NYCU Introduction to Machine Learning, Homework 2

[111550012], [鄭睿宏]

Part. 1, Coding (60%):

(25%) Logistic Regression w/ Gradient Descent Method

1. (5%) Show the hyperparameters (learning rate and iteration, etc) that you used and the weights and intercept of your model.

```
LR = LogisticRegression(
    learning_rate=0.03,
    num_iterations=1000,
)
```

2. (5%) Show the AUC of the classification results on the testing set.

```
2024-10-23 21:22:48.052 | INFO     | __main__:main:200 - LR: Accuracy=0.8095, AUC=0.8477
```

3. (15%) Show the accuracy score of your model on the testing set

```
2024-10-23 21:22:48.052 | INFO     | __main__:main:200 - LR: Accuracy=0.8095, AUC=0.8477
```

(25%) Fisher Linear Discriminant, FLD

4. (5%) Show the mean vectors m_i (i=0, 1) of each class, the within-class scatter matrix *Sw*, and the between-class scatter matrix *Sb* of the training set.

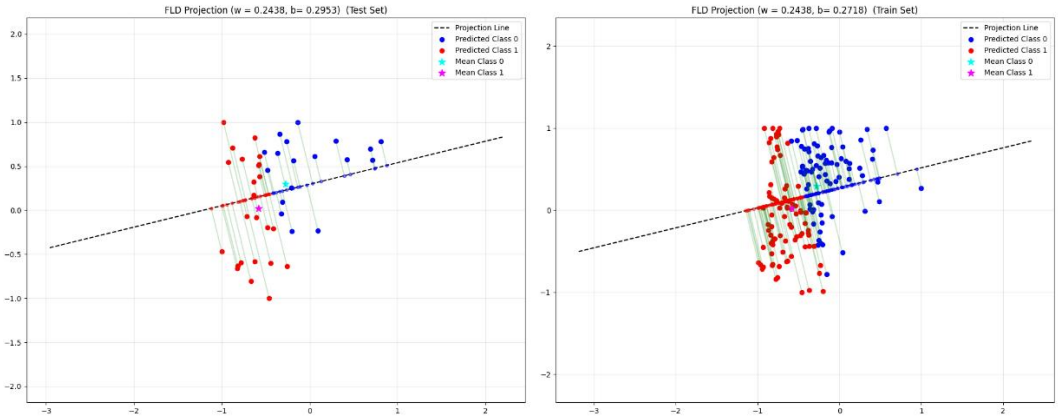
```
2024-10-23 21:22:48.053 | INFO     | __main__:main:223 - FLD: m0=[-0.27747695  0.29565197], m1=[-0.58535466  0.02331584] of cols=['10', '20']
2024-10-23 21:22:48.053 | INFO     | __main__:main:224 - FLD:
Sw=
[[17.17974856  5.44299487]
 [ 5.44299487 44.81848741]]
2024-10-23 21:22:48.053 | INFO     | __main__:main:225 - FLD:
Sb=
[[0.09478869 0.08384622]
 [0.08384622 0.07416696]]
```

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

```
2024-10-23 21:22:48.053 | INFO     | __main__:main:226 - FLD:
w=
[-0.0166359  -0.00405607]
```

6. (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. (Also, plot for training data). Show the accuracy score on the testing set.

```
2024-10-23 21:22:48.053 | INFO     | __main__:main:227 - FLD: Accuracy=0.7619
```



(10%) Code Check and Verification

7. (10%) Lint the code and show the PyTest results.

Result of linting code

```
(mlenv) C:\Users\Ray\Downloads\HW2>flake8 main.py

(mlenv) C:\Users\Ray\Downloads\HW2>
```

PyTest results

```
(mlenv) C:\Users\Ray\Downloads\HW2>pytest ./test_main.py -s
============================================================ test session starts ============================================================
platform win32 -- Python 3.11.9, pytest-8.3.3, pluggy-1.5.0
rootdir: C:\Users\Ray\Downloads\HW2
collected 2 items

test_main.py (395, 2) (395,)
2024-10-23 21:16:52.917 | INFO     | test_main:test_logistic_regression:35 - accuracy=0.9517
.(395, 2) (395,)
2024-10-23 21:16:52.926 | INFO     | test_main:test_fld:45 - accuracy=0.8759
.

============================================================= 2 passed in 2.71s =============================================================
```

Part. 2, Questions (40%):

1. (10%)
   - Is logistic ‘regression’ used for regression problems?
   - If not, what task is it primarily used? (without any additional techniques and modification); If yes, how can it be implemented?
   - Why are we using the logistic function in such a task? (list two reasons)
   - If there are multi-class, what should we use to substitute it?

   No, logistic regression is mainly used for classification problems. It is appropriate for describing probabilities since it uses the logistic function to make sure the expected output is between 0 and 1. Additionally, by introducing non-linearity, this function helps the model capture more intricate correlations between the target variable and features. Multinomial logistic regression (also known as softmax regression), is commonly employed for multi-class classification issues. To manage more than two classes, binary logistic regression is extended.

2. (15%) When a trained classification model shows exceptionally high precision but unusually low recall and F1-score, what potential issues might arise? How can these issues be resolved? List at least three solutions.

   Potential Issues: The model's high probability threshold for positive predictions may be too conservative, resulting in low positive examples and high recall.
   Solution: Lowering the decision threshold for positive predictions could improve recall and F1-score, though it may decrease precision. This will allow the model to catch more positive instances, reducing the missed positives.

   Potential Issues: Class imbalance, which can lead to a biased model, reducing recall.
   Solution: To address this, techniques like oversampling or under sampling the minority class or assigning class weights can be used to ensure more attention to the positive class.

   Potential Issues: The model may have overfitted to the negative class, leading to low recall. **Solution:** To improve recall, focus on a metric that considers both precision and recall, such as the F1-score, or directly improve recall by tuning hyperparameters or cross-validation.

3. (15%) In this homework, we use Cross-Entropy as the loss function for Logistic Regression. Can we use Mean Square Error (MSE) instead? Why or why not? Please explain in detail.

No, there are several reasons why Mean Squared Error (MSE) is inappropriate. The first is the nature of the problem; in regression tasks, where the objective is to predict a continuous value, MSE is frequently utilized. It calculates the mean squared difference between the actual and projected values. Yet, the goal of logistic regression, a classification problem, is to predict discrete class labels. The second one is gradient behavior, although both MSE and CE can reach a minimum value of 0 if classified correctly, but there is a big difference, especially as predictions approach the correct class. The model will undergo the sigmoid function in range between 0 and 1 in training. Cross-Entropy Loss produces larger gradients when the model is far from the correct prediction, allowing faster updates. Mean Squared Error produces smaller gradients as the model improves, causing slower convergence and the vanishing gradient problem, making it less effective in classification tasks.