CSCI 5521: Introduction to Machine Learning (Spring 2025)[1]

Homework 1

Due date: Feb 19, 2025 11:59pm

1. (**30 points**) Find the Maximum Likelihood Estimation (MLE) of $\boldsymbol{\theta}$ in the following probabilistic density functions. In each case, consider a random sample of size n. Show your calculation:

   (a) $f(x|\theta) = \frac{x}{\theta^2}\exp\{\frac{-x^2}{\theta^2}\}, x \geq 0, \theta \neq 0$

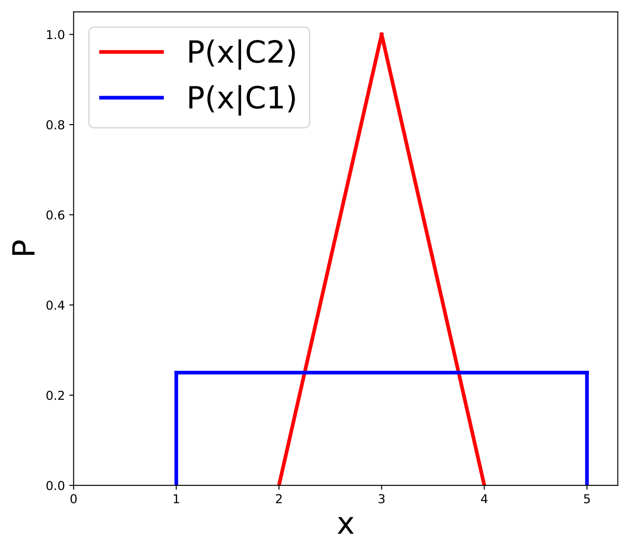
   (b) $f(x|\alpha, \theta) = \alpha\theta^{-\alpha}x^{-\theta-1}, x \geq 0, \alpha > 0, \theta > 0$

   (c) $f(x|\theta) = \frac{x}{\theta}, 0 \leq x \leq \theta, \theta > 0$ (Hint: You can draw the likelihood function)

2. (**30 points**) We want to build a pattern classifier with continuous attribute using Bayes' Theorem. The object to be classified has one feature, x in the range $1 \leq x < 5$. The conditional probability density functions for each class are listed below:

$$P(x|C_1) = \begin{cases} \frac{1}{4} & if\ 1 \leq x < 5 \\ 0 & otherwise \end{cases}$$

$$P(x|C_2) = \begin{cases} x-2 & if\ 2 \leq x < 3 \\ 4-x & if\ 3 \leq x < 4 \\ 0 & otherwise \end{cases}$$



   (a) Assuming equal priors, where $P(C_1) = P(C_2) = 0.5$, classify an object with the attribute value $x = \frac{5}{2}$.

   (b) Assuming unequal priors, where $P(C_1) = 0.75, P(C_2) = 0.25$, classify an object with the attribute value $x = 3$.

[1]Instructor: Catherine Qi Zhao. TA: Hanchen Cui, Hunmin Lee, Haoyi Shi, James Yang. Email: csci5521.s2025@gmail.com

(c) Consider a decision function $\phi(x)$ of the form $\phi(x) = (|x - 3|) - \alpha$ with one free parameter α in the range $0 \leq \alpha \leq 1$. You classify a given input x as class 2 if and only if $\phi(x) < 0$, or equivalently $3 - \alpha < x < 3 + \alpha$, otherwise you choose x as class 1. Assume equal priors, $P(C_1) = P(C_2) = 0.5$, what is the optimal decision boundary - that is, what is the value of α which minimizes the probability of misclassification? What is the resulting probability of misclassification with this optimal value for α? (Hint 1: take advantage of the symmetry around x = 3; Hint 2: express the probability of misclassification as a sum of integrals, each representing a specific range of x.)

3. (**40 points**) In this programming exercise you will implement three multivariate Gaussian classifiers about two classes. Given n training samples from `training_data.txt` (d-dimensional features $\mathbf{X} \in \mathbb{R}^{n \times d}$ and labels $\mathbf{y} \in \mathbb{R}^n$), all the classifiers will compute their corresponding parameters of both classes (mean $\mathbf{m} \in \mathbb{R}^d$, covariance matrix $\mathbf{S} \in \mathbb{R}^{d \times d}$ and prior $p \in \mathbb{R}$) for the purpose of classifying new data (test data `test_data.txt`). These three classifiers have different assumptions as follows:

Classifiers	means ($\mathbf{m}_1$, $\mathbf{m}_2$)	covariance matrices ($\mathbf{S}_1$, $\mathbf{S}_2$)	priors (p_1, p_2)
C1	$\mathbf{m}_1 \neq \mathbf{m}_2$	$\mathbf{S}_1 \neq \mathbf{S}_2$	$p_1 \neq p_2$
C2	$\mathbf{m}_1 \neq \mathbf{m}_2$	$\mathbf{S}_1 = \mathbf{S}_2$	$p_1 \neq p_2$
C3	$\mathbf{m}_1 \neq \mathbf{m}_2$	$\mathbf{S}_1 = \mathbf{S}_2$, $\mathbf{S}_1$ and $\mathbf{S}_2$ are diagonal	$p_1 \neq p_2$

Table 1: Parameter Assumptions of Three Classifiers.

We will next discuss how to compute the parameters. Please follow the way we taught in lectures (Page 12 of `Bayes_Parametric_Multivariate_II.pdf`) to estimate the parameters, otherwise the autograder may report errors.

- **C1** assumes that all the parameters (means, covariance matrices, priors) are learned independently for both classes. In other words, we will first split the training data into two sets (($\mathbf{X}_1$, $\mathbf{y}_1$), ($\mathbf{X}_2$, $\mathbf{y}_2$)) according to the labels (Class 1/Class 2), and then estimate the corresponding parameters following the formulas learnt from lectures.
- **C2** follows the same steps to compute the means and priors as C1. However, instead of directly utilizing independent covariance matrices, C2 computes a covariance matrix of all the data (*i.e.*, Class 1 and Class 2) to obtain a shared covariance matrix $\mathbf{S}$.
- **C3** follows the same steps to compute the parameters as C2. However, C3 additionally assumes that each feature is irrelevant to others (*e.g.*, the "size" and "color" of the "apple"), making the shared covariance matrix $\mathbf{S}$ diagonal. In other

words, we will only consider the variance of each feature and set the covariance between any two different features as 0.

After computing all the parameters for three classifiers, you can finally predict the labels of test data based on the discriminant functions. Please fill in the template code (`MyDiscriminant.py`), run the experiments (`hw1.py`) and answer several questions following the previously written answers. The only submitted code `MyDiscriminant.py` is written in a *scikit-learn* convention, where you have a *fit* function for model training and a *predict* function for generating predictions on given samples.

(a) In the lectures, we introduced different forms of discriminant functions (Page 13 for C1, Page 14 for C2, Page 17 for C3). What if you replace the discriminant functions of C2 and C3 with the discriminant function of C1? Will their prediction results change? Show in your report and briefly explain the reason.

(b) Report the precision and recall on the test set for each assumption. The definition of precision and recall can be found here. Based on the results, which model will you choose? Briefly explain your reasons. (Hint: May consider model complexity and overfitting issues. Any valid answers will be accepted and granted with full marks.)

Submission

- **Things to submit:**

  1. hw1_sol.pdf: a document containing all your answers for the written questions (including those in problem 3).
  2. `MyDiscriminant.py`: a Python source file containing three python classes for Problem 3. Use the skeleton file `MyDiscriminant.py` and fill in the missing parts. For each class object, the *fit* function should take the training features and labels as inputs, and update the model parameters. The *predict* function should take the test features as inputs and return the predictions.

- **Submit**: All material must be submitted electronically via Gradescope. **Note that There are two entries for the assignment, *i.e.,* Hw1-Written (for hw1_sol.pdf) and Hw1-Programming (for the Python code), please submit your files accordingly.** We will grade the assignment with vanilla Python with NumPy, and code submitted as iPython notebooks will not be graded.