NYCU Introduction to Machine Learning,Homework 2

109550064 陳姵帆

Part. 1, Coding (60%):

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

```
print(f"mean vector of class 1: {m1}\n", f"mean vector of class 2: {m2}")
✓ 0.1s

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**

```
print(f"Within-class scatter matrix SW: {sw}")
✓ 0.1s

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**

```
print(f"Between-class scatter matrix SB: {sb}")
✓ 0.2s

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**

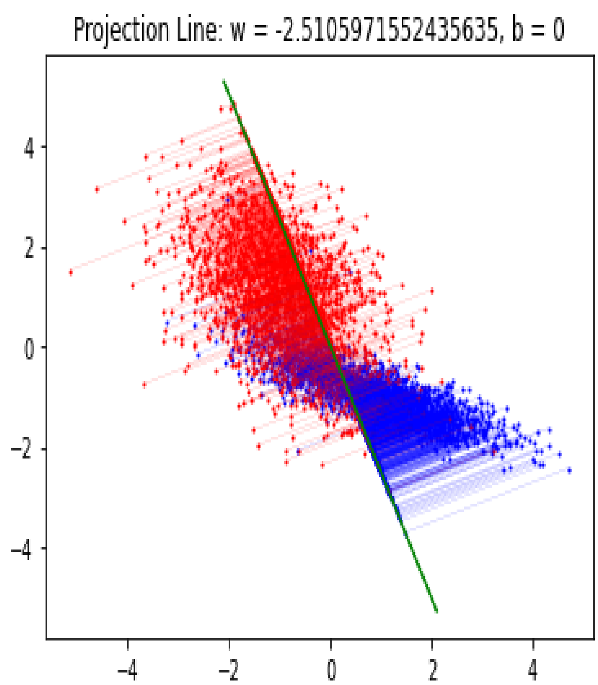
```
print(f" Fisher’s linear discriminant: {w}")
✓ 0.2s

Fisher’s linear discriminant: [[-0.000224   ]
 [ 0.00056237]]
```

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**)
   * For k = 2 or 4 and the voting number equal, then the decision goes to class 0.

   k = 1
   ```
   Accuracy of test-set 0.8488
   ```
   k = 2
   ```
   Accuracy of test-set 0.8704
   ```
   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)
   blue: class0, red: class1



Part. 2, Questions (40%):

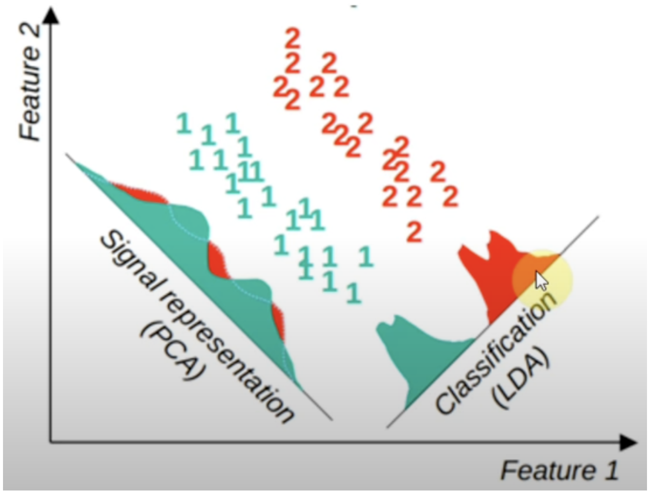
Please write/type by yourself. DO NOT screenshot the solution from others.

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

PCA is an ***unsupervised*** dimensionality reduction technique. It ignores the class label and aims to maximize the whole dataset's variation after projection(reduced the dimensions). The result from PCA might not be easy to separate different classes.

On the other hand, Fisher's Linear Discriminant is ***supervised*** dimensionality reduction technique. It takes the class label into consideration and aims to maximize the variance between different categories and minimize the variance within a class. We can expect the result from LDA is easier for us to separate different classes and the data of each class is more concentrated.

Here is an example that PCA and LDA give different results.



reference: [Linear Discriminant Analysis (LDA) vs Principal Component Analysis (PCA) - YouTube]
[LDA vs. PCA – Towards AI]

(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).
considering multi = K

The input space D is greater than K (class number) (D doesn't need to equal to K)
And the expected output space is D'
[original 2-class project to 1D space]

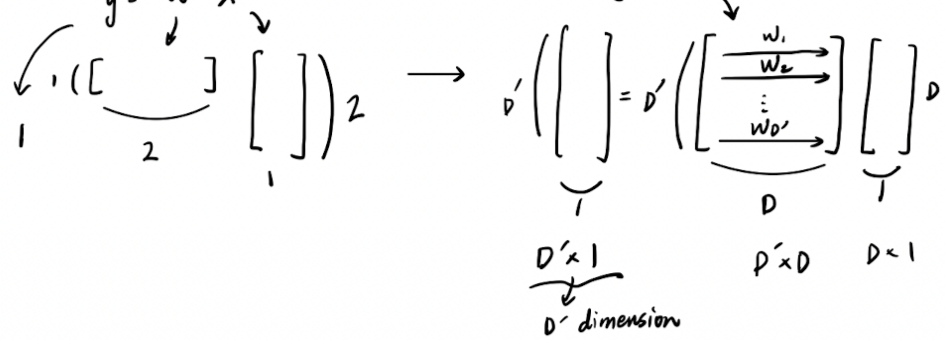
① Change the weight vector w from 2×1 (2-class) to $D \times D'$ (multi-class) (D)

2-class: project 2 class to 1D

$y = w^T x$

multi-class: project multiple class to D' dimension

$y = W^T x$



② S_W : Within class covariance matrix when $K > 2$

2-class

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

$\longrightarrow$

multi-class

need to sum up all class's within class variance

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

$$m_k = \frac{1}{N_k} \sum_{n \in C_k} x_n$$

③ S_B: Between class covariance matrix when $K > 2$

2-class

$$S_B = (\underline{m_2} - \underline{m_1})(m_2 - m_1)^T$$

class 1 class 2 互減

multi-class

$$S_B = \sum_{k=1}^{K} N_k (m_k - m)(m_k - m)^T$$

class k 内の資料数量

$$m = \frac{1}{N}\sum_{n=1}^{N} x_n$$

所有資料的平均

改成各個 class 対總体平均の相差

→ Consider the case where FLD projects data to a 1D space $D' = 1$

$$J(w) = \frac{w^T S_B w}{w^T S_W w}$$

An equivalent objective

$$\min_w -\frac{1}{2} w^T S_B w, \quad s.t. \; w^T S_W w = 1$$

(分母)

Lagrangian function

$$L_p = -\frac{1}{2} w^T S_B w + \frac{1}{2}\lambda (w^T S_W w - 1)$$

we have $S_B w = \lambda S_W w \Rightarrow S_W^{-1} S_B w = \lambda w$

The optimal w is the eigenvector of $S_W^{-1} S_B$ that corresponds to the largest eigenvalue

---

→ Consider the case where FLD projects data to a multi dimensional space, $D' > 1$

We cannot directly extend the objective to learn a multi dimensional projection since

$$J(w) = \frac{w^T S_B w}{w^T S_W w} \begin{matrix} \to \text{矩陣} \\ \to \text{矩陣} \end{matrix} \Big\} \to \text{矩陣不可相除}$$

A choice of the objective is

$$J(w) = \mathrm{Tr}\{(W S_W W^T)^{-1} (W S_B W^T)\}$$

The columns of the optimal W are the eigenvectors of $S_W^{-1} S_B$ that correspond to the D' largest eigenvalues.

(找前D'大の eigenvalue 对应到の eigenvector)

(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).

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

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

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

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

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

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

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

3.

3. prove

*顏色不一樣是不同の變數 (given by Q)

(b) $$J(w) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} = J_w = \frac{w^T S_B w}{w^T S_w w}$$

① 分子

$$(m_2-m_1)^2 = \left[w^T (m_2 - m_1)\right]^2$$

$$= \left[w^T m_2 - w^T m_1\right]^2$$

看成 1x1 matrix (symmetric, ∴ $A^2 = AA^T$)

$$= (w^T m_2 - w^T m_1)(w^T m_2 - w^T m_1)^T$$

$$= \left[w^T (m_2 - m_1)\right] \left[w^T (m_2 - m_1)\right]^T$$

$(AB)^T = B^T A^T$

$$= \left[w^T (m_2 - m_1)\right] \left[(m_2 - m_1)^T w\right]$$

$$= w^T (m_2 - m_1)(m_2 - m_1)^T w$$

$$= w^T S_B w$$

② 分母

$$s_1^2 + s_2^2 = \sum_{n \in C_1} (y_n - m_1)^2 + \sum_{n \in C_2} (y_n - m_2)^2$$

$$= \sum_{n \in C_1} (w^T x_n - w^T x_1)^2 + \sum_{n \in C_2} (w^T x_n - w^T x_2)^2$$

from 分子 we know

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

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

$$= w^T S_w w$$

$$\text{Thus} \quad J(w) = \frac{w^T S_B w}{w^T S_W w} \quad \#$$

(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).

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

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

4.

$\because$ logistic sigmoid activation function

$\therefore$ $y_k = \dfrac{1}{1 + e^{-a_k}}$

& by chain rule

棘の負號

$$\frac{\partial E}{\partial a_k} = \frac{\partial E}{\partial y_k}\frac{\partial y_k}{\partial a_k} = -\left(\frac{t_k}{y_k} + \frac{t_k - 1}{1 - y_k}\right) \cdot \frac{e^{-a_k}}{(1 + e^{-a_k})^2}$$

$$= -\frac{t_k(1 - y_k) + y_k(t_k - 1)}{y_k(1 - y_k)} \cdot y_k(1 - y_k)$$

$$= -\frac{t_k - y_k}{y_k(1 - y_k)} \cdot y_k(1 - y_k) = y_k - t_k$$

$$* \; (\ln x)' = \frac{1}{x}, \quad \underbrace{\left(\frac{1}{1 + e^{-x}}\right)'}_{S'(x)} = \frac{e^{-x}}{(1 + e^{-x})^2}$$

$$= S(x)\,(1 - S(x))$$

証明

シグモイド関数の定義と 関数の商の微分により、

$$\begin{aligned} S'(x) &= \left(\frac{1}{1+e^{-x}}\right)' \\ &= \frac{e^{-x}}{(1+e^{-x})^2} \\ &= \frac{1}{1+e^{-x}} \frac{e^{-x}}{1+e^{-x}} \\ &= \frac{1}{1+e^{-x}} \left(1 - \frac{1}{1+e^{-x}}\right) \\ &= S(x)(1 - S(x)) \end{aligned}$$

が成り立つ。

(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).

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

5.

If we take K separate binary classification to perform, we can use a network having K outputs each of which has a logistic sigmoid activation function. Associated with each output is a binary class label $t_k \in \{0, 1\}$ where $k = 1, \cdots K$ (t_k為是否屬於 class k), we assume that the class label are independent, given input vector, the conditional distribution of the target is

$$p(t \mid x, w) = \prod_{k=1}^{K} y_k(x, w)^{t_k} \left[1 - y(x, w)\right]^{1-t_k}$$

Taking the negative logarithm of the corresponding likelihood function than gives the following error functi

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

where y_{nk} denotes $y_k(x_n, w)$

The binary target variables $t_k \in \{0, 1\}$ have a 1 of K coding scheme indicating the class, and the network output is interpreted as

$$y_k(x, w) = p(t_k = 1 \mid x)$$

leading to the following error function

$$E(w) = -\sum_{n=1}^{N} \sum_{k=1}^{K} t_{kn} \ln y_k(x_n, w)$$

($t_{nk} = 1$ 的代入)