HW 1

Arnab Dey
Student ID: 5563169
Email: dey00011@umn.edu

Solution 1.a

The expected loss of a function $f(x)$ is in modelling y using loss function $\ell(f(x), y)$ is given by

$$\begin{aligned}\mathbb{E}_{(x,y)}[\ell(f(x), y)] &= \int_x \int_y \ell(f(x), y) p(x, y) dy dx \\ &= \int_x \left[\int_y \ell(f(x), y) p(y|x) dy \right] p(x) dx.\end{aligned}$$

When $\ell(f(x), y) = (f(x) - y)^2$, to find the optimal $f(x)$, we take gradient of the expected loss and equate it to 0. Therefore, denoting the optimal $f(x)$ as $f^*(x)$,

$$\begin{aligned}&\frac{\partial}{\partial f} \mathbb{E}_{(x,y)}[\ell(f(x), y)] = 0 \\ \implies &\int_x \frac{\partial}{\partial f} \left[\int_y (f(x) - y)^2 p(y|x) dy \right] p(x) dx = 0 \\ \implies &\int_x \left[(f^*(x) - y) p(y|x) dy \right] p(x) dx = 0 \\ \implies &\int_x \left[\int_y f^*(x) p(y|x) dy \right] p(x) dx = \int_x \left[\int_y y p(y|x) dy \right] p(x) dx \\ \implies &\int_y f^*(x) p(y|x) dy = \int_y y p(y|x) dy \\ \implies &f^*(x) = \mathbb{E}[y|x].\end{aligned}$$

Solution 1.b

We know that,

$$\frac{\partial}{\partial x} |x| = sgn(x), \text{ for } x \neq 0,$$

where,

$$sgn(x) = \begin{cases} -1 & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ 1 & \text{if } x > 0. \end{cases}$$

Now, the optimal value of x such that $\int sgn(x - y) dy = 0$ is the median of y as equal number of positive and negative samples of y gives the minimum value of the integral. Therefore,

$$\arg\min_f \mathbb{E}_{(x,y)}[|f(x) - y|]$$

can be found from the solution of

$$\int_x \left[\int_y sgn(f(x) - y) p(y|x) dy \right] p(x) dx = 0.$$

Hence, if the optimal value of $f(x)$ is denoted by $f^*(x)$,

$$f^*(x) = \text{median}(y|x)$$

Solution 2

Let $f \coloneqq \frac{1}{2}\left((A^TA)^{-1}A^T\right)^T c$. As the columns of $A \in \mathbb{R}^{m\times n}$ are linearly independent, $(A^TA)^{-1}$ exists and therefore the definition of f is valid. Also, let $g \coloneqq d - f^Tf + 2b^Tf$, where $g \in \mathbb{R}$. Therefore,

$$\begin{aligned}
\|Aw - b + f\|_2^2 + g &= (Aw - b + f)^T(Aw - b + f) + g \\
&= ((Aw - b)^T + f^T)((Aw - b) + f) + g \\
&= (Aw - b)^T(Aw - b) + (Aw - b)^Tf + f^T(Aw - b) + f^Tf + g \\
&= \|Aw - b\|_2^2 + w^TA^Tf - b^Tf + f^TAw - f^Tb + f^Tf + g \\
&= \|Aw - b\|_2^2 + \frac{1}{2}w^TA^T((A^TA)^{-1}A^T)^Tc + c^T((A^TA)^{-1}A^T)Aw - 2b^Tf + f^Tf + g \\
&= \|Aw - b\|_2^2 + \frac{1}{2}w^T(A^TA)(A^TA)^{-1}c + \frac{1}{2}c^Tw + d \\
&= \|Aw - b\|_2^2 + \frac{1}{2}w^Tc + \frac{1}{2}c^Tw + d \\
&= \|Aw - b\|_2^2 + c^Tw + d.
\end{aligned}$$

The last line follows from the fact that c^Tw is a scalar. □

Therefore,

$$\min_w \|Aw - v\|_2^2 + c^Tw + d = \min_w \|Aw - b + f\|_2^2 + g.$$

To find the optimal value of w which minimizes $\|Aw - b + f\|_2^2 + g$, we define the cost function,

$$J(w) = (Aw - (b - f))^T(Aw - (b - f)) + g,$$

and equate $\frac{\partial J(w)}{\partial w}$ to 0. Let the optimal value of w is denoted by w^*. Therefore,

$$\begin{aligned}
&\frac{\partial J(w)}{\partial w} = 0 \\
\implies &A^T(Aw^* - (b - f)) = 0 \\
\implies &A^TAw^* = A^T(b - f) \\
\implies &w^* = (A^TA)^{-1}A^T(b - f).
\end{aligned}$$

Solution 3.i

Summary of LDA: In Fisher's Linear Discriminant Analysis (LDA), the objective is to find the projection matrix $\boldsymbol{W} \in \mathbb{R}^{d\times k}$ such that, the N samples of d-dimensional data, $\boldsymbol{X}$, from K different classes, when projected onto $\boldsymbol{W}$ are well separable. Therefore, the projected data for a sample $\boldsymbol{x}$ is,

$$\boldsymbol{z} = \boldsymbol{W}^T\boldsymbol{x},$$

where $\boldsymbol{x} \in \mathbb{R}^d \in \boldsymbol{X}$. We denote the different classes as $C_i, i \in \{1, 2, \ldots, K\}$.

The within-class scatter matrix for class C_i is given by,

$$\boldsymbol{S}_i = \sum_{t=1}^{N} r_i^t(\boldsymbol{x}^t - \boldsymbol{m}_i)(\boldsymbol{x}^t - \boldsymbol{m}_i)^T,$$

where $r_i^t = 1$, if $\boldsymbol{x}^t \in C_i$ and 0 otherwise and

$$\boldsymbol{m}_i = \frac{\sum_{t=1}^N \boldsymbol{x}^t r_i^t}{\sum_{t=1}^N r_i^t}$$

is the mean of samples belonging to class C_i. The overall within class scatter matrix is,

$$\boldsymbol{S}_W = \sum_{i=1}^{K} \boldsymbol{S}_i.$$

Also, the overall mean of all the samples are,

$$\boldsymbol{m} = \frac{1}{K} \sum_{i=1}^{K} \boldsymbol{m}_i.$$

The between class scatter matrix is,

$$\boldsymbol{S}_B = \sum_{i=1}^{K} \left(\sum_{t=1}^{N} r_i^t \right) (\boldsymbol{m}_i - \boldsymbol{m})(\boldsymbol{m}_i - \boldsymbol{m})^T.$$

In Fisher's LDA we are interested to find the matrix $\boldsymbol{W}$ which maximizes

$$J(\boldsymbol{W}) = \frac{|\boldsymbol{W}^T \boldsymbol{S}_B \boldsymbol{W}|}{|\boldsymbol{W}^T \boldsymbol{S}_W \boldsymbol{W}|}.$$

The largest eigen-vectors of $\boldsymbol{S}_W^{-1}\boldsymbol{S}_B$ are the solution. Therefore, to project the data onto $\mathbb{R}$, we need to take the largest eigen-vector of $\boldsymbol{S}_W^{-1}\boldsymbol{S}_B$ as the projection matrix and to project the data onto $\mathbb{R}^2$, we need to take the largest two eigen-vectors of $\boldsymbol{S}_W^{-1}\boldsymbol{S}_B$ as the projection matrix.

Once the projection is done, we need to find a threshold to classify any new data. For Q3.i, I have taken the projected overall mean as the threshold in case the projection is onto $\mathbb{R}$. Therefore, denoting the projection vector as $\boldsymbol{w}_1 \in \mathbb{R}^d$,

$$threshold = \boldsymbol{w}_1^T \boldsymbol{m}.$$

If the projected value of any new data is larger than the *threshold*, it is classified as belonging to C_1, otherwise labeled as C_2.

Result: The training and test error percentage for 10-fold cross validation is tabulated in Table 1. The mean and standard deviation of the error for both training and test data are shown in Table 2.

Table 1: Q3.i: Error table for 10-fold cross validation: Boston50 dataset

	fold 1	fold 2	fold 3	fold 4	fold 5	fold 6	fold 7	fold 8	fold 9	fold 10
Training(%)	14.25	13.82	14.25	14.04	14.04	14.25	12.28	13.60	16.01	14.22
Validation(%)	12.0	20.0	14.0	14.0	18.0	8.0	34.0	30.0	4.0	23.21

Table 2: Q3.i: Mean and Std of error on Boston50 dataset

Dataset	Mean	Standard deviation
Training(%)	14.08	0.86
Validation(%)	17.72	8.91

Solution 3.ii

Gaussian Generative Model: For classification as required in Q3.ii, the first step is to use maximum likelihood estimator (MLE) to estimate the parameters of the gaussian pdf for each class *i.e* (μ_k, Σ_k) for all

$k \in \{1, 2, \ldots, 10\}$ in case of digits data. Output of the MLE process is the class priors, $\hat{P}(C_i)$, means, m_i and covariance matrices, S_i for all $i \in \{1, 2, \ldots, 10\}$. The expressions are given below:

$$\begin{aligned} \hat{P}(C_i) &= \frac{\sum_{t=1}^{N} r_i^t}{N} \\ m_i &= \frac{\sum_{t=1}^{N} r_i^t x^t}{\sum_{t=1}^{N} r_i^t} \\ S_i &= \frac{\sum_{t=1}^{N} r_i^t (x^t - m_i)(x^t - m_i)^T}{\sum_{t=1}^{N} r_i^t}, \end{aligned} \tag{1}$$

for all $i \in \{1, 2, \ldots, 10\}$ where $r_i^t = 1$ if $x^t \in C_i$ and 0 otherwise.

Once the parameters are estimated, we need to find the discriminant function to classify new data. Form Bayes rule,

$$\hat{P}(C_i|x) = \frac{\hat{P}(x|C_i)\hat{P}(C_i)}{\sum_{j=1}^{10} \hat{P}(x|C_i)\hat{P}(C_i)}.$$

Assuming class-conditional densities are Normal and the samples are d-dimensional, $i.e$

$$\hat{P}(x|C_i) = \frac{1}{(2\pi)^{\frac{d}{2}}|S_i|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(x - m_i)^T S_i^{-1}(x - m_i)\right],$$

and as the denominator is common for all $i \in \{1, 2, \ldots, 10\}$, we can define the discriminant function for class C_i as

$$\begin{aligned} g_i(x) &= \log(\hat{P}(x|C_i)) + \log(\hat{P}(C_i)) \\ &= -\frac{d}{2}\log(2\pi) - \frac{1}{2}\log(|S_i|) - \frac{1}{2}(x - m_i)^T S_i^{-1}(x - m_i) + \log(\hat{P}(C_i)). \end{aligned}$$

Also, we can see that the first term is common to all classes, therefore, we can drop that and use the discriminant function as,

$$g_i(x) = -\frac{1}{2}\log(|S_i|) - \frac{1}{2}(x - m_i)^T S_i^{-1}(x - m_i) + \log(\hat{P}(C_i)). \tag{2}$$

We label x to class C_i if $g_i(x) > g_j(x)$ for all $j \neq i$ where $i, j \in \{1, 2, \ldots, 10\}$.

Result: The training and test error percentage for 10-fold cross validation is tabulated in Table 3. The mean and standard deviation of the error for both training and test data are shown in Table 4.

Table 3: Q3.ii: Error table for 10-fold cross validation: Digits dataset

	fold 1	fold 2	fold 3	fold 4	fold 5	fold 6	fold 7	fold 8	fold 9	fold 10
Training(%)	24.04	29.48	32.76	27.81	27.13	35.91	34.86	28.06	27.87	27.00
Validation(%)	30.73	26.82	44.69	40.22	34.08	46.93	39.11	41.90	32.96	30.65

Table 4: Q3.ii: Mean and Std of error on Digits dataset

Dataset	Mean	Standard deviation
Training(%)	29.49	3.60
Validation(%)	36.81	6.36

Solution 4

Logistic regression (LR): Let $N, K, x \in \mathbb{R}^d, C_i, w_i \in \mathbb{R}^d, w_{i0} \in \mathbb{R}$ denote number of samples, number of classes, data sample, i^{th} class, weights and intercept of the linear model respectively. For logistic regression, I use the $softmax$ function, as shown below, to find the posterior:

$$y_i = \hat{P}(C_i|x) = \frac{\exp[w_i^T x + w_{i0}]}{\sum_{j=1}^{K} \exp[w_j^T x + w_{j0}]},$$

for all $i \in \{1, 2, \ldots, K\}$. The error function is

$$E = -\sum_{t=1}^{N}\sum_{i=1}^{K} r_i^t \log(y_i),$$

where $r_i^t = 1$ if $x^t \in C_i$ and 0 otherwise. The objective is to find optimal values of w_i, w_{i0} for all $i \in \{1, 2, \ldots, K\}$ such that the error E is minimized. I use gradient descent to find the updates to w_i, w_{i0}. The gradient of E with respect to w_i is

$$\begin{aligned}\frac{\partial E}{\partial w_i} &= \frac{\partial E}{\partial y_i}\frac{\partial y_i}{\partial w_i} \\ &= -\sum_{t=1}^{N}(r_i^t - y_i^t)x^t,\end{aligned}$$

and

$$\begin{aligned}\frac{\partial E}{\partial w_{i0}} &= \frac{\partial E}{\partial y_i}\frac{\partial y_i}{\partial w_{i0}} \\ &= -\sum_{t=1}^{N}(r_i^t - y_i^t).\end{aligned}$$

Therefore, the update equation for gradient descent algorithm is

$$\begin{aligned}w_i^{new} &= w_i^{old} + \eta\sum_{t=1}^{N}(r_i^t - y_i^t)x^t, \\ w_{i0}^{new} &= w_{i0}^{old} + \eta\sum_{t=1}^{N}(r_i^t - y_i^t),\end{aligned}$$

where η is the learning rate. I have taken $\eta = 0.001$. I have also kept a provision to add regularization term in my code for which the update equation becomes,

$$\begin{aligned}w_i^{new} &= w_i^{old} + \eta\sum_{t=1}^{N}(r_i^t - y_i^t)x^t - \eta C w_i^{old}, \\ w_{i0}^{new} &= w_{i0}^{old} + \eta\sum_{t=1}^{N}(r_i^t - y_i^t) - \eta C w_{i0}^{old},\end{aligned}$$

where C is the regularization term which I have taken as 10 in my code. I have used batch gradient descent.

Naive Bayes with marginal Gaussian distributions (GNB): In case of Naive Bayes, we assume that the features in the samples are uncorrelated. Therefore, the covariance matrices are diagonal. The approach is similar to as described in Solution 3.ii. But to compare GNB to linear LR, we need to make use

of shared covariance matrix for all classes. Therefore, we can use pooling of data to find a shared covariance matrix as follows

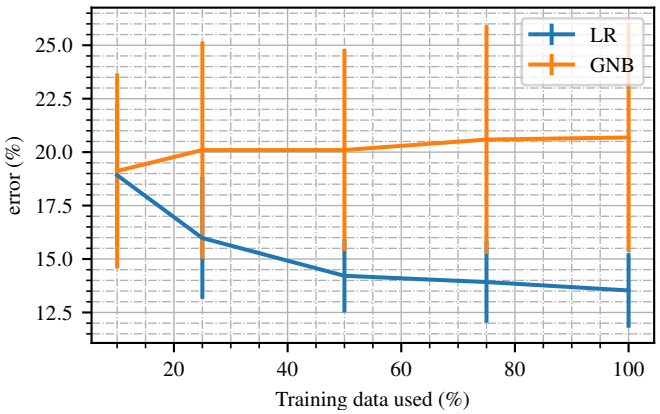
$$S = \sum_{i=1}^{K} \hat{P}(C_i) S_i, \tag{3}$$

where $\hat{P}(C_i), S_i$ are defined in (1). Now, if we use S in place of S_i for all $i \in \{1, 2, \ldots, K\}$ in (2), we can see that the term $-\frac{1}{2}\log(|S|)$ becomes common to all discriminators. Therefore, we can drop this term and use

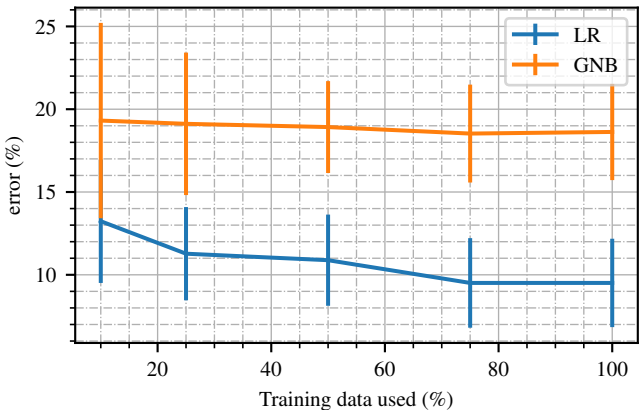
$$g_i(x) = -\frac{1}{2}(x - m_i)^T S^{-1}(x - m_i) + \log(\hat{P}(C_i)).$$

as the discriminator. In this case, we see that the quadratic term $\frac{1}{2}x^T S^{-1} x$ becomes common to all discriminators, thus, the discriminator gives rise to a linear function of x which can be compared to LR model.

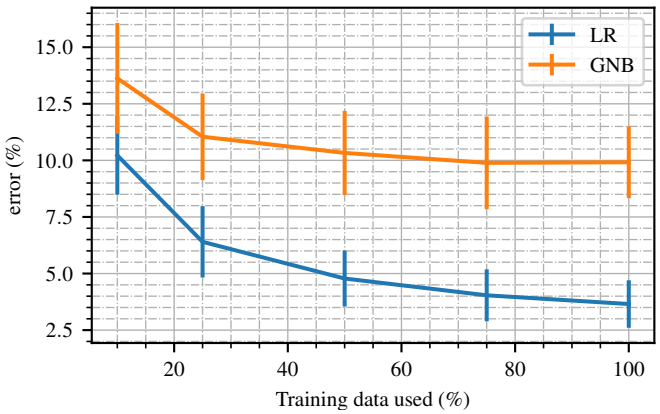
Results: Fig. 1a, Fig. 1b and Fig. 1c show the test data error percentage for LR anf GNB model, for 'Boston50', 'Boston75' and 'Digits' data respectively, with percentage of training data used to train the model in x-axis. It can be seen that the LR model performs better than GNB.



(a) Test set error plot on Boston50 dataset



(b) Test set error plot on Boston75 dataset



(c) Test set error plot on digits dataset

Figure 1: Q4: Test data error comparison between LR and GNB