# ECE368: Probabilistic Reasoning
# Lab 1: Classification with Multinomial and Gaussian Models

| **Name: Nathan Jones** | **Student Number: 1005003023** |
|---|---|

**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 **1**.2.(c) and two figures for Question **2**.1.(c) in the .pdf format; and 3) two Python files classifier.py and ldaqda.py that contain 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**)

spamwordcount[word], hamwordcount[word] = frequency the word appears in the spam / ham training sets
spamemailstotalwords, hamemailstotalwords = Total words in the in the spam / ham training sets
v = number of unique words in both spam and ham training sets

$$p_d[\text{word}] = \frac{\text{spamwordcount[word]} + 1}{\text{spamemailstotalwords} + v} \tag{1}$$

$$q_d[\text{word}] = \frac{\text{hamwordcount[word]} + 1}{\text{hamemailstotalwords} + v} \tag{2}$$

(b) Complete function learn_distributions in python file textsfclassifier.py based on the expressions. (1 **pt**)to the system.

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**)

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)} \tag{3}$$

$$p(x_n|y_n) = \frac{(x_{n,1} + x_{n,2} + \cdots + x_{n,D})!}{(x_{n,1})!(x_{n,2})! \cdots (x_{n,D})!} \prod_{d=1}^{D} p(w_d|y_n)^{x_{n,d}} \tag{4}$$

$$y = \begin{cases} 1 \text{ if } p(y=1|x) \geq p(y=0|x) \quad 0 \text{ if } p(y=1|x) < p(y=0|x) \end{cases} \tag{5}$$

$$\prod_{d=1}^{D} p_d[d]^{x_d} >= \prod_{d=1}^{D} q_d[d]^{x_d} \tag{6}$$

Essentially, the map rule is using bayes rule to find the probability of an email being spam based on the summed log probability of each word in the email being seen in the spam or ham training set, then comparing to see if its more likely spam or ham

(b) Complete function classify_new_email in textsfclassifier.py, and test the classifier on the testing set. The number of Type 1 errors is $\boxed{2}$, and the number of Type 2 errors is $\boxed{4}$. (1.5 **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**)

T = [1e-5, 1e-4, 1e-3, 1e-2, 1, 1e3, 1e4, 1e5, 1e10, 1e22]

$$\prod_{d=1}^{D} p_d[d]^{x_d} - log(T[n]) >= \prod_{d=1}^{D} q_d[d]^{x_d} \tag{7}$$

Where tradeoff T[n] can be altered to change whether or not the model is more likely to favor spam or ham

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**)

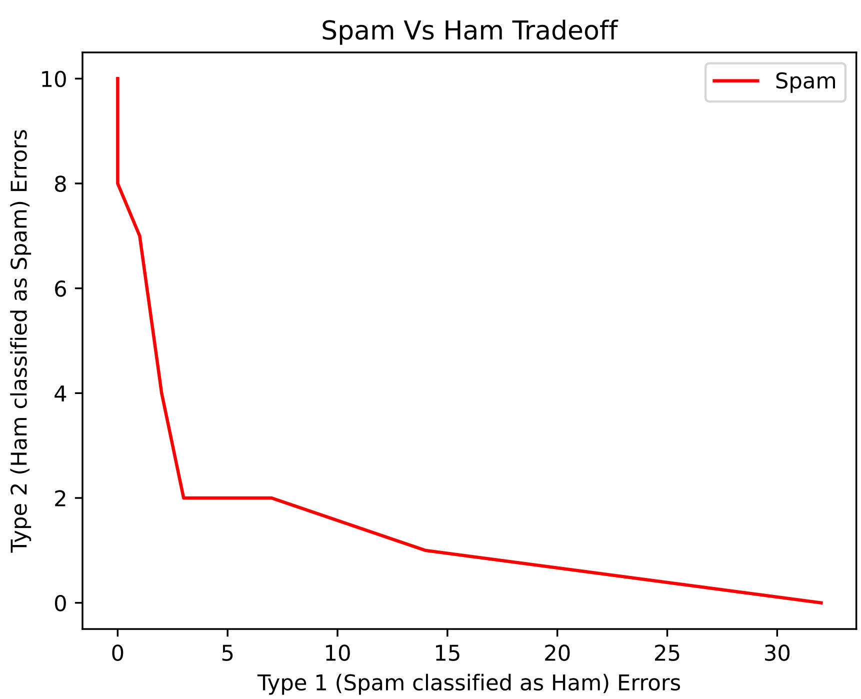


Figure 1: nbc.pdf

## 2 Linear/Quadratic Discriminant Analysis for Height/Weight Data

1. (a) Write down the maximum likelihood estimates of the parameters $\boldsymbol{\mu}_m$, $\boldsymbol{\mu}_f$, $\boldsymbol{\Sigma}$, $\boldsymbol{\Sigma}_m$, and $\boldsymbol{\Sigma}_f$ as functions of the training data $\{\mathbf{x}_n, y_n\}, n = 1, 2, \ldots, N$. (1 **pt**)

$$\mu_m = \frac{\sum n = 1^N [x_n] \cdot [y_n = 1]}{\sum_{n=1}^N [y_n = 1]} \tag{8}$$

$$\mu_f = \frac{\sum n = 1^N x_n \cdot [y_n = 2]}{\sum_{n=1}^N [y_n = 2]} \tag{9}$$

$$\Sigma = \frac{1}{N} \sum_{n=1}^N (x_n - \hat{\mu} y_n)(x_n - \hat{\mu} y_n)^T \tag{10}$$

$$\Sigma_m = \frac{1}{\sum n = 1^N [y_n = 1]} \sum_{n=1}^N (x_n - \mu_m)(x_n - \mu_m)^T \cdot [y_n = 1] \tag{11}$$

$$\Sigma_f = \frac{1}{\sum n = 1^N [y_n = 0]} \sum_{n=1}^N (x_n - \mu_f)(x_n - \mu_f)^T \cdot [y_n = 2] \tag{12}$$

(b) In the case of LDA, write down the decision boundary as a linear equation of $\mathbf{x}$ with parameters $\boldsymbol{\mu}_m$, $\boldsymbol{\mu}_f$, and $boldsymbol\Sigma$. Note that we assume $\pi = 0.5$. (0.5 **pt**)

$$(\boldsymbol{\mu}_m - \boldsymbol{\mu}_f)^T \boldsymbol{\Sigma}^{-1} \mathbf{x} - \frac{1}{2} \boldsymbol{\mu}_m^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_m + \frac{1}{2} \boldsymbol{\mu}_f^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}_f = 0 \tag{13}$$

In the case of QDA, write down the decision boundary as a quadratic equation of $\mathbf{x}$ with parameters $\boldsymbol{\mu}_m$, $\boldsymbol{\mu}_f$, $boldsymbol\Sigma_m$, and $\boldsymbol{\Sigma}_f$. Note that we assume $\pi = 0.5$. (0.5 **pt**)

$$\mathbf{x}^T (\boldsymbol{\Sigma}_m^{-1} - \boldsymbol{\Sigma}_f^{-1}) \mathbf{x} + 2(\boldsymbol{\mu}_f^T \boldsymbol{\Sigma}_f^{-1} - \boldsymbol{\mu}_m^T \boldsymbol{\Sigma}_m^{-1}) \mathbf{x} + \boldsymbol{\mu}_m^T \boldsymbol{\Sigma}_m^{-1} \boldsymbol{\mu}_m - \boldsymbol{\mu}_f^T \boldsymbol{\Sigma}_f^{-1} \boldsymbol{\mu}_f - \log \frac{|\boldsymbol{\Sigma}_m|}{|\boldsymbol{\Sigma}_f|} = 0 \tag{14}$$

(c) Complete function discrimAnalysis in ldaqda.py to visualize LDA and QDA models and the corresponding decision boundaries. Please name the figures as lda.pdf, and qda.pdf. (1 **pt**)

**lda.pdf and qda.pdf on pages 4 and 5**

2. The misclassification rates are $\boxed{0.11818}$ for LDA, and $\boxed{0.10909}$ for QDA. (1 **pt**)

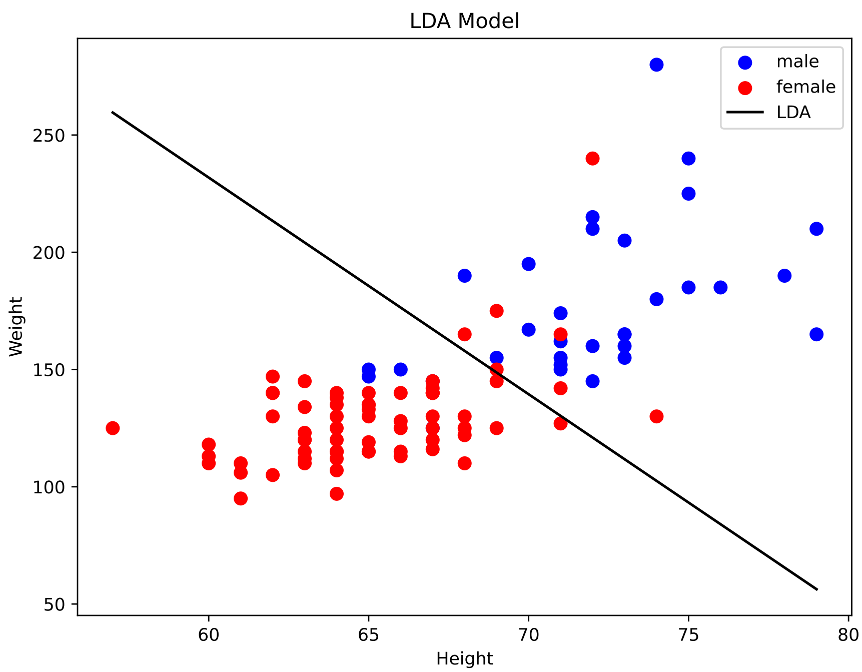


Figure 2: lda.pdf

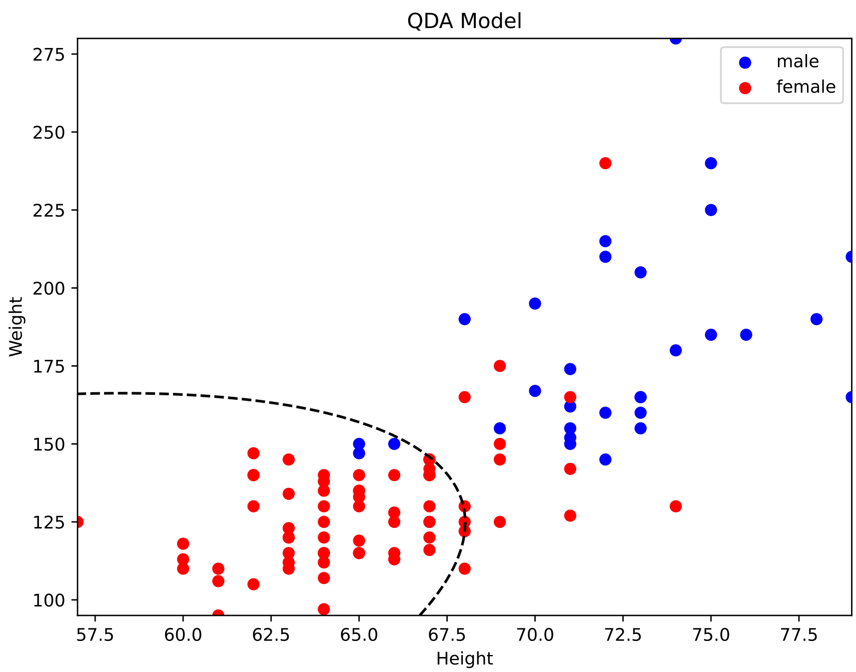


Figure 3: qda.pdf