NYCU Introduction to Machine Learning, Homework 2

Part. 1, Coding (60%):

1. (5%) Compute the mean vectors m_i (i=1, 2) of each 2 classes on **training data**

```
mean vector of class 1:
[ 0.99253136 -0.99115481]
mean vector of class 2:
[-0.9888012   1.00522778]
```

2. (5%) Compute the within-class scatter matrix S_W on **training data**

```
Within-class scatter matrix SW:
[[ 4337.38546493 -1795.55656547]
 [-1795.55656547  2834.75834886]]
```

3. (5%) Compute the between-class scatter matrix S_B on **training data**

```
Between-class scatter matrix SB:
[[ 3.92567873 -3.95549783]
 [-3.95549783  3.98554344]]
```

4. (5%) Compute the Fisher's linear discriminant w on **training data**

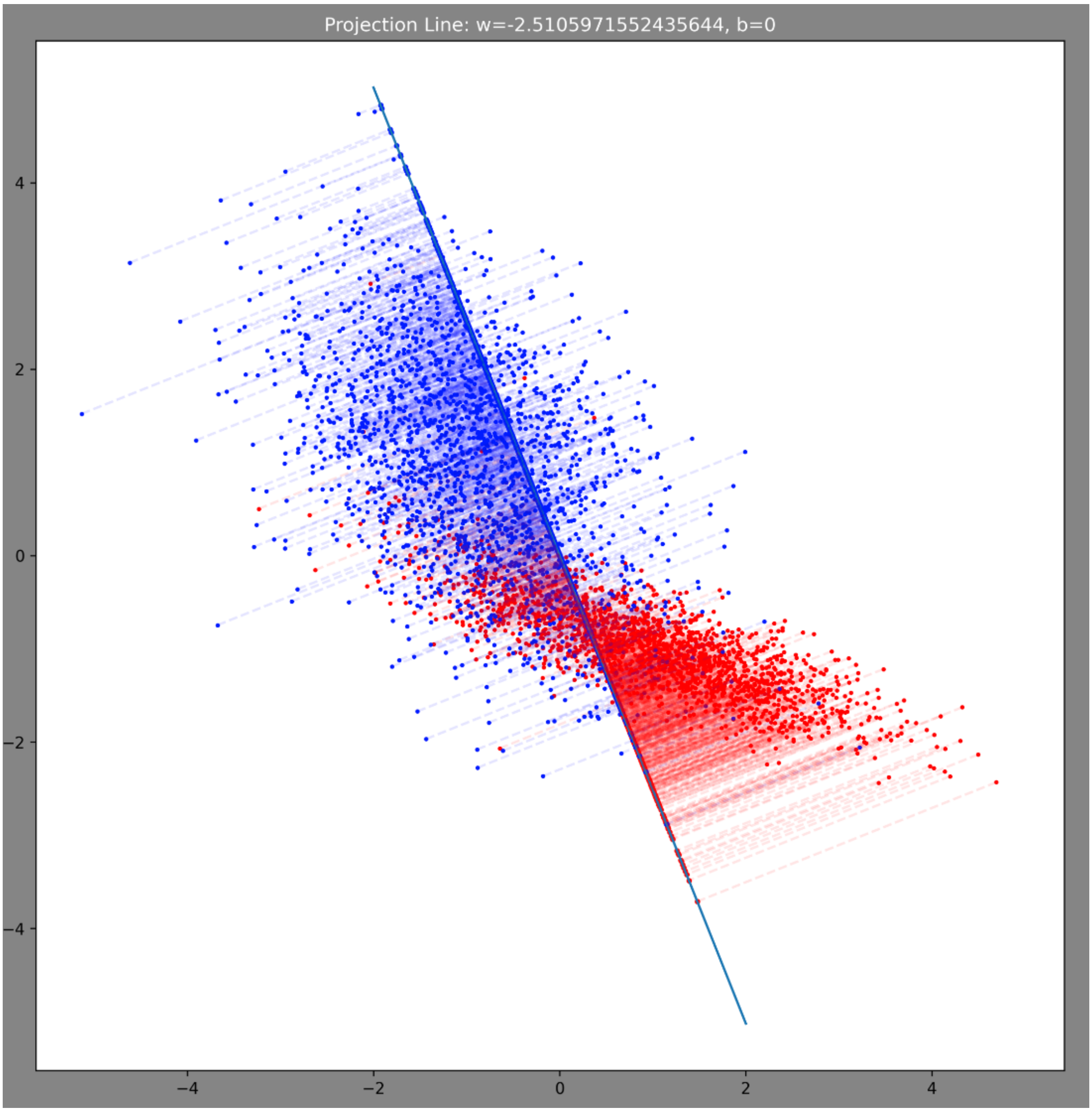
```
Fisher's linear discriminant:
[-0.37003809  0.92901658]
```

5. (20%) Project the **testing data** by Fisher's linear discriminant to get the class prediction by K-Nearest-Neighbor rule and report the accuracy score on **testing data** with K values from 1 to 5 (you should get accuracy over **0.88**)

```
k = 1: Accuracy of test-set 0.8488
k = 2: Accuracy of test-set 0.8488
k = 3: Accuracy of test-set 0.8792
k = 4: Accuracy of test-set 0.8824
k = 5: Accuracy of test-set 0.8912
```

6. (20%) Plot the **1) best projection line** on the **training data** and show the slope and intercept on the title *(you can choose any value of **intercept** for better visualization)* **2) colorize the data** with each class **3)** project all data points on your projection line.

Your result should look like the below image (This image is for reference, not the answer)



Part. 2, Questions (40%):

Please write/type by yourself. DO NOT screenshot the solution from others.

(10%) 1. What's the difference between the Principle Component Analysis and Fisher's Linear Discriminant?

The main difference between PCA and FLD is their technique for dimensionality reduction. FLD is a supervised dimensionality reduction while PCA is unsupervised. Another difference is that FLD aims at maximizing the separability between groups, while PCA focuses on maximizing variation in the data set.

(10%) 2. Please explain in detail how to extend the 2-class FLD into multi-class FLD (the number of classes is greater than two).

First, we need to extend the formulation of the within-class covariance matrix S_W to k ≥ 2 (formulation below).

$$S_W = \sum_{k=1}^{K} S_k \text{ , where } S_k = \sum_{n \in C_k} (x_n - m_k)(x_n - m_k)^T \text{ , } m_k = \frac{1}{N_k} \sum_{n \in C_k} x_n$$

Second, we also need to extend the between-class covariance matrix to k > 2, just like the formation below.

$$S_B = \sum_{k=1}^{K} N_k (m_k - m)(m_k - m)^T \text{ , where } m = \frac{1}{N} \sum_{n=1}^{N} x_n$$

After that, because the number of classes is no longer 2, the Lagrangian function needed to be revised into the following equation.

$$\mathscr{L}_P = -\frac{1}{2} w^T S_B w + \frac{1}{2} \lambda (w^T S_W w - 1) \text{ , and } S_B w = \lambda S_w w \Longrightarrow S_W^{-1} S_B w = \lambda w$$

In the end, we need to optimize w by finding the eigenvector of $S_w^{-1} S_B$ which maximizes the eigenvalue.

(6%) 3. By making use of Eq (1) ~ Eq (5), show that the Fisher criterion Eq (6) can be written in the form Eq (7).

$$y = \mathbf{w}^{\mathrm{T}} \mathbf{x} \qquad \text{Eq (1)}$$

$$\mathbf{m}_1 = \frac{1}{N_1} \sum_{n \,\in\, \mathcal{C}_1} \mathbf{x}_n \qquad \mathbf{m}_2 = \frac{1}{N_2} \sum_{n \,\in\, \mathcal{C}_2} \mathbf{x}_n \qquad \text{Eq (2)}$$

$$m_2 - m_1 = \mathbf{w}^{\mathrm{T}} (\mathbf{m}_2 - \mathbf{m}_1) \qquad \text{Eq (3)}$$

$$m_k = \mathbf{w}^{\mathrm{T}} \mathbf{m}_k \qquad \text{Eq (4)}$$

$$s_k^2 = \sum_{n \in \mathcal{C}_k} (y_n - m_k)^2 \qquad \text{Eq (5)}$$

$$J(\mathbf{w}) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} \qquad \text{Eq (6)}$$

$$J(\mathbf{w}) = \frac{\mathbf{w}^{\mathrm{T}} \mathbf{S}_{\mathrm{B}} \mathbf{w}}{\mathbf{w}^{\mathrm{T}} \mathbf{S}_{\mathrm{W}} \mathbf{w}} \qquad \text{Eq (7)}$$

$$J(w)=\frac{(m_2-m_1)^2}{s_1^2+s_2^2}=\frac{[w^T(m_2-m_1)]^2}{\sum_{n\in C_1}(w^Tx_n-w^Tm_1)^2+\sum_{n\in C_2}(w^Tx_n-w^Tm_2)^2}$$

$$=\frac{[w^T(m_2-m_1)][w^T(m_2-m_1)]^T}{\sum_{n\in C_1}[w^T(x_n-m_1)][w^T(x_n-m_1)]^T+\sum_{n\in C_2}[w^T(x_n-m_2)][w^T(x_n-m_2)]^T}$$

$$=\frac{w^T(m_2-m_1)(m_2-m_1)^Tw}{\sum_{n\in C_1}w^T(x_n-m_1)(x_n-m_1)^Tw+\sum_{n\in C_2}w^T(x_n-m_2)(x_n-m_2)^Tw}$$

$$=\frac{w^T(m_2-m_1)(m_2-m_1)^Tw}{w^T[\sum_{n\in C_1}(x_n-m_1)(x_n-m_1)^T+\sum_{n\in C_2}(x_n-m_2)(x_n-m_2)^T]w}$$

$$=\frac{w^TS_Bw}{w^TS_Ww}$$

(7%) 4. Show the derivative of the error function Eq (8) with respect to the activation a_k for an output unit having a logistic sigmoid activation function satisfies Eq (9).

$$E(\mathbf{w})=-\sum_{n=1}^{N}\{t_n\ln y_n+(1-t_n)\ln(1-y_n)\} \qquad \text{Eq (8)}$$

$$\frac{\partial E}{\partial a_k}=y_k-t_k \qquad \text{Eq (9)}$$

$$E(w)=-\sum_{n=1}^{N}\{t_n\ ln(\sigma(a))+(1-t_n)\ ln(1-\sigma(a))\}$$

$$\frac{\partial E}{\partial a_k}=-\frac{\partial}{\partial a_k}\Big(t_k\ ln(\sigma(a_k))+(1-t_k)\ ln(1-\sigma(a_k))\Big)$$

$$=-\Big(\frac{t_k}{\sigma(a_k)}\sigma(a_k)(1-\sigma(a_k))-\frac{1-t_k}{1-\sigma(a_k)}\sigma(a_k)(1-\sigma(a_k))\Big)$$

$$=-t_k(1-\sigma(a_k))+(1-t_k)\sigma(a_k)$$

$$=-t_k+t_k\sigma(a_k)+\sigma(a_k)-t_k\sigma(a_k)$$

$$=\sigma(a_k)-t_k=y_k-t_k$$

(7%) 5. Show that maximizing likelihood for a multiclass neural network model in which the network outputs have the interpretation $y_k(x, w) = p(t_k = 1 \mid x)$ **is equivalent to the minimization of the cross-entropy error function Eq (10).**

$$E(\mathbf{w}) = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{kn} \ln y_k(\mathbf{x}_n, \mathbf{w}) \qquad \text{Eq (10)}$$

$$\because\ y_k(x_n, w) = p(t_k = 1 \mid x_n) = \prod_{n=1}^{N}\prod_{k=1}^{K} p(C_k \mid x_n)^{t_{nk}} = \prod_{n=1}^{N}\prod_{k=1}^{K} y_{nk}^{t_{nk}}$$

$\therefore$ Maximize likelihood is equivalent to minimize

$$-ln(\prod_{n=1}^{N}\prod_{k=1}^{K} y_{nk}^{t_{nk}}) = E(w) = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk}\ ln(y_{nk})$$