NYCU Pattern Recognition, Homework 2

[312551814], [Malik DAHMANI]

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

```
def __init__(self, learning_rate: float = 1e-3, num_iterations: int = 1000):
```

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

```
2024-04-24 12:55:31.013 | INFO     | __main__:main:163 - LR: Weights: [-0.76173996 -0.45533988  1.33021566 -1.09893384  0.15278147], Interc
ept: -0.042622180865445745
```

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

```
| INFO     | __main__:main:164 - LR: AUC=0.8295
```

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

```
| INFO     | __main__:main:165 - LR: Accuracy=0.8333
```

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

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

```
| INFO     | __main__:main:175 - 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.

```
2024-04-24 17:34:18.203 | INFO     | __main__:main:179 - FLD: Sw=
[[41.93041055 15.7202037 ]
 [15.7202037  37.25186904]]
2024-04-24 17:34:18.203 | INFO     | __main__:main:180 - FLD: Sb=
[[ 0.00120757 -0.00312586]
 [-0.00312586  0.00809147]]
```

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

```
2024-04-24 17:34:18.204 | INFO     | __main__:main:181 - FLD: w=
[-0.53138255  0.84713198]
```

8. (15%) Obtain predictions for the testing set by measuring the distance between the projected value of the testing data and the projected m

eans of the training data for the two classes. Show the accuracy score on the testing set.

```
| INFO     | __main__:main:181 - FLD: Accuracy=0.7381
```

(10%) Code Check and Verification

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

```
============================= test session starts =============================
platform win32 -- Python 3.12.3, pytest-8.1.1, pluggy-1.5.0
rootdir: h:\Desktop\NYCU\Pattern recognition\DM2
collected 2 items

test_main.py ..                                                          [100%]

============================== 2 passed in 5.63s ==============================
Finished running tests!
```

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.

Mean Square Error (MSE) is not a good loss function for logistic regression for several reasons. First, when we run logistic regression, we are looking for the probability that an object belongs to a particular class based on its attributes. Thus, the output of the model is a probability between 0 and 1, whereas the MSE measures the error between continuous values, which doesn't correspond to the problem formulation.

Moreover, MSE is not convex because logistic regression is nonlinear due to the sigmoid function. Therefore, it will be difficult to find the global minimum.

Instead of MSE, logistic regression typically uses the cross-entropy loss as the appropriate 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.).

First, an advantage of using the k-NN method over the threshold method is that in the threshold method, the decision is based on a single criterion. In the k-NN method, the number of neighbors can be adjusted according to the data, making it more flexible. This makes it more adaptable to different data sets.

Another advantage of the k-NN method is that it is more resistant to noise and unbalanced data sets because it relies on local information to make predictions. With the threshold method, noise in the data set can affect the classification and misclassify certain values.

A disadvantage of k-nearest neighbors is that it requires more resources than the threshold method, to store and process all the training data. As the size and dimensionality of the data increases, calculating the distance and finding the nearest neighbors becomes costlier. For very large datasets, this can result in significant memory consumption, which can be a limiting factor in certain application contexts.

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?

The relationship between the Sigmoid function and the SoftMax function is that they are both activation functions. Their formulas are:

$$\text{Softmax } \sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

$$\text{Sigmoid } S(x) = \frac{1}{1+e^{-x}}$$

Furthermore, both functions are sigmoidal, and in the binary case, the SoftMax reduces to the sigmoidal function.

Sigmoid function is used for binary classifications, i.e. when there are only two classes. The sigmoid function returns the probability of belonging to class 1.

SoftMax function applies to multi-class problems. It takes as parameters a vector with the same number of entries as classes and returns the probability of belonging to each class, while checking that the sum of probabilities is equal to 1.