NYCU Introduction to Machine Learning, Homework 2

109550027 紀竺均

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}", f"mean vector of class 2: {m2}")
✓ 0.6s                                                                Python
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}")
✓ 0.4s                                                                Python
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}")
✓ 0.6s                                                                Python
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**

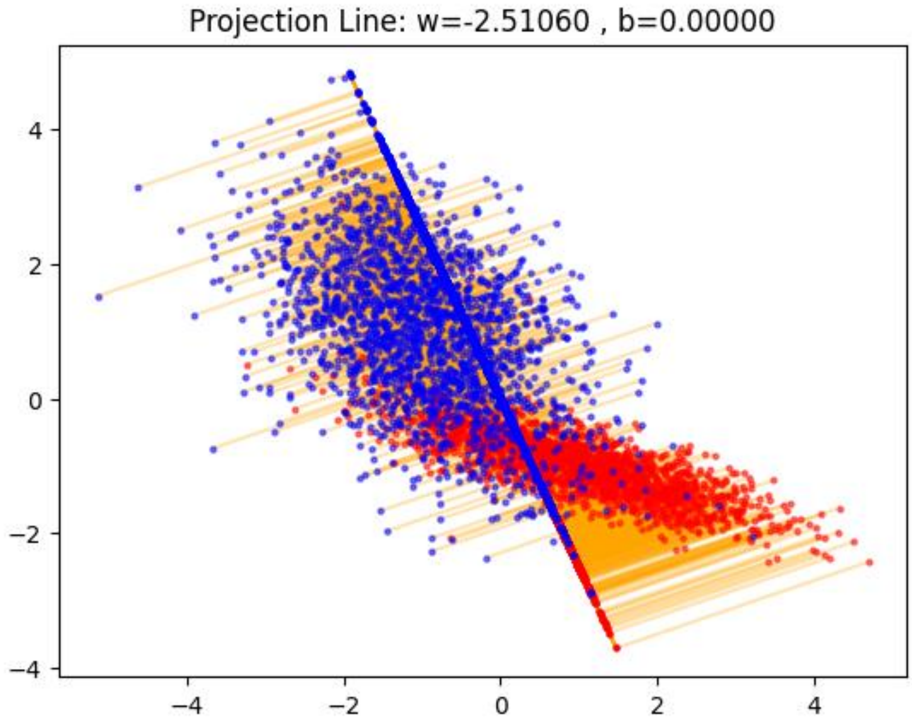
```
print(f" Fisher’s linear discriminant: {w}")
✓ 0.4s                                                                Python
Fisher’s linear discriminant: [-0.000224    0.00056237]
```

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**)

```
print("k = ", k, f"Accuracy of test-set {acc}")
✓ 36.4s                                                               Python
k =  1 Accuracy of test-set 0.8488
k =  2 Accuracy of test-set 0.8312
k =  3 Accuracy of test-set 0.8792
k =  4 Accuracy of test-set 0.8744
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
   2) colorize the data with each class

3) project all data points on your projection line.



Part. 2, Questions (40%):

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

Ans: PCA is an unsupervised dimensionality reduction method while FLD is a supervised technique that takes the class labels into account. Besides, FLD aims to minimize the within class variance and maximize the between class variance.

(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).

Ans:

2. $K=2$: $S_W = S_1 + S_2$

→ $K>2$: $S_W = \sum_{k=1}^{K} S_k$, $S_k = \sum_{n \in C_k} (x_n - m_k)(x_n - m_k)^T$

$K=2$: $S_B = (m_2 - m_1)(m_2 - m_1)^T$

$K>2$: $S_B = \sum_{k=1}^{K} N_k (m_k - m)(m_k - m)^T$

$\left(m_k = \frac{1}{N_k} \sum_{n \in C_k} x_n \; ; \; m = \frac{1}{N} \sum_{n=1}^{N} x_n\right)$

$K=2$: $J(w) = \frac{w^T S_B w}{w^T S_W w}$

$K>2$: $J(W) = Tr\{(W S_W W^T)^{-1}(W S_B W^T)\}$.

$K=2$: $w = S_W^{-1}(m_2 - m_1)$

$K>2$: W = eigenvector of $S_W^{-1} S_B$ that has Maximal 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).

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

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

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

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

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

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

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

Ans:

3. $$J(w) = \frac{(m_2-m_1)^2}{S_1^2+S_2^2} = \frac{(m_2-m_1)^2}{\sum_{n\in C_1}(y_n-m_k)^2+\sum_{n\in C_2}(y_n-m_k)^2} \quad (\text{by Eq } 5)$$

$$= \frac{(m_2-m_1)^2}{\sum_{n\in C_1}(w^T\cdot x_n-m_k)^2+\sum_{n\in C_2}(w^T x_n-m_k)^2} \quad (\text{by Eq } 1)$$

$$= \frac{[w^T(m_2-m_1)]^2}{\sum_{n\in C_1}(w^T\cdot x_n-w^T m_k)^2+\sum_{n\in C_2}(w^T x_n-w^T m_k)^2} \quad \begin{matrix}\xrightarrow{\text{by}} \text{Eq } 3 \\ \to \text{Eq } 4\end{matrix}$$

$$= \frac{(w^T)^2(m_2-m_1)^2}{(w^T)^2\cdot\left(\sum_{n\in C_1}(x_n-m_1)^2+\sum_{n\in C_2}(x_n-m_2)^2\right)}$$

$$= \frac{(w^T)^2\cdot S_B}{(w^T)^2\cdot S_W} \quad (\text{左右兩邊同乘 } \frac{w}{w}$$

$$= \frac{w^T\cdot S_B\cdot w}{w^T\cdot S_W\cdot w}.$$

(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).

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

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

Ans:

4. $\frac{\partial E}{\partial a_k} = \frac{\partial E}{\partial y_k} \cdot \frac{\partial y_k}{\partial a_k}$

$y_k = \sigma(a_k) = \frac{1}{1+e^{-a}}$

$\frac{\partial \sigma}{\partial a} = \frac{1}{(1+e^{-a})^2} \cdot (e^{-a}) = \frac{1}{(1+e^{-a})} \cdot \frac{e^{-a}}{(1+e^{-a})} = \sigma(1-\sigma)$

$\Rightarrow \frac{\partial y_k}{\partial a_k} = y_k(1-y_k)$

So, $\frac{\partial E}{\partial a_k} = -\left(t_k \cdot \frac{1}{y_k} + (1-t_k) \cdot \frac{1}{(1-y_k)} \cdot (-1)\right) \cdot \frac{\partial y_k}{\partial a_k}$

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

$= -t_k(1-y_k) + (1-t_k) \cdot y_k = y_k - t_k$

(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).

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

Ans:

5. For multiclass neural network:

conditional distribution: $p(t|w_1, w_2 \ldots w_K) = \prod_{k=1}^{K} y_k^{t_k}$

→ likelihood function: $p(T|w_1, w_2 \ldots w_K) = \prod_{n=1}^{N} \prod_{k=1}^{K} y_{nk}^{t_{nk}}$ (for N points)

→ negative log likelihood:

$-\ln(P(T|w_1, \ldots w_K)) = -\ln\left(\prod_{n=1}^{N} \prod_{k=1}^{K} y_{nk}^{t_{nk}}\right)$

$= -\sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} \cdot \ln y_{kn}$

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