NYCU Introduction to Machine Learning,Homework 2

Deadline: Nov. 1, 23:59

Part. 1, Coding (60%):

In this coding assignment, you are required to implement Fisher's linear discriminant by using only NumPy, then train your model on the provided dataset, and evaluate the performance on testing data. Find the sample code and data on the GitHub page https://github.com/NCTU-VRDL/CS_CS20024/tree/main/HW2

Please note that only NumPy can be used to implement your model, you will get 0 point by calling sklearn.discriminant_analysis.LinearDiscriminantAnalysis.

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]]
```

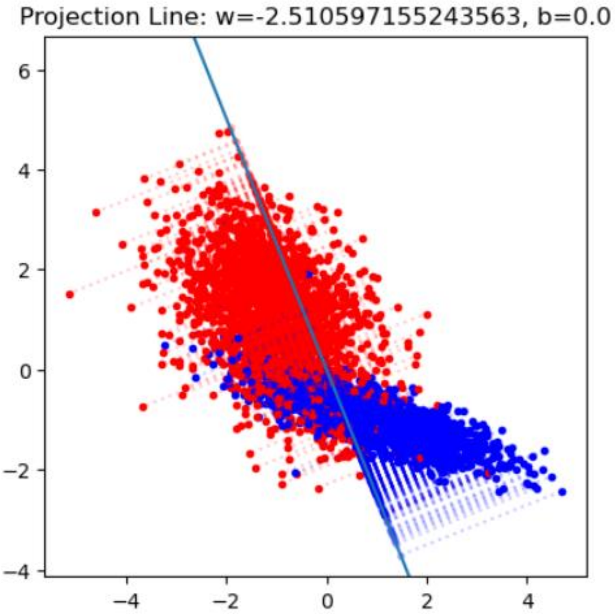
5. (20%) Project the testing data by Fisher's linear discriminant to get the class prediction by K-Nearest-Neighbor rule and report the accuracy score on testing data with K values from 1 to 5 (you should get accuracy over 0.88)

```
0.8488
0.8704
0.8792
0.8824
0.8912
```

```
print(f"Accuracy of test-set {acc}")
```

```
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?

A:Principle Component Analysis(PCA) doesn’ t need to know which class the data belongs to, while fisher’ s Linear Discriminant(FLD) need to know which class the data belongs to, PCA only consider how to maximize the variance of the data, but FLD not only consider to maximize the variance from different class but to minimize the variance of the same class.

(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).

2. We extend S_w and S_b as follows

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

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

To get the maximum value of $J(w) = \frac{w^T S_B w}{w^T S_w w}$, we extend the function as follows:

s.t $= w^T S_w w = 1$

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

$\Rightarrow S_B w = \lambda S_w w \Rightarrow S_w^{-1} S_B w = \lambda w$

by finding the best w corresponding to the maximum λ, we find the best vector for projection.

(6%) 3. By making use of Eq (1) ~ Eq (5), show that the Fisher criterion Eq (6) can be written in t he 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. $S_k^2 = \sum_{n\in C_k} (y_n - m_k)^2 = \sum_{n\in C_k} (W^T x - W^T m_k) = \sum_{n\in C_k} W^T (X - m_k)(X - m_k)^T W$

$= W^T S_k W$

$\Rightarrow S_1^2 + S_2^2 = W^T S_1 W + W^T S_2 W$

$= W^T S_W W$

$(m_2 - m_1)^2 = (W^T m_2 - W^T m_1)^2$

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

$= W^T S_B W$

$\Rightarrow J(w) = \dfrac{(m_2 - m_1)^2}{S_1^2 + S_2^2} = \dfrac{W^T S_B W}{W^T S_W W}$ #

(7%) 4. Show the derivative of the error function Eq (8) with respect to the activation a_k for an ou tput 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. $E(w) = -\sum_{n=1}^{N} \{t_n \ln y_n + (1-t_n) \ln(1-y_n)\}$, where $y = \sigma(a_n)$

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

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

$$= y_k - t_k$$

(7%) 5. Show that maximizing likelihood for a multiclass neural network model in which the netw ork outputs have the interpretation $y_k(x, w) = p(t_k = 1 \mid x)$ is equivalent to the minimization of

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

the cross-entropy error function Eq (10).

5. $P(t_k=1|X) = \prod_{n=1}^{N}\prod_{k=1}^{K} P(C_k|X)^{t_{nk}} = \prod_{n=1}^{N}\prod_{k=1}^{K} y_{nk}^{t_{nk}}$

to find the maximum value of $E(w_1, \dots w_K)$, we find the minimum value for negative log likelyhood function $-\ln P(t_k=1|X) = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \ln y_{nk}$

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