ECE368: Probabilistic Reasoning

Lab 1: Classification with Multinomial and Gaussian Models

Name: Lisa Li	**Student Number:** 1003924532

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**)

$$\hat{p}_d = \frac{\sum_{i=1}^{N} x_{di}\, \mathbb{I}(y_i = 1) + 1}{\sum_{i=1}^{N}\sum_{d=1}^{D} x_{di}\, \mathbb{I}(y_i = 1) + len(D)} \qquad \hat{q}_d = \frac{\sum_{i=1}^{N} x_{di}\, \mathbb{I}(y_i = 0) + 1}{\sum_{i=1}^{N}\sum_{d=1}^{D} x_{di}\, \mathbb{I}(y_i = 0) + len(D)}$$

# of words in all SPAM ↑ ; ↑ # of words in vocab

(b) Complete function learn_distributions in python file classifier.py based on the expressions. (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**)

$$\log(\pi) + \sum_{d=1}^{D} x_d \log(\hat{p}_d) \underset{y_n=0}{\overset{y_n=1}{\gtrless}} \log(1-\pi) + \sum_{d=1}^{D} x_d \log(\hat{q}_d)$$

(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 [4]. (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**)

If $L_{FN} = c\, L_{FP}$

$p(y=1|x) > \tau \cdot p(y=0|x)$ where $\tau = \frac{c}{1+c}$

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**)

(d) If we do not use Laplace smoothing and simply use maximum likelihood estimation in the training phase, what will go wrong? What kind of emails such a classifier would fail to classify? (0.5 **pt**)

Say w_d is in SPAM but not HAM ; $P(w_d \mid \text{HAM}) = 0$. This will cause $y_{\text{pred,HAM}} = 0$. It'd fail to classify emails w/ words that were found in one but not the other during training.

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**)

$$\mu_m = \frac{1}{N_m} \sum_{n=1}^{N} \underline{x}_n \, \mathbb{I}(y=1) \qquad \Sigma_m = \frac{1}{N_m} \sum_{n=1}^{N} (\underline{x}_n - \mu_m)(\underline{x}_n - \mu_m)^T \mathbb{I}(y=1)$$

$$\mu_f = \frac{1}{N_f} \sum_{n=1}^{N} \underline{x}_n \, \mathbb{I}(y=2) \qquad \Sigma_f = \frac{1}{N_f} \sum_{n=1}^{N} (\underline{x}_n - \mu_f)(\underline{x}_n - \mu_f)^T \mathbb{I}(y=2)$$

$$\mu = \frac{1}{N} \sum_{n=1}^{N} \underline{x}_n \qquad \Sigma = \frac{1}{N} \sum_{n=1}^{N} (\underline{x}_n - \mu)(\underline{x}_n - \mu)^T$$

(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**)

$$\log(\pi) - \tfrac{1}{2}\mu_m^T \Sigma^{-1} \mu_m + x^T \Sigma^{-1} \mu_m = \log(1-\pi) - \tfrac{1}{2}\mu_f^T \Sigma^{-1} \mu_f + x^T \Sigma^{-1} \mu_f$$

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**)

$$\log(\pi) - \tfrac{1}{2}\mu_m^T \Sigma_m^{-1} \mu_m + x^T \Sigma_m^{-1} \mu_m - \tfrac{1}{2} x^T \Sigma_m^{-1} x - \tfrac{1}{2}\log|\Sigma_m| =$$
$$\log(1-\pi) - \tfrac{1}{2}\mu_f^T \Sigma_f^{-1} \mu_f + x^T \Sigma_f^{-1} \mu_f - \tfrac{1}{2} x^T \Sigma_f^{-1} x - \tfrac{1}{2}\log|\Sigma_f|$$

(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**)

2. The misclassification rates are [11.8%] for LDA, and [10.9%] for QDA. (1 **pt**)