ECE368: Probabilistic Reasoning
Lab 1: Naïve Bayes Classifier

You can complete this lab in a group of two. Please provide the name and student number of both members.

Name: Yubin (Olivia) Zhang	**Student Number:** 1008864837
Name: Rachel Min Yu	**Student Number:** 1006761751

You should hand in: 1) A scanned .pdf version of this sheet with your answers (file size should be under 2 MB); 2) one figure for Question 2.(c); and 3) A Python file classifier.py that contains your code. All these files should be uploaded to Quercus.

1 Naïve Bayes Classifier for Spam Filtering

1. (a) Write down the estimators for p_d and q_d as functions of the training data $\{\mathbf{x}_n, y_n\}, n = 1, 2, \ldots, N$ using the technique of "Laplace smoothing". (1 **pt**)

$$p_d = \frac{\sum_{i=0}^{N} x_{id} \mathbb{1}(y_i = 1) + 1}{\sum_{i=0}^{N} l_i \mathbb{1}(y_i = 1) + D}$$

$$q_d = \frac{\sum_{i=0}^{N} x_{id} \mathbb{1}(y_i = 0) + 1}{\sum_{i=0}^{N} l_n (y_i = 0) + 1}$$

   (b) Complete function learn_distributions in python file classifier.py based on the expressions you derived in part (a). (1 **pt**)

2. (a) Write down the MAP rule to decide whether $y = 1$ or $y = 0$ based on its feature vector $\mathbf{x}$ for a new email $\{\mathbf{x}, y\}$. The d-th entry of $\mathbf{x}$ is denoted by x_d. Please incorporate p_d and q_d in your expression. Please assume that $\pi = 0.5$. (1 **pt**)

$$\hat{y}_{MAP} = \arg\max_y \frac{P(y)P(x|y)}{P(y)} = \arg\max_y P(x|y)$$

$$= \frac{(x_1 + x_2 + \cdots + x_D)!}{x_1! x_2! \cdots x_D!} \prod_{d=1}^{D} P(w_d|y)^{x_d}$$

$$\prod_{d=1}^{D} p_d^{x_d} \underset{\text{ham}}{\overset{\text{spam}}{\gtrless}} \prod_{d=1}^{D} q_d^{x_d}$$

(b) Complete function classify_new_email in classifier.py, and test the classifier on the testing set. The number of Type 1 errors is [2], and the number of Type 2 errors is [5]. (1 **pt**)

(c) Write down the modified decision rule in the classifier such that these two types of error can be traded off. Please introduce a new parameter to achieve such a trade-off. (0.5 **pt**)

A new parameter λ (critical value) so that now we classify the result like this:

$$\prod_{d=1}^{D} p_d^{x_d} \underset{\text{ham}}{\overset{\text{spam}}{\gtrless}} \lambda \prod_{d=1}^{D} q_d^{x_d}$$

When $r > 1$, there's more probably to be classify as ham. Type I error decrease, Type II increases. Vice varsa for $r < 1$.

Write your code in file classifier.py to implement your modified decision rule. Test it on the testing set and plot a figure to show the trade-off between Type 1 error and Type 2 error. In the figure, the x-axis should be the number of Type 1 errors and the y-axis should be the number of Type 2 errors. Plot at least 10 points corresponding to different pairs of these two types of error in your figure. The two end points of the plot should be: 1) the point with zero Type 1 error; and 2) the point with zero Type 2 error. Please save the figure with name **nbc.pdf**. (1 **pt**)

3. Why do we need Laplace smoothing? Briefly explain what would go wrong if we do use the maximum likelihood estimators in the training process. (0.5 **pt**)

- Laplace smoothing will ensure that every word has a non-zero probability. It also prevents the model from giving too much confidance to words that appeared frequently and assigned some probability to unseen words (eg. it, I, has commonly used words could be classify as keywords for spam if we don't use laplace smoothing)
- If we use MLE alone, the model would fail to classify correctly if we have an unseen word in test email