IITM-CS5691 : Pattern Recognition and Machine Learning
Assignment 1

Release Date: March 9, 2021
Due Date : March 19, 23:59

Roll No: CS18B045
Name: Rishika Varma K
Collaborators (if any): Karedla Roshini, M Harini Saraswathy
References (if any):

- Use LaTeX to write-up your solutions (in the solution blocks of the source LaTeX file of this assignment), and submit the resulting single pdf file at GradeScope by the due date. (Note: **No late submissions** will be allowed, other than one-day late submission with 10% penalty or four-day late submission with 30% penalty! Within GradeScope, indicate the page number where your solution to each question starts, else we won't be able to grade it! You can join GradeScope using course entry code **5VDNKV**).
- For the programming question, please submit your code (rollno.ipynb file and rollno.py file in rollno.zip) directly in moodle, but provide your results/answers in the pdf file you upload to GradeScope.
- Collaboration is encouraged, but all write-ups must be done individually and independently, and mention your collaborator(s) if any. Same rules apply for codes written for any programming assignments (i.e., write your own code; we will run plagiarism checks on codes).
- If you have referred a book or any other online material for obtaining a solution, please cite the source. Again don't copy the source *as is* - you may use the source to understand the solution, but write-up the solution in your own words.
- Points will be awarded based on how clear, concise and rigorous your solutions are, and how correct your code is. Overall points for this assignment would be **min**(your score including bonus points scored, 50).

1. (10 points) [GETTING YOUR BASICS RIGHT!]

   (a) (1 point) You have a jar of $1,000$ coins. 999 are fair coins, and the remaining coin will always land heads. You take a single coin out of the jar and flip it 10 times in a row, all of which land heads. What is the probability your next toss with the same coin will land heads? Explain your answer. How would you call this probability in Bayesian jargon?

   Solution: The coin that was picked up could be the biased coin with probability $1/1000$ and a fair coin with probability $999/1000$.

   If coin is fair then probability of a head is $1/2$ whereas for the biased coin it is 1. For 10 flips the probability of getting all heads will be $(1/2)^{10}$ and 1 respectively.

   Given that the first 10 flips are heads, the probability that the coin picked could be the biased coin is given by:

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}, \text{ if } P(B) \neq 0,$$

$$P(\mathrm{biasedCoin} \mid 10\mathrm{heads}) = \frac{P(\mathrm{biasedCoin} \cap 10\mathrm{heads})}{P(10\mathrm{heads})}$$

$$P(\mathrm{biasedCoin} \mid 10\mathrm{heads}) = \frac{1 * \frac{1}{1000}}{1 * \frac{1}{1000} + \left(\frac{1}{2}\right)^{10} * \frac{999}{1000}} = \frac{2^{10}}{2^{10} + 999}$$

Similarly, probability that the coin picked up is fair is given by:

$$P(\mathrm{fairCoin} \mid 10\mathrm{heads}) = \frac{P(\mathrm{fairCoin} \cap 10\mathrm{heads})}{P(10\mathrm{heads})}$$

$$P(\mathrm{fairCoin} \mid 10\mathrm{heads}) = \frac{\left(\frac{1}{2}\right)^{10} * \frac{999}{1000}}{1 * \frac{1}{1000} + \left(\frac{1}{2}\right)^{10} * \frac{999}{1000}} = \frac{999}{2^{10} + 999}$$

Now to calculate probability of next coin also being heads, we need to probability of that happening in both cases.

$$\begin{aligned} P(11\mathrm{thcoinhead} \mid 10\mathrm{heads}) =& P(11\mathrm{thcoinhead} \mid \mathrm{fairCoin}) * P(\mathrm{fairCoin} \mid 10\mathrm{heads}) \\ &+ P(11\mathrm{thcoinhead} \mid \mathrm{biasedCoin}) * P(\mathrm{biasedCoin} \mid 10\mathrm{heads}) \\ &= \frac{1}{2} * \frac{999}{2^{10} + 999} + 1 * \frac{2^{10}}{2^{10} + 999} \\ &= 0.753 \end{aligned}$$

This probability is called the posterior in bayesian jargon.

(b) (3 points) Consider the i.i.d data $\mathbf{X} = \{x_i\}_{i=1}^n$, such that each $x_i \sim \mathcal{N}(\mu, \sigma^2)$. We have seen ML estimates of μ, σ^2 in class by setting the gradient to zero. How can you argue that the stationary points so obtained are indeed global maxima of the likelihood function? Next, derive the bias of the MLE of μ, σ^2.

Solution: This can be proved by showing that throughout the domain it is continuous and the slope of the curve is always decreasing which would imply that the curve formed has a stationary point which will become the global maxima. This can be analysed by finding the double derivative and showing that it is always negative.

$$f(\theta) = \log(L(\theta)) = \Sigma_{i=1}^{n} \log(N(x_i|\mu, \sigma^2)) = \Sigma_{i=1}^{n} \frac{(x-\mu)^2}{2\sigma^2} + c \tag{1}$$

This function is continuous. Partially double deriving w.r.t μ, σ.

$$\begin{aligned} \frac{\partial f(\theta)}{\partial \mu} &= \frac{1}{\sigma^2} \sum_{i=1}^{n} (x_i - \mu) \\ \frac{\partial^2 f(\theta)}{\partial \mu^2} &= \frac{-n}{\sigma^2} < 0 \end{aligned} \tag{2}$$

$$\begin{aligned} \frac{\partial f(\theta)}{\partial \sigma} &= \frac{2}{\sigma^3} \sum_{i=1}^{n} (x_i - \mu)^2 \\ \frac{\partial^2 f(\theta)}{\partial \sigma^2} &= \frac{-6}{\sigma^4} \sum_{i=1}^{n} (x_i - \mu)^2 < 0 \end{aligned} \tag{3}$$

Therefore, hence proved.
The bias will be as follows:

$$Bias(\theta) = E[\theta] - \theta \tag{4}$$

For μ,

$$\begin{aligned} Bias[\mu] &= E[\Sigma_{i=1}^{n}(x_i/n)] - \mu \\ &= \Sigma_{i=1}^{n}(E[x_i/n]) - \mu \\ &= \mu - \mu = 0 \end{aligned} \tag{5}$$

For σ^2

$$Var[x] = E[x^2] - E[x]^2 \tag{6}$$

$$\begin{aligned} Bias[\sigma^2] &= E[\Sigma_{i=1}^{n}(x_i - \mu)^2/n] - \sigma^2 \\ &= E[\Sigma_{i=1}^{n}(x_i^2 + \mu^2 - 2x_i\mu)/n] - \sigma^2 \\ &= E[\Sigma_{i=1}^{n}(x_i^2)/n] + E[\Sigma_{i=1}^{n}(\mu^2)/n] - 2 * E[\Sigma_{i=1}^{n}(x_i\mu)/n] - \sigma^2 \\ &= E[\Sigma_{i=1}^{n}(x_i^2)/n] + E[\mu^2] - 2 * E[\mu^2] - \sigma^2 \\ &= E[(x_i^2)] - E[\mu^2] - \sigma^2 \\ &= \sigma^2 + E[x_i]^2 - (var(\mu) + E[\mu]^2) - \sigma^2 \\ &= E[\mu]^2 - Var(\mu) - E[\mu]^2 \\ &= -Var[(1/n)\Sigma_{i=1}^{n} x_i] = -\frac{1}{n^2} Var[\Sigma_{i=1}^{n} x_i] = -\frac{n\sigma^2}{n^2} = \frac{-1}{n}\sigma^2 \end{aligned} \tag{7}$$

(c) (2 points) Consider a hyperplane $\mathbb{H}$ in $\mathbb{R}^d$ passing through zero. Prove that $\mathbb{H}$ is a subspace of $\mathbb{R}^d$ and is of dimension $d-1$.

Solution: Since $\mathbb{H}$ is a hyperplane it can be represented in the form of a linear equation of n variables where each variable represents a dimension. This is because, the equation of a hyperplane is of the form

$$w^T y = b, w, y \in \mathbb{R}^d, b \in \mathbb{R} \tag{8}$$

where y is a point in $\mathbb{H}$ Let w be represented by the tuple $(a_1, a_2, ...a_d)^T$ and y arbitrarily be $(x_1, x_2..., x_d)^T$. Then the above equation becomes

$$a_d * x_n + a_{d-1} * x_{d-1} + ... + a_1 * x_1 = b \tag{9}$$

But it is given that $\mathbb{H}$ passes through origin. This implies that when all the values of the dimensions are 0 the equation must be satisfied.

$$\begin{aligned} a_d * 0 + a_{d-1} * 0 + ... + a_1 * 0 = b \\ \implies b = 0 \end{aligned} \tag{10}$$

Therefore the equation now becomes

$$a_d * x_d + a_{d-1} * x_{d-1} + ... + a_1 * x_1 = 0 \tag{11}$$

The points in H can be represented as a tuple of the form:
$\{(x_1, x_2..., x_d) \mid a_d * x_d + a_{d-1} * x_{d-1} + ... + a_1 * x_1 = 0, x_1, x_2...x_n \in \mathbb{R}\}$.
For any tuple in $\mathbb{H}$ it is clear that it belongs to $\mathbb{R}$ because $\mathbb{H}$ is a hyperplane in $\mathbb{R}^d$ and so the dimension values cannot be complex. And thus the tuple values are also consequently always real. Therefore $\mathbb{H}$ is a subset of $\mathbb{R}^d$.
To prove that $\mathbb{H}$ is a vector space, it is necessary to prove that

$$X \in \mathbb{H}, Y \in \mathbb{H} \implies X + Y \in \mathbb{H} \tag{12}$$

$$X \in \mathbb{H} \implies aX \in \mathbb{H}, a \in \mathbb{R} \tag{13}$$

To prove (4), Assume $X = (p_1, p_2...., p_n)$ and $Y = (q_1, q_2...., q_n)$ belong to $\mathbb{H}$. Then,

$$\begin{aligned} X \in \mathbb{H} &\implies a_1 * p_1 + a_2 * p_2....a_d * p_d = 0 \\ Y \in \mathbb{H} &\implies a_1 * q_1 + a_2 * q_2....a_d * q_d = 0 \end{aligned} \tag{14}$$

Adding above equations,

$$\begin{aligned} & a_1 * (p_1 + q_1) + a_2 * (p_2 + q_2)....a_d * (p_d + q_d) = 0 \\ & \implies (p_1 + q_1, p_2 + q_2, ..., p_d + q_d) \in \mathbb{H} \\ & \implies (p_1, p_2..., p_d) + (q_1, q_2..., q_d) \in \mathbb{H} \\ & \implies X + Y \in \mathbb{H} \end{aligned} \tag{15}$$

Similarly to prove (5), multiply first equation in (6) with α.

$$\begin{aligned} & \alpha * (a_1 * p_1 + a_2 * p_2 .... a_d * p_d) = \alpha * 0 \\ \implies & a_1 * (\alpha * p_1) + a_2 * (\alpha * p_2) ..... + a_d * (\alpha * p_d) = 0 \\ \implies & (\alpha * p_1, \alpha * p_2, ..., \alpha * p_d) \in \mathbb{H} \\ \implies & \alpha * (p_1, p_2 ... p_d) \in \mathbb{H} \\ \implies & \alpha * X \in \mathbb{H} \end{aligned} \tag{16}$$

Therefore hence proved that $\mathbb{H}$ is a subset of $\mathbb{R}^d$ and a vector space and so it is a subspace of $\mathbb{R}^d$. To find dimension, we take the initial representation of hyperplane which is $w^T * y = b$. But we have concluded that value of b in $\mathbb{H}$ is 0. Therefore it becomes $w^T * y = 0$ now as not all components of w are zero it is clear that rank of w is 1. Here from the equation it can be observed that y represents the null space of w^T. As dimension of w^T is d, from rank nullity theorem, nullity will be become $d - 1$. Thus dimension of nullspace of w^T is $d - 1$ and so dimension of $\mathbb{H}$ which is the null space is $d - 1$.

(d) (2 points) We saw a mixture of two 1D Gaussians ($N(\mu_1, \sigma_1^2)$ and $N(\mu_2, \sigma_2^2)$) in class with parameters π_1, π_2 for the mixing proportions. Is the likelihood of this model convex or not convex? Give proof to support your view.

Solution:

(e) (2 points) Show that there always exists a solution for the system of equations, $A^T Ax = A^T b$, where $x \in \mathbb{R}^m$, $A \in \mathbb{R}^{n \times m}$ and $b \in \mathbb{R}^n$. Further, show that for some solution x^* of this system of equations, Ax^* is the projection of b onto the column space of A.

Solution: To prove: $A^T Ax = A^T b$ is always consistent
Let $c = A^T b$.

$$c \in Range(A^T) \tag{17}$$

But it is known that

$$Range(A^T) = Range(A^T A) \tag{18}$$

Proof for (11) :

$$\begin{aligned} rank(T) + nullity(T) &= dim(V) (\text{from rank-nullity-dimension theorem}) \\ \implies rank(A^T A) + nullity(A^T A) &= dim(V) \\ rank(A) + nullity(A) &= dim(V) \\ \implies rank(A) &= rank(A^T A) \end{aligned} \tag{19}$$

Similarly,

$$\text{rank}(A^T) = \text{rank}(AA^T) \tag{20}$$

$$\begin{aligned} & & \text{range}(AB) \subseteq \text{range}(A) \\ \implies \text{range}(A^TA) \subseteq \text{range}(A^T) & \implies & \text{rank}(A) = \text{rank}(A^TA) <= \text{rank}(A^T) \\ & & \text{rank}(A^T) = \text{rank}(AA^T) <= \text{rank}(A) \\ & \implies & \text{rank}(A) = \text{rank}(A^T) = \text{rank}(A^TA) = \text{rank}(AA^T) \end{aligned} \tag{21}$$

Since $A^TA : V- > V$ and $A^T : W- > V$ it is clear that $\text{Range}(A^TA) = \text{Range}(A^T)$. Therefore applying this to (10),

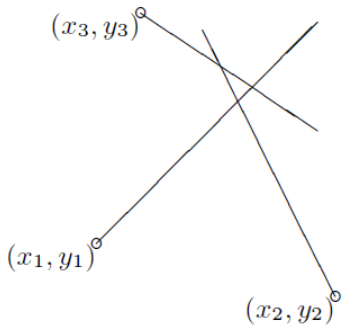
$$c \in \text{Range}(A^T) = \text{Range}(A^TA) \implies \exists x, A^TAx = c \tag{22}$$

Thus $A^TAx = A^Tb$ is always consistent. For a solution x_*,

$$\begin{aligned} A^TAx^* &= A^Tb \\ \implies A^T(Ax^* - b) &= 0 \end{aligned} \tag{23}$$

So $Ax^*-b \in \text{Nullspace}(A^T)$. Which means that Ax^*-b is perpendicular to row space of A^T implies perpendicular to column space of A. This clearly means that Ax^* is the projection of b on column space of A.

2. (5 points) [OF SAILORS AND BEARINGS...] A sailor infers his location (x, y) by measuring the bearings of three buoys whose locations (x_n, y_n) are given on his chart. Let the true bearings of the buoys be θ_n (measured from north as explained here). Assuming that his measurement $\tilde{\theta}_n$ of each bearing is subject to Gaussian noise of small standard deviation σ, what is his inferred location, by maximum likelihood?



The sailor's rule of thumb says that the boat's position can be taken to be the centre of the cocked

hat, the triangle produced by the intersection of the three measured bearings as in the figure shown. Can you persuade him that the maximum likelihood answer is better?

Solution: Our dataset will consist of the points in the cocked hat, and the probability of this can be computed from the values of θ_n. We know that the slope of line can be calculated from 2 points. Using this we can calculate probability of that point from probability of corresponding slopes.

$$\begin{aligned} P(x,y) &= \Pi_{i=1}^{3} P(90 - Tan^{-}(\frac{y-y_i}{x-x_i})) \\ &= \Pi_{i=1}^{3} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(90-Tan^{-}(\frac{y-y_i}{x-x_i})-\theta_i)}{2\sigma^2}} \\ &= \frac{1}{2\pi\sqrt{2\pi}\sigma^3} e^{-\frac{(90-Tan^{-}(\Sigma_{i=1}^{3}\frac{y-y_i}{x-x_i})-\theta_1-\theta_2-\theta_3)}{2\sigma^2}} \end{aligned} \tag{24}$$

We need to find the point with maximum probability and we can do this by partially differentiating log of probability wrt x and y.

$$\begin{aligned} log(P(x,y)) &= -\frac{(90 - Tan^{-}(\Sigma_{i=1}^{3}\frac{y-y_i}{x-x_i}) - \theta_1 - \theta_2 - \theta_3)}{2\sigma^2} - log(2\pi\sqrt{2\pi}\sigma^3) \\ \frac{\partial log(P(x,y)}{\partial x} &= -\frac{1}{2\sigma^2}(\Sigma_{i=1}^{3}\frac{1}{1+(\frac{y-y_i}{x-x_i})^2} * \frac{y_i - y}{(x-x_i)^2}) = 0 \\ &\implies \Sigma_{i=1}^{3}\frac{y_i - y}{(x-x_i)^2 + (y-y_i)^2} = 0 \\ \frac{\partial log(P(x,y)}{\partial y} &= -\frac{1}{2\sigma^2}(\Sigma_{i=1}^{3}\frac{1}{1+(\frac{y-y_i}{x-x_i})^2} * \frac{1}{(x_i - x)}) = 0 \\ &\implies \Sigma_{i=1}^{3}\frac{x_i - x}{(x-x_i)^2 + (y-y_i)^2} = 0 \end{aligned} \tag{25}$$

Solving those 2 equations we can get the value of x,y which are optimal. But doing this is not easy as these are multi variate polynomial equations. So instead of this we can model the points in the cocked hat into a gaussian distribution since the probability is similar to that. The expected value of a gaussian distribution is the mean μ. The maximum likelihood estimate for μ can be found using some of the data points in the cocked hat picked uniformly to represent the triangle and used to calculate the bearing. This will be more optimal than directly taking the centroid.

3. (5 points) [REVEREND BAYES DECIDES]

   (a) (2 points) Consider a classification problem in which the loss incurred on mis-classifying an input vector from class C_k as C_j is given by loss matrix entry L_{kj}, and for which the loss

incurred in selecting the reject option is ψ. Find the decision criterion that will give minimum expected loss, and then simplify it for the case of 0-1 loss (i.e., when $L_{kj} = 1 - I_{kj}$, with I_{kj} being 1 for $k = j$ and 0 otherwise).

Solution: Generally, selecting a class is based on picking the class that has least expected loss value. But here we have an additional constraint saying that if the reject option is picked then an additional loss of ψ is incurred. Taking this into account our new constraint will be that the class which has the least total expected loss will be picked. This can be written as:

$$h(x) = argmin_j((\Sigma_{C_t \in \mathbb{R}} L_{tj} * P(C_t \mid x)) + \psi * 1_{\exists i, P(C_j|x) < P(C_i|x)})$$

On simplifying it to the 0-1 case by substituting loss matrix values we get

$$h(x) = argmin_j((\Sigma_{t \neq j, C_t \in \mathbb{R}} P(C_t \mid x)) + \psi * 1_{\exists i, P(C_j|x) < P(C_i|x)})$$

(b) (2 points) Let L be the loss matrix defined by $L = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{bmatrix}$ where L_{ij} indicates the loss for an input x with i being the true class and j the predicted class. All the three classes are equally likely to occur. The class densities are $P(x|C_1 = 1) \sim N(-2, 1)$, $P(x|C_2 = 2) \sim N(0, 1)$ and $P(x|C_3) \sim N(2, 1)$. Find the Bayes classifier $h(x)$.

Solution: Given that all 3 classes are equally likely, therefore

$$P(C_1) = P(C_2) = P(C_3) = 1/3 \tag{26}$$

$h(x)$ is given by

$$h(x) = argmin_i(\Sigma_{C_j \in \mathbb{R}} L_{ij} * P(C_j \mid x) \tag{27}$$

But from Baye's theorem we know that,

$$P(C_j \mid x) \propto P(x \mid C_j) * P(C_j) \tag{28}$$

From (10) we know that $P(C_j)$ are equal for all j. Therefore,

$$P(C_j \mid x) \propto P(x \mid C_j) \tag{29}$$

Thus,

$$\begin{aligned} &\Sigma_{C_j \in \mathbb{R}} L_{ij} * P(C_j \mid x) \propto \Sigma_{C_j \in \mathbb{R}} L_{ij} * P(x \mid C_j) \\ \implies &argmin_i(\Sigma_{C_j \in \mathbb{R}} L_{ij} * P(C_j \mid x) = argmin_i(\Sigma_{C_j \in \mathbb{R}} L_{ij} * P(x \mid C_j) \end{aligned} \tag{30}$$

Now substituting values of $P(x \mid C_j)$ and L as given in question:

$$\begin{aligned} h(x) &= argmin_i(\Sigma_{C_j \in \mathbb{R}} L_{ij} * P(x \mid C_j) \\ &= argmin_i(0 + 1 * N(0,1) + 2 * N(2,1), \\ &\quad 1 * N(-2,1) + 0 + 1 * N(2,1), 2 * N(-2,1) + 1 * N(0,1) + 0) \end{aligned} \tag{31}$$

$$h(x) = argmin_i(1 * N(0,1) + 2 * N(2,1), 1 * N(-2,1) + 1 * N(2,1), 2 * N(-2,1) + 1 * N(0,1)) \tag{32}$$

For $h(x) = C_1$ x value should be such that first expression is greater than the other 2. Applying this condition on first and third expressions,

$$\begin{aligned} &\frac{1}{(2\pi)^{1/2}} * e^{-}(1/2)(x)^2 + 2 * \frac{1}{(2\pi)^{1/2}} * e^{-}(1/2)(x-2)^2 \\ &> 2 * \frac{1}{(2\pi)^{1/2}} * e^{-}(1/2)(x+2)^2 + \frac{1}{(2\pi)^{1/2}} * e^{-}(1/2)(x)^2 \end{aligned} \tag{33}$$

$$\implies e^{-}(1/2)(x-2)^2 > e^{-}(1/2)(x+2)^2$$

$$\implies (x+2)^2 > (x-2)^2 \tag{34}$$

$$\implies x > 0$$

$$\begin{aligned} &\frac{1}{(2\pi)^{1/2}} * e^{-}(1/2)(x)^2 + 2 * \frac{1}{(2\pi)^{1/2}} * e^{-}(1/2)(x-2)^2 \\ &> 1 * \frac{1}{(2\pi)^{1/2}} * e^{-}(1/2)(x+2)^2 + \frac{1}{(2\pi)^{1/2}} * e^{-}(1/2)(x-2)^2 \end{aligned} \tag{35}$$

$$\implies e^{-}(1/2)x^2 + e^{-}(1/2)(x-2)^2 > e^{-}(1/2)(x+2)^2 \tag{36}$$

The solution set of above 2 equations will be $(-\infty, (1/2)\ln((-e^2 + \sqrt{e^4+4})/2)$
Similarly, for $h(x) = C_3$,

$$\begin{aligned} &\frac{1}{(2\pi)^{1/2}} * e^{-}(1/2)(x)^2 + 2 * \frac{1}{(2\pi)^{1/2}} * e^{-}(1/2)(x-2)^2 \\ &< 2 * \frac{1}{(2\pi)^{1/2}} * e^{-}(1/2)(x+2)^2 + \frac{1}{(2\pi)^{1/2}} * e^{-}(1/2)(x)^2 \end{aligned} \tag{37}$$

$$\implies e^{-}(1/2)(x-2)^2 < e^{-}(1/2)(x+2)^2$$

$$\implies (x+2)^2 < (x-2)^2 \tag{38}$$

$$\implies x < 0$$

$$\begin{aligned} &\frac{1}{(2\pi)^{1/2}} * e^{-}(1/2)(x)^2 + 2 * \frac{1}{(2\pi)^{1/2}} * e^{-}(1/2)(x+2)^2 \\ &> 1 * \frac{1}{(2\pi)^{1/2}} * e^{-}(1/2)(x+2)^2 + \frac{1}{(2\pi)^{1/2}} * e^{-}(1/2)(x-2)^2 \end{aligned} \tag{39}$$

$$\implies e^{-}(1/2)x^2 + e^{-}(1/2)(x+2)^2 > e^{-}(1/2)(x-2)^2 \tag{40}$$

Here too similar to above, the intersection of both equations will be $((1/2)\ln((e^2+\sqrt{e^4+4})/2, \infty)$. For $h(x) = C_2$,

$$\begin{aligned} &\frac{1}{(2\pi)^{1/2}} * e^{-}(1/2)(x)^2 + 2 * \frac{1}{(2\pi)^{1/2}} * e^{-}(1/2)(x-2)^2 \\ &> 1 * \frac{1}{(2\pi)^{1/2}} * e^{-}(1/2)(x+2)^2 + \frac{1}{(2\pi)^{1/2}} * e^{-}(1/2)(x-2)^2 \end{aligned} \tag{41}$$

$$\implies e^{-}(1/2)x^2 + e^{-}(1/2)(x-2)^2 > e^{-}(1/2)(x+2)^2 \tag{42}$$

$$\begin{aligned} &\frac{1}{(2\pi)^{1/2}} * e^{-}(1/2)(x)^2 + 2 * \frac{1}{(2\pi)^{1/2}} * e^{-}(1/2)(x+2)^2 \\ &> 1 * \frac{1}{(2\pi)^{1/2}} * e^{-}(1/2)(x+2)^2 + \frac{1}{(2\pi)^{1/2}} * e^{-}(1/2)(x-2)^2 \end{aligned} \tag{43}$$

$$\implies e^{-}(1/2)x^2 + e^{-}(1/2)(x+2)^2 > e^{-}(1/2)(x-2)^2 \tag{44}$$

Solving above equations we get $x \in ((1/2)\ln((-e^2 + \sqrt{e^4+4})/2, (1/2)\ln((e^2 + \sqrt{e^4+4})/2)$
Thus the Bayes classifier is given by

$$\begin{aligned} h(x) =& C_1, x < (1/2)\ln((-e^2 + \sqrt{e^4+4})/2 \\ & C_2, (1/2)\ln((-e^2 + \sqrt{e^4+4})/2 <= x <= (1/2)\ln((e^2 + \sqrt{e^4+4})/2 \\ & C_3, x > (1/2)\ln((e^2 + \sqrt{e^4+4})/2 \end{aligned}$$

(c) (1 point) Consider two classes C_1 and C_2 with equal priors and with class conditional densities of a feature x given by Gaussian distributions with respective means μ_1 and μ_2, and same variance σ^2. Find equation of the decision boundary between these two classes.

Solution: Given that C_1 and C_2 have equal priors. Therefore

$$P(C_1) = P(C_2) \tag{45}$$

The decision boundary is given by the solution set of x where posteriors of both classes are equal.

$$\begin{aligned} P(C_1 \mid x) &= P(C_2 \mid x) \\ \frac{P(x \mid C_1) * P(C_1)}{P(x)} &= \frac{P(x \mid C_2) * P(C_2)}{P(x)} \\ P(x \mid C_1) * P(C_1) &= P(x \mid C_2) * P(C_2) \end{aligned}$$

Applying equation (30) and values given in the question,

$$
\begin{aligned}
P(x \mid C_1) &= P(x \mid C_2) \\
\implies \frac{1}{(2\pi)^{1/2} * \sigma} * e^{-}(1/(2 * \sigma^2))(x - \mu_1)^2 &= \frac{1}{(2\pi)^{1/(2*\sigma^2)} * \sigma} * e^{-}(1/2)(x - \mu_2)^2 \\
\implies e^{-}(1/(2 * \sigma^2))(x - \mu_1)^2 &= e^{-}(1/(2 * \sigma^2))(x - \mu_2)^2 \\
\implies (x - \mu_1)^2 &= (x - \mu_2)^2 \\
\implies x &= (\mu_1 + \mu_2)/2
\end{aligned}
$$

4. (10 points) [DON'T MIX YOUR WORDS!]
Consider two documents D_1, D_2 and a background language model given by a Categorical distribution (i.e., assume $P(w|\theta)$ is known for every word w in the vocabulary V). We use the maximum likelihood method to estimate a unigram language model based on D_1, which will be denoted by θ_1 (i.e, $p(w \mid \theta_1)$ = "nos. of times word w occurred in D_1"/ $\mid D_1 \mid$, where $\mid D_1 \mid$ denotes the total number of words in D_1). Assume document D_2 is generated by sampling words from a two-component Categorical mixture model where one component is $p(w|\theta_1)$ and the other is $p(w|\theta)$. Let λ denote the probability that D_1 would be selected to generate a word in D_2. That makes $1 - \lambda$ the probability of selecting the background model. Let $D_2 = (w_1, w_2, ...w_k)$, where w_i is a word from the vocabulary V. Use the mixture model to fit D_2 and compute the ML estimate of λ using the EM (Expectation-Maximization) algorithm.

   (a) (2 points) Given that each word w_i in document D_2 is generated independently from the mixture model, write down the log-likelihood of the whole document D_2. Is it easy to maximize this log-likelihood?

Solution:

$$P(w \mid \theta_2) = \lambda * P(w \mid \theta_1) + (1 - \lambda) * P(w \mid \theta) \tag{46}$$

Therefore from the words in D_2 likelihood and consequently log likelihood would be,

$$
\begin{aligned}
L(\theta_2) &= \Pi_{i=1}^{k}(\lambda * P(w_i \mid \theta_1) + (1 - \lambda) * P(w_i \mid \theta)) \\
log(L(\theta_2)) &= \Sigma_{i=1}^{k}(log(\lambda * P(w_i \mid \theta_1) + (1 - \lambda) * P(w_i \mid \theta)))
\end{aligned} \tag{47}
$$

It is not easy to maximise the log likelihood because it is a log of a sum which is not easy to manipulate and analyse. The equations are also coupled.

   (b) (4 points) Write down the E-step and M-step updating formulas for estimating λ. Show your derivation of these formulas.

Solution:
E-step: For all $1 <= i <= k$, calculating $P(z = k \mid w_i; \theta_t)$ to estimate λ as it is given that θ, θ_1 are known.

$$\begin{aligned} P(z_i \mid x; \theta) &= \frac{P(z_i, x; \theta)}{\Sigma_{j=1}^{k} P(z_j, x; \theta)} \\ &= \frac{\pi_i P(x \mid \theta)}{\Sigma_{j=1}^{k} \pi_j P(x \mid \theta)} \end{aligned} \tag{48}$$

$$\begin{aligned} P(z = 0 \mid w_i; \lambda_t) &= \frac{\lambda_t P(w_i \mid \theta_1)}{\lambda_t P(w_i \mid \theta_1) + (1 - \lambda_t) P(w_i \mid \theta)} \\ P(z = 1 \mid w_i; \lambda_t) &= \frac{(1 - \lambda_t) P(w_i \mid \theta)}{\lambda_t P(w_i \mid \theta_1 + (1 - \lambda_t) P(w_i \mid \theta)} \end{aligned} \tag{49}$$

M-step: Maximising ELBO over θ and setting it as θ_{t+1}

$$\begin{aligned} \theta_{t+1} &= argmax_\theta(g(\theta)) \\ &= argmax_\theta(\Sigma_{x \in D} \Sigma_z P(z \mid x; \theta_t) * \log(P(x, z; \theta))) \end{aligned} \tag{50}$$

$$\begin{aligned} g(\lambda) = \Sigma_{i=1}^{k} (& (\frac{\lambda_t P(w_i \mid \theta_1)}{\lambda_t P(w_i \mid \theta_1) + (1 - \lambda_t) P(w_i \mid \theta)}) * \log(\lambda P(w_i \mid \theta_1))) \\ & + \frac{(1 - \lambda_t) P(w_i \mid \theta))}{\lambda_t P(w_i \mid \theta_1) + (1 - \lambda_t) * P(w_i \mid \theta)}) * \log((1 - \lambda) P(w_i \mid \theta)))) \end{aligned} \tag{51}$$

To get argmax of $g(\lambda)$ we need to differentiate w.r.t λ and make equal to 0.

$$\begin{aligned} \frac{\partial g(\lambda)}{\partial \lambda} = \Sigma_{i=1}^{k} (& \frac{1}{(\lambda_t P(w_i \mid \theta_1) + (1 - \lambda_t) P(w_i \mid \theta))}) * ((\lambda_t P(w_i \mid \theta_1)) * 1/\lambda \\ & - (1 - \lambda_t) P(w_i \mid \theta)) * 1/(1 - \lambda)) = 0 \end{aligned} \tag{52}$$

$$\implies \frac{\lambda - 1}{\lambda} = \frac{\Sigma_{i=1}^{k} (\frac{(1 - _t) P(w_i \mid \theta)}{(\lambda_t P(w_i \mid \theta_1) + (1 - \lambda_t) P(w_i \mid \theta))})}{\Sigma_{i=1}^{k} (\frac{\lambda_t P(w_i \mid \theta_1}{(\lambda_t P(w_i \mid \theta_1) + (1 - \lambda_t) P(w_i \mid \theta))})} \tag{53}$$

$$\begin{aligned} \implies \lambda &= \frac{\Sigma_{i=1}^{k} (\frac{\lambda_t P(w_i \mid \theta_1)}{(\lambda_t P(w_i \mid \theta_1) + (1 - \lambda_t) P(w_i \mid \theta))})}{\Sigma_{i=1}^{k} (\frac{\lambda_t P(w_i \mid \theta_1 + (1 - _t) P(w_i \mid \theta)}{(\lambda_t P(w_i \mid \theta_1) + (1 - \lambda_t) P(w_i \mid \theta))})} \\ &= \frac{\Sigma_{i=1}^{k} (\frac{\lambda_t P(w_i \mid \theta_1)}{(\lambda_t P(w_i \mid \theta_1) + (1 - \lambda_t) P(w_i \mid \theta))})}{\Sigma_{i=1}^{k} (1)} \\ &= \frac{1}{k} * \Sigma_{i=1}^{k} (\frac{\lambda_t P(w_i \mid \theta_1)}{(\lambda_t P(w_i \mid \theta_1) + (1 - \lambda_t) P(w_i \mid \theta))}) \end{aligned} \tag{54}$$

$$\implies \lambda_{t+1} = \frac{1}{k} * \Sigma_{i=1}^{k} (\frac{\lambda_t P(w_i \mid \theta_1)}{(\lambda_t P(w_i \mid \theta_1) + (1 - \lambda_t) P(w_i \mid \theta))}) \tag{55}$$

(c) (4 points) In the previous parts of the question, we assume that the background language model $P(w|\theta)$ is known. How will your E-step and M-step change if you do not know the parameter θ and only know θ_1? Show your derivation.

Solution: In above case including θ as well in estimation we get,
E-step:

$$\begin{aligned} P(z_i \mid x;\theta) &= \frac{P(z_i, x;\theta)}{\Sigma_{j=1}^{k} P(z_j, x;\theta)} \\ &= \frac{\pi_i P(x \mid \theta)}{\Sigma_{j=1}^{k} \pi_j P(x \mid \theta)} \end{aligned} \tag{56}$$

$$\begin{aligned} P(z=0 \mid w_i;\lambda_t,\theta_t) &= \frac{\lambda_t P(w_i \mid \theta_1)}{\lambda_t P(w_i \mid \theta_1) + (1-\lambda_t) P(w_i \mid \theta_t)} \\ P(z=1 \mid w_i;\lambda_t,\theta_t) &= \frac{(1-\lambda_t) P(w_i \mid \theta_t)}{\lambda_t P(w_i \mid \theta_1 + (1-\lambda_t) P(w_i \mid \theta_t)} \end{aligned} \tag{57}$$

M-step:

$$\begin{aligned} \theta_{t+1} &= argmax_\theta(g(\theta)) \\ &= argmax_\theta(\Sigma_{x \in D} \Sigma_z P(z \mid x;\theta_t) * \log(P(x,z;\theta))) \end{aligned} \tag{58}$$

$$\begin{aligned} g(\lambda,\theta) = \Sigma_{i=1}^{k} (& (\frac{\lambda_t P(w_i \mid \theta_1)}{\lambda_t P(w_i \mid \theta_1) + (1-\lambda_t) P(w_i \mid \theta_t)}) * \log(\lambda P(w_i \mid \theta_1))) \\ & + \frac{(1-\lambda_t) P(w_i \mid \theta_t))}{\lambda_t P(w_i \mid \theta_1) + (1-\lambda_t) * P(w_i \mid \theta_t)}) * \log((1-\lambda) P(w_i \mid \theta)))) \end{aligned} \tag{59}$$

To get argmax of $g(\lambda,\theta)$ we need to partially differentiate w.r.t λ and θ and make equal to 0. From above for we get

$$\lambda = \frac{1}{k} * \Sigma_{i=1}^{k} (\frac{\lambda_t P(w_i \mid \theta_1)}{(\lambda_t P(w_i \mid \theta_1) + (1-\lambda_t) P(w_i \mid \theta_t))}) \tag{60}$$

$$\frac{\partial \lambda}{\partial \theta} = 0 \tag{61}$$

Substituting this in $g(\lambda,\theta)$ to get optimal θ,

$$\begin{aligned} \frac{\partial g(\lambda,\theta)}{\partial \theta} = \Sigma_{i=1}^{k} 0 + \frac{(1-\lambda_t) P(w_i \mid \theta_t))}{\lambda_t P(w_i \mid \theta_1) + (1-\lambda_t) * P(w_i \mid \theta_t)}) * \frac{1}{P(w_i \mid \theta)} * \frac{\partial P(w_i \mid \theta)}{\partial \theta} = 0 \\ \Sigma_{i=1}^{k} \frac{(1-\lambda_t) P(w_i \mid \theta_t))}{\lambda_t P(w_i \mid \theta_1) + (1-\lambda_t) * P(w_i \mid \theta_t)}) * \frac{1}{P(w_i \mid \theta)} * \frac{\partial P(w_i \mid \theta)}{\partial \theta} = 0 \end{aligned} \tag{62}$$

Therefore the solution to the above equation will the estimate for θ.

(d) (3 points) [BONUS] The previous parts of the question deal with MLE based density estimation. If you were to employ a Bayesian estimation method to infer λ, how will you proceed? That is, what prior would you choose for λ, and what is the the formula for the posterior? Is this posterior easily computable (i.e., has a closed-form expression or can be computed efficiently)? You can assume that both $P(w|\theta_1)$ and $P(w|\theta)$ are known and only λ is not known.

Solution:

5. (10 points) [DENSITY ESTIMATION - THE ONE RING TO RULE THEM ALL!] With density estimation ring already in your finger, you have all you need to master simple linear regression (even before seeing regression formally in class). Simple linear regression is a model that assumes a linear relationship between an input (aka independent) variable x and an output (aka dependent) variable y. Let us assume that the available set of observations, $\mathbb{D} = \{x_i, y_i\}_{i=1}^n$, are iid samples from the following model that captures the relationship between y and x:

$$y_i = w_0 + w_1 x_i + \epsilon_i; \qquad \epsilon_i \sim \mathcal{N}(0, \sigma^2)$$

In this model, note that x_i is not a random variable, whereas ϵ_i and hence y_i are random variables, with ϵ_i being modeled as a Gaussian noise that is independent of each other and doesn't depend on x_i. Value of σ is assumed to be known for simplicity.
We would like to learn the parameters $\theta = \{w_0, w_1\}$ of the model, i.e., we would like to use MLE to estimate the exact parameter values or Bayesian methods to infer the (posterior) probability distribution over the parameter values.

(a) (2 points) Compute the probability distribution $P(y_i|x_i, \theta)$, and use it to write down the log likelihood of the model.

Solution: y can be modelled as a gaussian as ϵ is gaussian. Thus parameters will be μ, σ^2.

$$\begin{aligned} \mu = E[y_i] &= E[w_0] + E[w_1 x_i] + E[\epsilon_i] \\ &= w_0 + w_1 x_i + 0 = w_0 + w_1 x_i \\ \sigma^2[y_i] &= Variance[y_i] = Variance[\epsilon_i] = \sigma^2 \end{aligned} \tag{63}$$

$$P(y_i \mid x_i, \theta) = N(\mu, \sigma^2) = N(w_0 + w_1 x_i, \sigma^2) \tag{64}$$

The log likelihood can be found by directly putting values of μ and σ^2 in log likelihood result of a gaussian distribution. It will become

$$\begin{aligned} log(L(\theta)) &= -N * log(\sqrt{2\pi}\sigma) - \Sigma_n (y_i - \mu)^2/(2\sigma^2) \\ &= -N * log(\sqrt{2\pi}\sigma) - \Sigma_n (y_i - w_0 - w_1 x_i)^2/(2\sigma^2) \end{aligned} \tag{65}$$

(b) (3 points) Derive the ML estimates for w_0 and w_1 by optimizing the above log likelihood.

Solution: Differentiating log likelihood wrt w_0, w_1 to get ML estimates.

$$\begin{aligned} \frac{\partial L(\theta)}{\partial w_0} &= 0 + \Sigma_n 2 * (y_i - w_0 - w_1 x_i)/(2\sigma^2) = 0 \\ &\implies \Sigma_n (y_i - w_0 - w_1 x_i) = 0 \\ \implies w_0 &= \frac{\Sigma_n y_i - w_1 x_i}{n} = (\Sigma_n y_i)/n - w_1 * (\Sigma_n x_i)/n \end{aligned} \tag{66}$$

$$\begin{aligned} \frac{\partial L(\theta)}{\partial w_1} &= 0 + \Sigma_n 2(x_i - (\Sigma_n x_i)/n) * (y_i - w_0 - w_1 x_i)/(2\sigma^2) = 0 \\ \implies \Sigma_n (x_i - (\Sigma_n x_i)/n)(y_i - (\Sigma_n y_i)/n - w_1 * (\Sigma_n x_i)/n - w_1 x_i) &= 0 \\ \implies w_1 &= \frac{\Sigma_n (y_i - (\Sigma_n y_i)/n)(x_i - (\Sigma_n x_i)/n)}{\Sigma_n (x_i - (\Sigma_n x_i)/n)^2} \\ \implies w_0 = (\Sigma_n y_i)/n - (&\frac{\Sigma_n (y_i - (\Sigma_n y_i)/n)(x_i - (\Sigma_n x_i)/n)}{\Sigma_n (x_i - (\Sigma_n x_i)/n)^2}) * (\Sigma_n x_i)/n \end{aligned} \tag{67}$$

(c) (2 points) If σ is also not known before, derive the ML estimate for σ.

Solution: Differentiating log likelihood wrt σ to get ML estimate.

$$\begin{aligned} \frac{\partial L(\theta)}{\partial \sigma} &= -N/\sigma + \Sigma_n (y_i - w_0 - w_1 x_i)^2/(\sigma^3) = 0 \\ &\implies \Sigma_n (y_i - w_0 - w_1 x_i)^2/(\sigma^3) = N/\sigma \\ &\implies \sigma = \sqrt{\frac{\Sigma_n (y_i - w_0 - w_1 x_i)^2}{N}} \end{aligned} \tag{68}$$

(d) (3 points) For Bayesian inference, assume that the parameters w_0, w_1 are independent of each other and follow the distributions $\mathcal{N}(\mu_0, \sigma_0^2)$ and $\mathcal{N}(\mu_1, \sigma_1^2)$ respectively. Compute the posterior distributions for each parameter. How does the mode of this posterior (i.e., MAP estimate) relate to the MLE of w_0 and w_1 derived above?

Solution: We need the posterior which is $P(w_0, w_1 \mid x, y)$.Since x does not vary, $P(x, y \mid w_0, w_1)$ can be replaced by $P(y \mid x, w_0, w_1)$ and this is known from above.

$$\begin{aligned} P(w_0, w_1 \mid x, y) &\propto P(x, y \mid w_0, w_1) * P(w_0, w_1) \\ &\propto P(y \mid x, w_0, w_1) * P(w_0, w_1) \\ P(w_0, w_1) &= P(w_0) * P(w_1) = N(\mu_0, \sigma_0^2) * N(\mu_1, \sigma_1^2) \\ P(y \mid x, w_0, w_1) &= N(\mu, \sigma^2) = N(w_0 + w_1 x_i, \sigma^2) \end{aligned} \tag{69}$$

$$\log(L(P(y \mid x, w_0, w_1))) = -N * \log(\sqrt{2\pi}\sigma) - \Sigma_n (y_i - w_0 - w_1 x_i)^2/(2\sigma^2)$$

To find argmax of posterior we can partially differentiate log of the posterior wrt w_0, w_1 after substituting above values appropriately and equating to 0. This would become,

$$\log(P(w_0, w_1 \mid x, y)) \propto (-\Sigma_{x_i,y_i}(y_i - w_0 - w_1 x_i)^2/(2\sigma^2) - n\log(\sqrt{2\pi}\sigma)) - (w_0 - \mu_0)^2/(2\sigma_0^2) \\ -\log(\sigma_0) - (w_1 - \mu_1)^2/(2\sigma_1^2) - \log(\sigma_1) - \log(2\pi) \tag{70}$$

We can remove terms not containing $_{0,1}$ as these will become 0 on differentiating anyway.

$$\log(P(w_0, w_1 \mid x, y)) \propto -(\Sigma_{x_i,y_i}(y_i - w_0 - w_1 x_i)^2/(2\sigma^2)) - (w_0 - \mu_0)^2/(2\sigma_0^2) - (w_1 - \mu_1)^2/(2\sigma_1^2) \tag{71}$$

Hence MAP is given by,

$$argmax_{w_0,w_1}(-(\Sigma_{x_i,y_i}(y_i - w_0 - w_1 x_i)^2/(2\sigma^2)) - (w_0 - \mu_0)^2/(2\sigma_0^2) - (w_1 - \mu_1)^2/(2\sigma_1^2)) \tag{72}$$

Partial Differentiating wrt w_0,

$$\begin{aligned} \Sigma_{x_i,y_i} \frac{y_i - w_1 x_i}{\sigma^2} - \frac{n w_0}{\sigma^2} &= \frac{w_0 - \mu_0}{\sigma_0^2} \\ \implies w_0 = ((\Sigma y_i - w_1 * x_i) + \frac{\sigma^2}{\sigma_0^2}\mu_0)/(n + \frac{\sigma^2}{\sigma_0^2}\mu_0) & \end{aligned} \tag{73}$$

Partial Differentiating wrt w_1,

$$\begin{aligned} \Sigma_{x_i,y_i} \frac{y_i - w_1 x_i}{\sigma^2} - \frac{n w_0}{\sigma^2} &= \frac{w_0 - \mu_0}{\sigma_0^2} \\ \implies w_0 = ((\Sigma y_i - w_1 * x_i) + \frac{\sigma^2}{\sigma_0^2}\mu_0)/(n + \frac{\sigma^2}{\sigma_0^2}\mu_0) & \end{aligned} \tag{74}$$

In the above mle, w_1, w_2 donot vary. Thus in this equation if σ_0 and σ_1 tend to infinity then that implies there is no variation. When that is applied in above equation we see that it transforms back into the linear regression we started with.

6. (10 points) [LET'S ROLL UP YOUR CODING SLEEVES...] **Learning Binary Bayes Classifiers from data via Density Estimation**
   Derive Bayes classifiers under assumptions below and employing maximum likelihood approach to estimate class prior/conditional densities, and return the results on a test set.

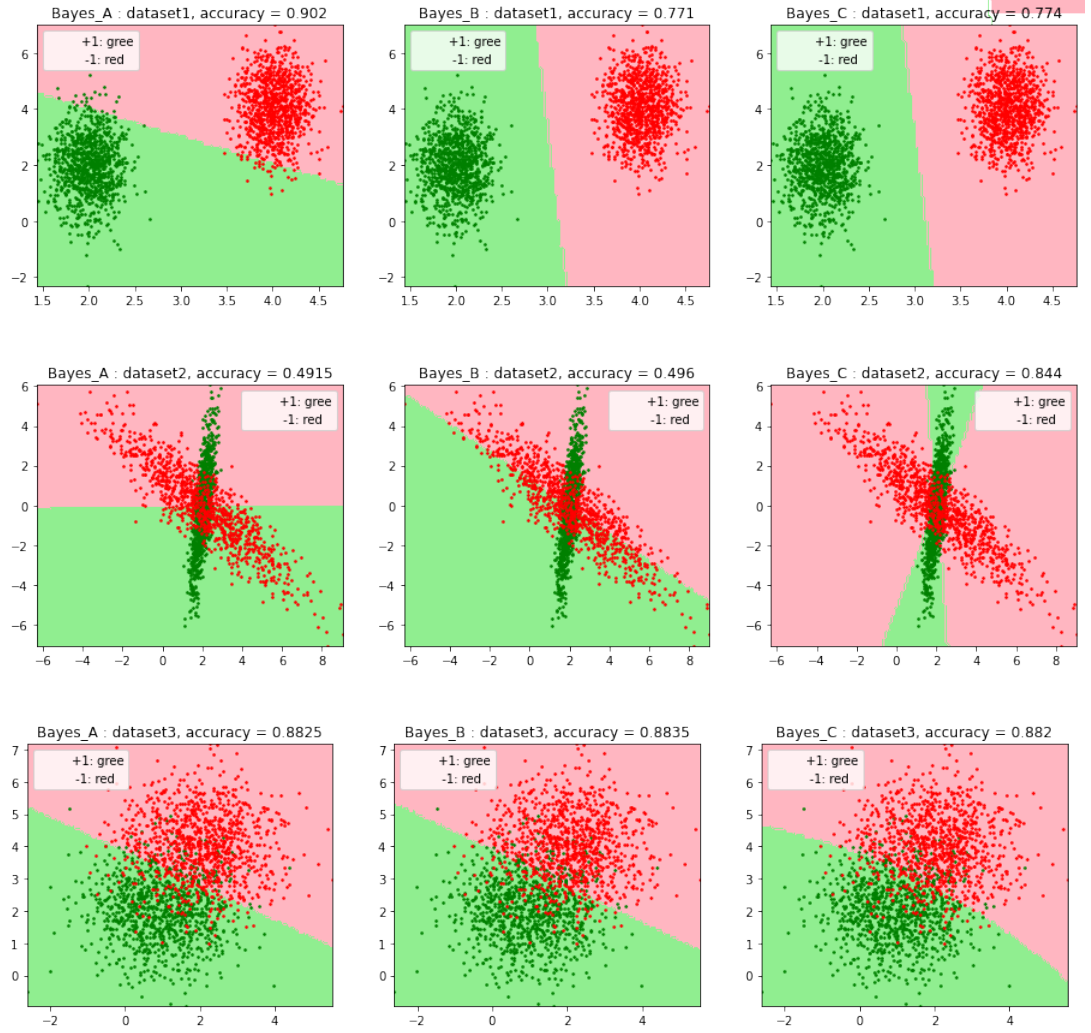
   1. **BayesA** Assume $X|Y = -1 \sim \mathcal{N}(\mu_-, I)$ and $X|Y = 1 \sim \mathcal{N}(\mu_+, I)$
   2. **BayesB** Assume $X|Y = -1 \sim \mathcal{N}(\mu_-, \Sigma)$ and $X|Y = 1 \sim \mathcal{N}(\mu_+, \Sigma)$
   3. **BayesC** Assume $X|Y = -1 \sim \mathcal{N}(\mu_-, \Sigma_-)$ and $X|Y = 1 \sim \mathcal{N}(\mu_+, \Sigma_+)$

Please see this folder for the template .ipynb file containing the helper functions, and you've to add the missing code to this file (specifically, three functions function_for_A, function_for_B and function_for_C, and associated plotting/ROC code snippets) to implement the above three algorithms for the three datasets given in the same folder.

Please provide your results/answers in the pdf file you upload to GradeScope, but please submit your code separately in this moodle link. The code submitted should be a rollno.zip file containing two files: rollno.ipynb file (including your code as well as the exact same results/plots uploaded to Gradescope) and the associated rollno.py file.

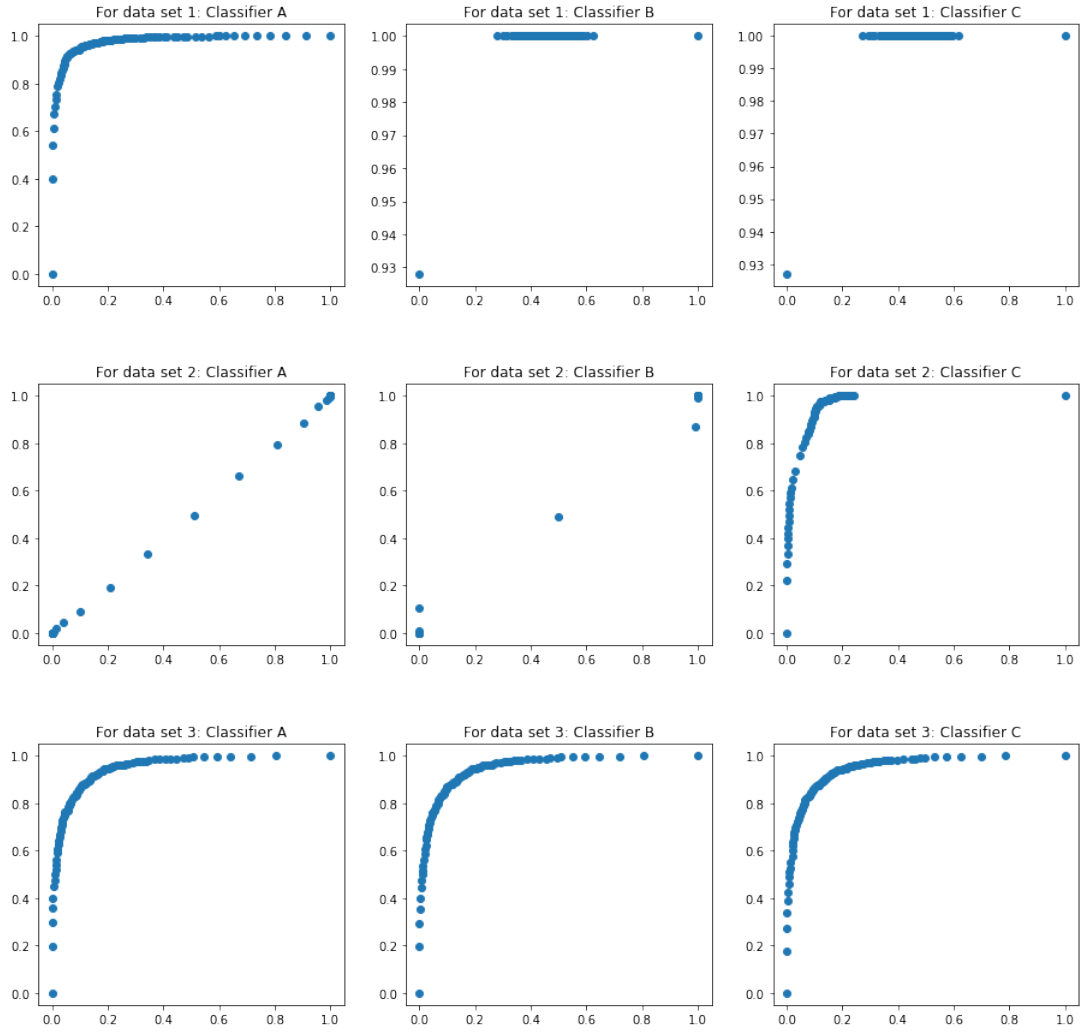
(a) (3 points) Plot all the classifiers (3 classification algorithms on 3 datasets = 9 plots) on a 2D plot, Add the training data points also on the plots. (Color the positively classified area light green, and negatively classified area light red as in Fig 4.5 in Bishop's book).

Solution:



(b) (3 points) Give the ROC curves for all the classifiers. Note that a ROC curve plots the FPR (False Positive Rate) on the x-axis and TPR (True Positive Rate) on the y-axis. (9 plots)

Solution:



(c) (2 points) Provide the error rates for the above classifiers (three classifiers on the three datasets as 3×3 table, with appropriately named rows and columns).

Solution:

Error_rate	BayesA	BayesB	BayesC
datasetA	0.098	0.1245	0.1245
datasetB	0.5085	0.504	0.0745
datasetC	0.1175	0.1185	0.118

(d) (2 points) Summarise and explain your observations based on your plots and the assumptions given in the problem. Also briefly comment whether a non-parametric density estimation approach could have been used to solve this problem, and if so, what the associated pros/cons are compared to the parametric MLE based approach you have implemented.

Solution: From observation it is noticeable that of the given models BayesC is better in general as it is able to classify non linear separation as well. In the first data set we see that BayesB and BayesC are completely separated and area under roc curve is also higher but they cant be the better classifiers since accuracy is lower than that of BayesA. Clearly for this dataset the training data does not fully represent the test data. For data set 2 the data is not linearly separable and this recognised only by BayesC. From the roc curves as well it is observed that for BayesA and BayesB the curve is completely different and not how it ought to be. The data set 3 is not well classified on average by any of the models and the roc curves also have comparable areas under them so all classifiers are equally faulty.

Here all the data is in the form of clusters even if those clusters may not be linearly separable. A suitable model would be clustering which can be done in both parametric and non-parametric density estimation. Although non-parametric can be used as the can model much more complex distributions it requires a storing and computing for entire data set. However as the datasets here are not too complex I believe parametric clustering is sufficient.