be a noisy ring and $X|Y = 2$ should be a noisy disk and they should overlap significantly.

(b) Let $f_1(r, u)$ and $f_2(r, u)$ denote the conditional pdfs of (R, U) given $Y = 1$ and given $Y = 2$, respectively. Derive expressions for f_1 and f_2. Find an expression for the family of Neyman-Pearson classifiers, say $(h_t : 0 < t < \infty)$ using f_1 and f_2. Is this the same family of classifiers that you would have derived if you had used rectangular coordinates instead of polar coordinates? Why or why not?

(c) (Approximately) compute and plot the optimal ROC curve traced out by the family of Neyman-Pearson classifiers. Do this by generating $n = 10^5$ iid samples from (X, Y), called the *test data set*, and estimating the DR (detection rate) and FAR (false alarm rate) for each h_t as you vary t to trace out the ROC curve.[5]

(d) Approximately, what is the optimal classification error rate, i.e, $\inf_h \mathbb{P}(h(X) \neq Y)$, where the infimum is taken over all possible classifiers? What classifier achieves this optimal error rate?

[4] Gamma(a, b) has pdf

$$f(x; a, b) \triangleq \frac{1}{b^a \Gamma(a)} x^{a-1} \exp(-x/b) \mathbb{1}\{x \geq 0\}$$

and you can use the Matlab function `gamrnd(a,b,m,n)` to generate an $m \times n$ array of iid gamma(a, b) random variables. (This is just one of several parameterizations that are commonly used for the Gamma distribution.) Exponential(c) has pdf

$$f(x; c) \triangleq \frac{1}{c} \exp(-x/c) \mathbb{1}\{x \geq 0\}$$

and you can use the Matlab function `exprnd(c,m,n)` to generate an $m \times n$ array of iid exponential(c) random variables.

[5] You should only create a single test data set, i.e., you should use the same test data set for each h_t: For each t, you compute h_t on all 10^5 feature vectors (X) and compare h_t to the true class labels (Y) to approximate DR and FAR. One issue is how to choose an appropriate grid of t values. A good strategy is to first compute f_2/f_1 on all 10^5 test data points, sort the ratios, and choose every 100th value, to get a list of 1000 candidate values of t. Include the values $-\infty$ and ∞ in your list of t to get the points $(1, 1)$ and $(0, 0)$ on your ROC curve. Linearly interpolate points on the ROC curve so that you draw a smooth curve.

5. (Do not submit.) **Information versus the curse of dimensionality.** Now we will modify the distribution in problem 4 by increasing the dimensionality of the feature space from 2 to d. The distribution of (X_1, X_2) remains the same. Conditional on $Y = c$, $(X_3, \dots, X_d)$ are (conditionally) independent with $X_i \sim$ Normal$(c/i, 1)$. (Note that each new feature adds discriminatory power, because its distribution is different for each class, but that the amount of discriminatory power is decreasing. This is a fairly common situation for high-dimensional data.)

   (a) Let $f_1(r, u, x_3, \dots, x_d)$ and $f_2(r, u, x_3, \dots, x_d)$ denote the conditional pdfs of $(R, U, X_3, \dots, X_d)$ given $Y = 1$ and given $Y = 2$, respectively. Derive expressions for f_1 and f_2. Find an expression for the family of Neyman-Pearson classifiers, $\{(h_t : 0 < t < \infty\}$, using f_1 and f_2. Is this the same family of classifiers that you would have derived if you had used $X_{1:d}$ instead of $R, U, X_{3:d}$? Why or why not?

   (b) Repeat Problem 4c,d for the cases $d = 5$, $d = 50$, $d = 200$.

6. (Do not submit.) **Bias and variance in the k-nearest neighbor classifier.** On this problem you will compute the ROC curve for a k-nearest neighbor classifier on the same data distribution, but pretending that this distribution is *unknown*. Except for reproducing the optimal ROC curve from Problems 4 and 5, you should not use the true data distribution in this problem. In practice, for statistical classification, the true data distribution is unknown.

Download the text files `X.dat` and `Y.dat` that are posted with this HW.[6] The file `X.dat` contains an $m \times d$ matrix of $m = 2000$ iid samples of X using $d = 200$. The file `Y.dat` contains an $m \times 1$ vector of $m = 2000$ iid samples of Y. The ith rows of `X.dat` and `Y.dat` are paired, giving a joint sample of (X, Y), for $i = 1, \ldots, m$.

When testing the performance of a statistical classifier, it is of the utmost importance to use separate datasets for training and testing. So the first thing you should do is partition the downloaded data `X` and `Y` into a *training* dataset and a *testing* dataset. Use rows $1, \ldots, 1000$ for the training dataset and rows $1001, \ldots, 2000$ for the testing dataset. In parts of this problem you will investigate how performance varies with the amount of training data n, in which case you will use subsets of size n from the training dataset for training (n will never be more than 1000). Regardless of the amount of training data, use all 1000 rows of the testing data for testing (i.e., rows $1001, \ldots, 2000$ of the downloaded data).

Building a k-nearest neighbor classifier in Matlab is easy. First you train the model using only the training data:

```
Xtrain = X(1:n,1:d);
Ytrain = Y(1:n);
mdl = fitcknn(Xtrain,Ytrain,'NumNeighbors',k);
```

The function `fitcknn` trains the model (which for a k-NN classifier involves, at most, computing a data structure to facilitate searching later for neighbors). The first argument is the feature vector data arranged as an $n \times d$ matrix, where n is the number of training points and d is the number of feature dimensions. Since you are varying both n and d in this problem, you will need to specify them ahead of time. The second argument is the class label data arranged as an $n \times 1$ vector, where the ith label corresponds to the ith row of the training data. There are many optional arguments, but the only one that you will vary is the number of neighbors k.

Once you have trained the model and stored it in a variable (called `mdl` above), then you can use it to predict the label of new data like this: `yhat = predict(mdl,x)`, where `x` is a new feature vector and `yhat` is the predicted label. The default prediction uses majority voting (to approximate the Bayes classifier), but this only corresponds to a single point on the ROC curve. In this problem you want to approximate the entire curve. A k-NN classifier traces out an ROC curve by varying the fraction t of the k neighbors that need to be class 2 in order for the classifier to guess $Y = 2$. So, instead of using the `predict` function to actually predict, we will use it to get the k-NN approximation $\hat{r}_{c,k}(x)$, which is exactly the fraction of neighbors in each class. This is given by the second output argument of `predict`:

```
Xtest = X(1001:2000,1:d);
Ytest = Y(1001:2000);
[~,rhat] = predict(mdl,Xtest);
```

(This symbol ~ is tilde.) Note that we are predicting for the observations in the *testing* data, not the training data that we used for `mdl` (as long as $n \le 1000$). We are also simultaneously predicting on all 1000 testing points; there is no need to loop over the testing points and separately predict on each one. The output `rhat` will be a 1000×2 matrix, where `rhat(i,c)` gives $\hat{r}_{c,k}(x)$ (x being the ith testing data point). Once you know $\hat{r}_{c,k}$, you can build a family of classifiers by varying the threshold for when to guess class 2 versus class 1 as you compare $\hat{r}_{2,k}$ to $\hat{r}_{1,k}$.

(a) (Approximately) compute the ROC curve for the k-nearest neighbor classifier using this training data for the first $d = 2$ features.[7] Do this for each of $k = 1, 5, 21, 101, 501$. Draw the five curves all on the same plot together and with the optimal ROC curve that you computed in Problems 4 and 5. Now repeat for the cases $d = 5$, $d = 50$, and $d = 200$. (When you are done, you should have 4 plots, each with 6 ROC curves on it. Make sure that all of the plots and curves are clearly labeled.)

(b) Repeat part (a) using $n = 200$ training points. (Using $k > n$ is equivalent to using $k = n$.)

[6] You can load these files into Matlab with the commands `load X.dat` and `load Y.dat`. The 2000×200 matrix `X` and the 2000×1 vector `Y` will be in your workspace.

[7] You proceed the same way as Problems 4c and 5b, especially footnote 5, except that instead of a classifier based on comparing f_2/f_1 to a threshold, you compare the fraction of k-nearest neighbors that are class 2 to a threshold. This fraction is given in the second column of `rhat`. (Also, you do not need to generate your own training or testing data — use the appropriate part of the downloaded data as indicated above.) Linearly interpolate points on the ROC curve so that you draw a continuous curve. This is especially important for small k, since there are only $k + 2$ meaningful thresholds.

(c) Concerning the results from (a) and (b), what (if any) trends do you find when: (i) fixing d and k and varying n; (ii) fixing k and n and varying d; and (iii) fixing n and d and varying k? Use the analyses developed in the lecture on k-nearest neighbor classifiers (especially vis-a-vis bias and variance and the effects of n and k on the radius $R_k(x)$) to interpret the three trends. (Concerning (ii), it might help to think about the effect of added dimensions on the distances between training points.)

7. **Parametrization, over parametrization, and decision surfaces.** Consider a classification problem for feature vectors $x \in \mathbb{R}^d$ with s categories, $Y \in \{1, 2, \dots, s\}$.

(a) Here is a common way to write the softmax model for $r_c(x) \triangleq \mathbb{P}(Y = c|x) = \mathbb{E}[\mathbb{1}_{Y=c}|x]$:

$$r_c(x) = \frac{e^{\beta_c \cdot x}}{\sum_{\tilde{c}=1}^{s} e^{\beta_{\tilde{c}} \cdot x}} \quad \forall\, c = 1 : s \tag{1}$$

where $\beta_c \in \mathbb{R}^d$ for each $c = 1 : s$. Define $\tilde{\beta}_c = \beta_c - \beta_1$ for each $c = 2 : s$, and show that:

$$\tilde{r}_c(x) \triangleq \begin{cases} \frac{e^{\tilde{\beta}_c \cdot x}}{1+\sum_{\tilde{c}=2}^{s} e^{\tilde{\beta}_{\tilde{c}} \cdot x}} & c = 2 : s \\ \\ \frac{1}{1+\sum_{\tilde{c}=2}^{s} e^{\tilde{\beta}_{\tilde{c}} \cdot x}} & c = 1 \end{cases}$$

gives the exact same classifier as in (1), but with d fewer parameters. (In other words, (1) is "over parametrized," albeit less burdened by notation.)

First consider the case $c \neq 1$. We have

$$\begin{aligned} \tilde{r}_c(x) &= \frac{e^{\tilde{\beta}_c \cdot x}}{1 + \sum_{\tilde{c}=2}^{s} e^{\tilde{\beta}_{\tilde{c}} \cdot x}} \\ &= \frac{e^{\beta_c \cdot x - \beta_1 \cdot x}}{1 + \sum_{\tilde{c}=2}^{s} e^{\beta_{\tilde{c}} \cdot x - \beta_1 \cdot x}} \\ &= \frac{e^{\beta_c \cdot x}}{e^{\beta_1 \cdot x} + \sum_{\tilde{c}=2}^{s} e^{\beta_{\tilde{c}} \cdot x}} \\ &= \frac{e^{\beta_c \cdot x}}{\sum_{\tilde{c}=1}^{s} e^{\beta_{\tilde{c}} \cdot x}} = r_c(x) \end{aligned}$$

so indeed $\tilde{r}_c(x) = r_c(x)$ for all $c = 2 : s$.

Now we take the case $c = 1$. Notice

$$\sum_{c=1}^{s} \tilde{r}_c(x) = \sum_{c=1}^{s} r_c(x) = \sum_{c=1}^{s} \mathbb{P}(Y = c \mid x) = 1$$

so we must also have

$$\begin{aligned} \tilde{r}_1(x) &= 1 - \sum_{c=2}^{s} \tilde{r}_c(x) \\ &= 1 - \sum_{c=2}^{s} r_c(x) \qquad \text{(by first case)} \\ &= r_1(x) \end{aligned}$$

Finally, $r_c(x)$ has sd parameters $(\{\beta_c\}_{c=1:s}, \beta_i \in \mathbb{R}^d)$ while $\tilde{r}_c(x)$ has only $(s-1)d$ parameters $(\{\tilde{\beta}_c\}_{c=2:s}, \tilde{\beta}_i \in \mathbb{R}^d)$ so we conclude that $\tilde{r}_c(x)$ is a reparametrization of $r_c(x)$ with d fewer parameters.

(b) There is yet another common representation of softmax in which a constant offset, or "bias", is added to each exponent, e.g.

$$r_c(x) \triangleq \begin{cases} \frac{e^{\alpha_c + \beta_c \cdot x}}{1+\sum_{\tilde{c}=2}^{s} e^{\alpha_{\tilde{c}} + \beta_{\tilde{c}} \cdot x}} & c = 2 : s \\ \\ \frac{1}{1+\sum_{\tilde{c}=2}^{s} e^{\alpha_{\tilde{c}} + \beta_{\tilde{c}} \cdot x}} & c = 1 \end{cases}$$

Consider the minimum-error ("Bayesian") classifier

$$h^*(x) \triangleq \text{argmax}_{c=1:s} r_c(x) \tag{2}$$

Assume that $d \geq s-1$ and that each of $\beta_2, \beta_3, \ldots \beta_s$ are distinct. Show that all of the $\binom{s}{2}$ decision surfaces share an affine subspace of dimension at least $d-s+1$ or their intersection is empty.

Point: Despite what I drew in class, the categories defined by h^* form conical-like regions emanating from a common affine hyperplane. By "conical-like," I mean cones with boundaries replaced by a collection of flat surfaces. So, for example, if $d=2$ and $s=3$, then the decision boundaries are three lines passing through a common point.

For the optimal classifier h^*, we have s surfaces given by

$$\left\{ \frac{1}{1+\sum_{\bar{c}=2}^{s} \exp(\alpha_{\bar{c}} + \beta_{\bar{c}} \cdot x)}, \frac{\exp(\alpha_2 + \beta_2 \cdot x)}{1+\sum_{\bar{c}}^{s} \exp(\alpha_{\bar{c}} + \beta_{\bar{c}} \cdot x)}, \ldots, \frac{\exp(\alpha_s + \beta_s \cdot x)}{1+\sum_{\bar{c}}^{s} \exp(\alpha_{\bar{c}} + \beta_{\bar{c}} \cdot x)} \right\}$$

Then the decision boundaries are given by the intersection B_{ij} of any two of the s surfaces i, j – clearly there are $\binom{s}{2}$ such intersections.

We are interested in the intersection of all such precision boundaries, $\bigcap_{i,j=1:s} B_{ij}$.

For any two surfaces i, j, we have

$$\begin{aligned} B_{ij} &= \{x \in \mathbb{R}^d \mid r_i(x) = r_j(x)\} \\ &= \left\{ x \in \mathbb{R}^d \middle| \frac{\exp(\alpha_i + \beta_i \cdot x)}{1+\sum_{\bar{c}=2}^{s} \exp(\alpha_{\bar{c}} + \beta_{\bar{c}} \cdot x)}, \frac{\exp(\alpha_j + \beta_j \cdot x)}{1+\sum_{\bar{c}}^{s} \exp(\alpha_{\bar{c}} + \beta_{\bar{c}} \cdot x)} \right\} \\ &= \{x \in \mathbb{R}^d \mid \alpha_i + \beta_i \cdot x = \alpha_j + \beta_j \cdot x\} \\ &= \{x \in \mathbb{R}^d \mid (\beta_i - \beta_j) \cdot x = \alpha_j - \alpha_i\} \end{aligned}$$

Hence,

$$\bigcap_{i,j} B_{ij} = \left\{ x : x \cdot \left(\beta_1 - \sum_{c=2}^{s} \beta_c \right) = 1 - \sum_{c=2}^{s} \alpha_c \right\}$$

which corresponds precisely to the solution of the linear system of equations with $(s-1)$ equations and d variables. By assumption, $d \geq s-1$, so the solution space is at least $d-(s-1) = d-s+1$ dimensional.

For 2610 or for extra credit:

8. **Non-deterministic classifiers.** A non-deterministic classifier is allowed to make a stochastic prediction using an independent source of randomness. For example, we might have a Neyman-Pearson classifier that predicts $Y = 2$ if $f_2(x) > f_1(x)$ and predicts $Y = 1$ if $f_1(x) > f_2(x)$, but randomly chooses between $Y = 1, 2$ when $f_1(x) = f_2(x)$. The performance of a non-deterministic classifier is computed by averaging over its internal source of randomness. Consider two classifiers h_0 and h_1 with false alarm rates and detection rates given by $(\text{FAR}_0, \text{DR}_0)$ and $(\text{FAR}_1, \text{DR}_1)$, respectively. Consider a non-deterministic classifier h_α created by choosing $Z \sim \text{Bernoulli}(\alpha)$ and classifying with h_Z, where Z is chosen independently every time the classifier is used. Compute the FAR and DR for h_α.

$$
\begin{aligned}
h_\alpha(x) &= \begin{cases} h_0(x) & \text{if } Z = 0 \\ h_1(x) & \text{if } Z = 1 \end{cases} \\
\text{FAR} &= \mathbb{P}(h_Z(X) = 2 \mid Y = 1) \\
&= \mathbb{P}(h_1(X) = 2, Z = 1 \mid Y = 1) + \mathbb{P}(h_0(X) = 2, Z = 0 \mid Y = 1) \\
&= \mathbb{P}(h_1(X) = 2 \mid Y = 1)\mathbb{P}(Z = 1 \mid Y = 1) + \mathbb{P}(h_0(X) = 2 \mid Y = 1)\mathbb{P}(Z = 0 \mid Y = 1) \\
&= \mathbb{P}(h_1(X) = 2 \mid Y = 1)\alpha + \mathbb{P}(h_0(X) = 2 \mid Y = 1)(1 - \alpha) \\
&= \boxed{\alpha\text{FAR}_1 + (1 - \alpha)\text{FAR}_0} \\
\text{DR} &= \mathbb{P}(h_Z(X) = 2 \mid Y = 2) \\
&= \mathbb{P}(h_1(X) = 2, Z = 1 \mid Y = 2) + \mathbb{P}(h_0(X) = 2, Z = 0 \mid Y = 2) \\
&= \mathbb{P}(h_1(X) = 2 \mid Y = 2)\mathbb{P}(Z = 1 \mid Y = 2) + \mathbb{P}(h_0(X) = 2 \mid Y = 2)\mathbb{P}(Z = 0 \mid Y = 2) \\
&= \mathbb{P}(h_1(X) = 2 \mid Y = 2)\alpha + \mathbb{P}(h_0(X) = 2 \mid Y = 2)(1 - \alpha) \\
&= \boxed{\alpha\text{DR}_1 + (1 - \alpha)\text{DR}_0}
\end{aligned}
$$

9. **Shape of the optimal ROC curve.** Consider a two-class classification problem. Let $M \subseteq [0,1]^2$ denote the set of all possible (FAR, DR) pairs for this classification problem, including non-deterministic classifiers. That is,

$$M = \{(f,d) : \exists \text{ a possibly non-deterministic classifier } h \text{ with } (\text{FAR}_h, \text{DR}_h) = (f,d)\}.$$

(a) Prove that M is a convex set[8] containing the points $(0,0)$ and $(1,1)$.

(b) The optimal ROC curve is the function

$$D(f) = \sup\{d : (f,d) \in M\},$$

which gives the largest possible DR over all classifiers with FAR $= f$. (Recall that the sup—which is short for "supremum"—of a set is its least upper bound.) Prove that D is concave and non-decreasing with $D(0) \geq 0$ and $D(1) = 1$. (Note: Neyman-Pearson theory tells us how to find a classifier that actually achieves $D(f)$ for each f, but you should not use the Neyman-Pearson Lemma in this problem, because we did not discuss the fully general case with non-deterministic classifiers and with abstract random variables that might not have pmfs or pdfs.)

[8] A set M (in a vector space) is convex if $x, y \in M$ and $\lambda \in [0,1]$ implies $\lambda x + (1-\lambda)y \in M$.

10. **Area under the ROC curve.** The area under the ROC curve (in the unit box $[0,1] \times [0,1]$) is a commonly used measure of performance for a family of classifiers. This problem develops a way to think about what that area means quantitatively. The cumulative distribution function (cdf) of a random variable Z is $F(z) = \mathbb{P}(Z \leq z)$ and the complementary cdf (ccdf) is $G(z) = \mathbb{P}(Z > z) = 1 - F(z)$.

   (a) Let Z be a real-valued random variable with a continuous cdf F and ccdf G. Define $U = F(Z)$ and $V = G(Z)$. Show that U and V are uniform$(0,1)$ random variables.

   (b) Let Z_1 and Z_2 be independent real-valued random variables with continuous and strictly monotonic cdf's F_1 and F_2, respectively, and ccdfs G_1 and G_2. Define the functions $\alpha(s) = F_1(F_2^{-1}(s))$ and $\beta(s) = G_2(G_1^{-1}(s))$ for $s \in (0,1)$, where, e.g., F_2^{-1} is the inverse function of F_2. Show that

   $$\mathbb{P}(Z_1 \leq Z_2) = \textstyle\int_0^1 \alpha(s)\, ds = \int_0^1 \beta(s)\, ds.$$

   (c) Consider a pair of random variables (X,Y) where $Y \in \{1,2\}$ and $X \in \mathcal{S}$ for some arbitrary space $\mathcal{S}$. Consider a family of classifiers $(h_t : t \in \mathbb{R})$ of the form

   $$h_t(x) = 1 + \mathbb{1}\{g(x) \geq t\},$$

   where $g : \mathcal{S} \mapsto \mathbb{R}$. Define the random variable $Z = g(X)$ and the conditional cdfs $F_c(z) = \mathbb{P}(Z \leq z | Y = c)$. Suppose that F_1 and F_2 are each continuous and strictly increasing. Let A be the area under the ROC curve traced out by h_t as t varies over $\mathbb{R}$. Let (X_1, Y_1) and (X_2, Y_2) be independent copies of (X,Y). Show that

   $$A = \mathbb{P}(g(X_1) \leq g(X_2) \mid Y_1 = 1, Y_2 = 2).$$

In other words, the area under the ROC curve is the probability that g puts independent samples $X_1 \sim \mathbb{P}(X = x|Y = 1)$ and $X_2 \sim \mathbb{P}(X = x|Y = 1)$ “in the right order.”

11. **When is weak evidence better than no evidence?** In both this homework and the previous one, we have seen that the use of weak evidence in high dimensions can contribute to inferior performance; sometimes weak evidence is best ignored. How strong does the evidence found in additional features have to be in order to contribute to improved performance? In other words, where is the cutoff? Naturally, any useable answer will come down to the specifics of the problem at hand. In this problem we will examine an idealized situation in which we can answer a related question: When does a lot of weak evidence add up to definitive evidence?

    Let $a_1, a_2, \ldots$ be an infinite sequence of real numbers ($a_i \in \mathbb{R}^1,\ \forall i = 1, 2, \ldots$). In a two-category classification problem, the class-conditional densities (on $\mathbb{R}^d,\ d \geq 1$) are

$$X = (X_1, \ldots, X_d) \sim f_1(x_1, \ldots, x_d) = \prod_{i=1}^{d} \frac{1}{\sqrt{2\pi}} e^{-x_i^2/2} \qquad \text{for class 1}$$

$$X = (X_1, \ldots, X_d) \sim f_2(x_1, \ldots, x_d) = \prod_{i=1}^{d} \frac{1}{\sqrt{2\pi}} e^{-(x_i - a_i)^2/2} \qquad \text{for class 2}$$

    Let A_d be the area under the ROC curve of the Neyman-Pearson classifier:

$$h_t(x) \triangleq \begin{cases} 2 & \text{if } \dfrac{f_2(x)}{f_1(x)} \geq t \\ & \\ 1 & \text{otherwise} \end{cases}$$

    Show that $A_d \overset{d \to \infty}{\longrightarrow} 1$ if and only if $\sum_{i=1}^{\infty} a_i^2 = \infty$.