NYCU Introduction to Machine Learning, Homework 2

111550035, Yun-Cheng Tsai

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=5e-3,
    num_iterations=15000,
)
```

```
2024-10-26 20:35:20.214 | INFO     | __main__:main:154 - LR: Weights: [-0.50762019 -0.15151763
0.12130179 -0.50550986 -0.22863936], Intercept: -4.449611574123004
```

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

```
AUC=0.8818
```

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

```
LR: Accuracy=0.8571
```

(25%) Fisher Linear Discriminant, FLD

4. (5%) Show the mean vectors m_i (i=0, 1) of each class, the within-class scatter

```
2024-10-26 23:18:52.418 | INFO     | __main__:main:174 - FLD: m0=[-0.27747695  0.29565197], m1=[-0.58535466  0.02331584] of cols=['10', '20']
2024-10-26 23:18:52.418 | INFO     | __main__:main:175 - FLD:
Sw=
[[17.17974856  5.44299487]
 [ 5.44299487 44.81848741]]
2024-10-26 23:18:52.418 | INFO     | __main__:main:176 - FLD:
Sb=
[[0.09478869 0.08384622]
 [0.08384622 0.07416696]]
```

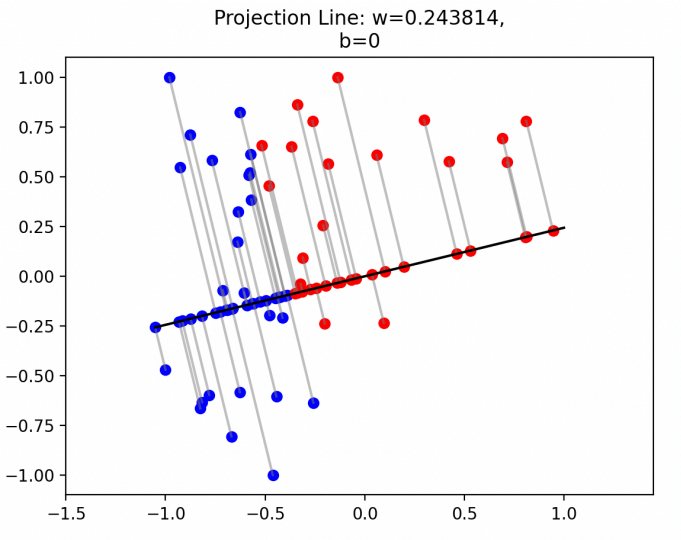
matrix *Sw,* and the between-class scatter matrix *Sb* of the training set.

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

```
2024-10-26 23:18:52.418 | INFO     | __main__:main:177 - FLD:
w=
[-0.97154001 -0.23687549]
```

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.

```
2024-10-26 20:35:20.215 | INFO     | __main__:main:173 - FLD: Accuracy=0.7619
```



(10%) Code Check and Verification

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

```
❯ flake8 main.py
```

```
❯ pytest test_main.py -s
============================= test session starts ==============================
platform darwin -- Python 3.9.20, pytest-8.3.3, pluggy-1.5.0
rootdir: /Users/tsai_m/Desktop/ML/HW2
collected 2 items

test_main.py (395, 2) (395,)
2024-10-26 20:41:11.012 | INFO     | test_main:test_logistic_regression:35 - accuracy=0.9793
.(395, 2) (395,)
2024-10-26 20:41:11.013 | INFO     | test_main:test_fld:45 - accuracy=0.9172
.

============================== 2 passed in 4.77s ===============================
```

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?

(1) No, logistic regression is used for classification problems (especially binary).
(2) Logistic regression is primarily used for binary classification tasks.
(3) First reason is that the logistic function maps any real number to a value between 0 and 1, this allow to interpret the result as a probability, and probability threshold may help to determine which class the data belongs to, giving a natural probabilistic view of class predictions. Second reason is that it is very fast at classifying unknown records and also easier to implement, interpret, and very efficient to train..
(4) We may use softmax regression, this is an approach that extend logistic regression to handle more than 2 classes at the same time.

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.

(1) This may due to unbalanced dataset, which means the data size of has a large difference between two classes. In addition, high threshold may made the decision too careful, which means a data is hard to be determined as positive class.
(2) Solutions:
   i. Adjust (Lower) the threshold, example: from 0.5 to 0.4 or lower.
   ii. Under-sampling and Over-sampling, increase or decrease the unbalanced class’s data size to balance the whole dataset.
   iii. Add class weight to the loss function, this make classes with smaller data size has higher influence to the model.

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.

(1) No
(2) MSE is not possible because the task is not convex, continuous or differentiable. In addition, there is not a closed form solution, like in logistic regression. This means that algorithms like gradient descent won't be able to find an optimal set of weights for the discrete results. As a result, gradient descent might only be able to find a local minimum for the function or be unable to find any minimum.