NYCU Pattern Recognition, Homework 2

[312552056], [鄭璟翰]

Part. 1, Coding (60%):

(25%) Logistic Regression w/ Gradient Descent Method (slide ref)

1. (0%) Show the hyperparameters (learning rate and iteration, etc) that you used

```
    LR = LogisticRegression(
        learning_rate=1.5e-2,  # You can modify the parameters as you want
        num_iterations=10000,  # You can modify the parameters as you want
    )
```

2. (5%) Show your weights and the intercept of your model.

```
2024-04-22 17:00:57.745 | INFO     | __main__:main:134 - LR: Weight
s: [-0.67822499  0.33390716  0.66008689 -0.05974767  0.31010599], I
ntercep: -2.2552486986688
```

3. (5%) Show the AUC of the classification results on the testing set.

```
2024-04-22 17:00:57.746 | INFO     | __main__:main:135 - LR: Accura
cy=0.8095, AUC=0.8614
```

4. (15%) Show the accuracy score of your model on the testing set.
   Show as previous screenshot.

(25%) Fisher Linear Discriminant, FLD (slide_ref)

5. (0%) Show the mean vectors mi (i=0, 1) of each class of the training set.

```
2024-04-22 17:00:57.749 | INFO     | __main__:main:148 - FLD: m0=[[
 0.35994138]
 [-0.04560139]], m1=[[0.32519126]
 [0.04435118]]
```

6. (5%) Show the within-class scatter matrix *Sw* and between-class scatter matrix *Sb* of the training set.

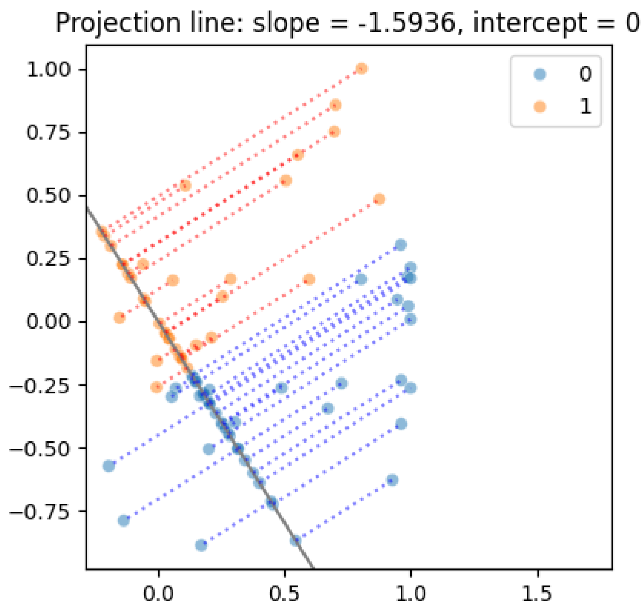
```
2024-04-22 17:00:57.750 | INFO     | __main__:main:149 - FLD:
Sw=
[[0.50298617 0.18634845]
 [0.18634845 0.45157657]]
2024-04-22 17:00:57.751 | INFO     | __main__:main:150 - FLD:
Sb=
[[ 0.00120757 -0.00312586]
 [-0.00312586  0.00809147]]
```

7. (5%) Show the Fisher's linear discriminant *w* of the training set.

```
2024-04-22 17:00:57.752 | INFO     | __main__:main:151 - FLD:
w=
[[-0.53152354]
 [ 0.84704352]]
```

8. (15%) Obtain predictions for the testing set by measuring the distance between the projected value of the testing data and the projected means of the training data for the two classes. Show the accuracy score on the testing set.

```
2024-04-22 17:00:57.753 | INFO     | __main__:main:152 - FLD: Accur
acy=0.7143
```



(10%) Code Check and Verification

9. (10%) Lint the code and show the PyTest results.

```
zjinghan@zjinghan-VivoBook-ASUSLaptop-X512JP-X512JP:~/Desktop/NYCU/
Pattern Recognition/HW2/release$ pytest
======================= test session starts ========================
platform linux -- Python 3.10.12, pytest-8.1.1, pluggy-1.4.0
rootdir: /home/zjinghan/Documents/NYCU/Pattern Recognition/HW2/rele
ase
collected 2 items

test_main.py ..                                               [100%]

======================== 2 passed in 5.60s =========================
zjinghan@zjinghan-VivoBook-ASUSLaptop-X512JP-X512JP:~/Desktop/NYCU/
Pattern Recognition/HW2/release$ flake8 main.py
zjinghan@zjinghan-VivoBook-ASUSLaptop-X512JP-X512JP:~/Desktop/NYCU/
Pattern Recognition/HW2/release$
```

Part. 2, Questions (40%):

1. (10%) Is it suitable to use Mean Square Error (MSE) as the loss function for Logistic Regression? Please explain in detail.

   MSE is not suitable to be used as the loss function for Logistic Regression. The reason is that the output of Logistic Regression represents probabilities. If MSE is used to compute the loss, it calculates the square of the difference between the probability (a value between 0 and 1) and the label (0 or 1). Regardless of whether the prediction is correct or not, such a loss will always be between 0 and 1, failing to reflect prediction errors. In contrast, using Binary Cross-Entropy as the loss function allows for amplifying the difference between the probability and the label through the logarithmic term, effectively reflecting the cost of prediction errors in the loss function.

2. (15%) In page 31 of the lecture material (linear_classification.pdf), we introduce two methods for performing classification tasks using Fisher's linear discriminator: 1) Determining a threshold, 2) Using the k-NN (k-nearest neighbors) rule. Please discuss at least three aspects, either advantages or disadvantages, of using the k-NN method compared to determining a threshold (resources, performance, etc.).

   (1) Different hyperparameters: k-NN requires determining k (an odd integer), while the other method requires determining a threshold. It is obvious that determining the hyperparameter for k-NN is simpler, as it only requires deciding an integer value. However, determining the threshold may require selecting a real-valued number, possibly necessitating adjustments through statistical methods (such as averaging or mode).

   (2) The execution times: k-NN relies on voting from the training data and requires execution for each testing data point, resulting in a time complexity of the size of the training data * the size of the testing data. On the other hand, the other method only requires determining the threshold, resulting in an execution time complexity proportional to the size of the testing data.

   (3) k-NN tends to achieve higher accuracy: During the classification process, k-NN considers the distribution of the training data and classifies based on the results of the training data. Thus, if the data has a mixed distribution in two classes, k-NN can better reflect the distribution of the training data. In contrast, the threshold method directly separates the two cases, which might lead to higher chances of misclassification when the data distribution is mixed.

3. (15%) In logistic regression, what is the relationship between the sigmoid function and the softmax function? In what scenarios will the two functions be used respectively?

   Sigmoid function.

$$\sigma(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1}$$
$$1-\sigma(x) = \frac{1}{e^x+1}$$

   Softmax function.

$$\sigma(x)_i = \frac{e^{x_i}}{\sum e^{x_j}}$$

   The Softmax function is an extension of the Sigmoid function. The Sigmoid function represents the probability of belonging to one class, with the probability of belonging to the other class being 1 minus that value. On the other hand, the Softmax function represents the results of each category when there are multiple categories. Therefore, the Sigmoid function is suitable when there are only two categories in the classification result, while the Softmax function is suitable when there are three or more categories in the classification result.