NYCU Introduction to Machine Learning, Homework 2

Deadline: Nov. 14, 23:59

Part. 1, Coding (50%):

109612019 林伯偉

In this coding assignment, you are requested to implement Logistic Regression and Fisher's Linear Discriminant by using only Numpy. After that, train your model on the provided dataset and evaluate the performance on the testing data.

(15%) Logistic Regression

Requirements:

- Use Gradient Descent to update your model
- Use CE (Cross-Entropy) as your loss function.

Criteria:

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

```
LR = LogisticRegression(learning_rate=0.000001, iteration=120000)
```

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

```
Part 1: Logistic Regression
Weights: [-0.05397537 -1.16311766  0.98944369 -0.11357452  0.03156962 -0.65005565], Intercept: -0.09744106185511861
```

3. (10%) Show the accuracy score of your model on the testing set. The accuracy score should be greater than 0.75.

```
Accuracy: 0.7540983606557377
```

(35%) Fishers Linear Discriminant (FLD)

Requirements:

- Implement FLD to reduce the dimension of the data from 2-dimensional to 1-dimensional.

Criteria:

4. (0%) Show the mean vectors m_i (i=0, 1) of each class of the training set.

```
Class Mean 0: [ 56.75925926 137.7962963 ], Class Mean 1: [ 52.63432836 158.97761194]
```

5. (5%) Show the within-class scatter matrix S_W of the training set.

```
With-in class scatter matrix:
[[ 19184.82283029 -16006.39331122]
 [-16006.39331122 106946.45135434]]
```

6. (5%) Show the between-class scatter matrix S_B of the training set.

```
Between class scatter matrix:
[[ 17.01505494 -87.37146342]
 [-87.37146342 448.64813241]]
```

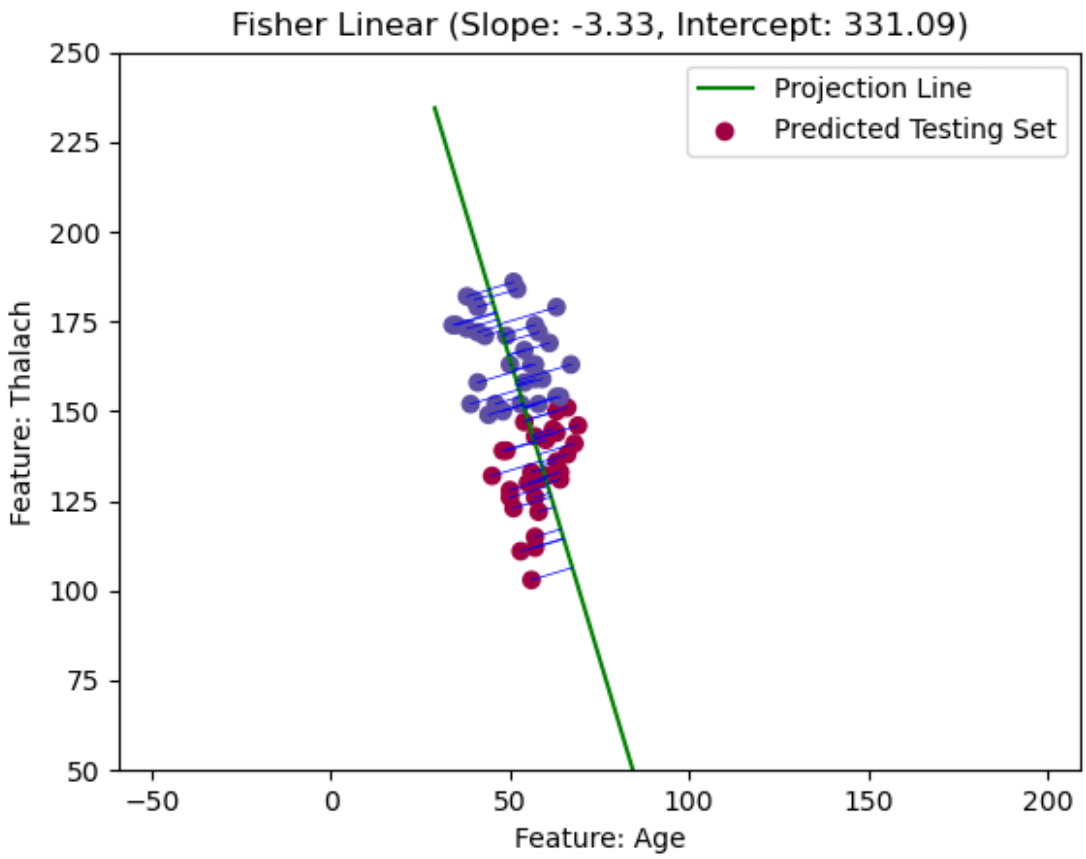
7. (5%) Show the Fisher's linear discriminant w of the training set.

```
w:
[ 0.28737344 -0.95781862]
```

8. (10%) 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. The accuracy score should be greater than 0.65.

```
Accuracy of FLD: 0.6557377049180327
```

9. (10%) Plot the projection line (x-axis: age, y-axis: thalach).
   1) Plot the projection line trained on the training set and show the slope and intercept on the title (you can choose any value of intercept for better visualization).
   2) Obtain the prediction of the testing set, plot and colorize them based on the prediction.
   3) Project all testing data points on your projection line.



Part. 2, Questions (50%):

1. (5%) What's the difference between the sigmoid function and the softmax function? In what scenarios will the two functions be used? Please at least provide one difference for the first question and answer the second question respectively.

   Ans:

   - The main difference between the sigmoid and softmax functions lies in their application and the number of classes they are designed to handle.
     The sigmoid function is commonly used in binary classification problems where there are only two classes (0 or 1). It squashes the output between 0 and 1, representing the probability of belonging to the positive class.
     The softmax function is used in multiclass classification problems where there are more than two classes. It calculates the probabilities of each class, and the class with the highest probability is chosen as the predicted class.

   - The sigmoid function is used when dealing with binary classification problems, where the outcome is either 0 or 1. For example, in logistic regression for predicting whether a question is True (1) or False (0). The softmax function is used when dealing with multiclass classification problems, where there are more than two classes. For instance, in image recognition tasks where an algorithm needs to classify an image into multiple categories.

2. (10%) In this homework, we use the cross-entropy function as the loss function for Logistic Regression. Why can't we use Mean Square Error (MSE) instead? Please explain in detail.

   Ans:

   - **Sensitivity to outliers:**
     MSE is sensitive to outliers or extreme values. In classification problems, particularly when dealing with probabilities, extreme values (close to 0 or 1) may be produced. MSE heavily penalizes these large errors, which might not be suitable for classification tasks where small deviations in probabilities are acceptable.

   - **Incompatibility with probability outputs:**
     MSE does not take into account the specific characteristics of probability outputs. In logistic regression, the output is a probability, and the sigmoid function is used to squash the predictions between 0 and 1. MSE, designed for continuous variables, may not appropriately penalize deviations in probabilities, especially near the boundaries.

- **Gradient descent challenges:**
  The gradient of MSE can be small, especially when the predicted probability is close to the actual class label (0 or 1). This can slow down the convergence of optimization algorithms like gradient descent. The optimization process may become inefficient, and the model might take longer to learn.

- **Bias toward large errors:**
  MSE gives high weight to large errors. In classification tasks, especially when dealing with imbalanced datasets, a small error in predicting the correct class (e.g., predicting 0.1 when the true label is 0) should not be heavily penalized. MSE, by nature, tends to focus on reducing large errors, which may not align with the goals of classification.

3. (15%) In a multi-class classification problem, assume you have already trained a classifier using a logistic regression model, which the outputs are P1, P2, ... Pc, how do you evaluate the overall performance of this classifier with respect to its ability to predict the correct class?

   3.1. (5%) What are the metrics that are commonly used to evaluate the performance of the classifier? Please at least list three of them.
   Ans: Precision, Sensitivity, Accuracy.

   3.2. (5%) Based on the previous question, how do you determine the predicted class of each sample?
   Ans: The predicted class for each sample is determined by selecting the class with the highest probability among P1, P2, ..., Pc. In other words, the class with the maximum predicted probability is chosen as the predicted class.

   3.3. (5%) In a class imbalance dataset (say 90% of class-1, 9% of class-2, and 1% of class-3), is there any problem with using the metrics you mentioned above and how to evaluate the model prediction performance in a fair manner?
   Ans: In imbalanced datasets, a model might achieve high accuracy by simply predicting the majority class. This is misleading and doesn't reflect the true predictive performance. It's better to examine the confusion matrix to understand the distribution of true positives, false positives, true negatives, and false negatives for each class and plot the precision-recall curves to visualize the trade-off between precision and recall for different probability thresholds which is particularly useful in imbalanced datasets.

4. (20%) Calculate the results of the partial derivatives for the following equations. (The first one is binary cross-entropy loss, and the second one is mean square error loss followed by a sigmoid function. σ is the sigmoid function.)

4.1. (10%)

$$\frac{\partial}{\partial x}\left(-t * \ln(\sigma(x)) - (1-t) * \ln(1-\sigma(x))\right)$$

$$L(x) = \frac{\partial}{\partial x}\left(-t \cdot \ln(\sigma(x))\right) - (1-t) \cdot \ln(1-\sigma(x))$$

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

$$x = w_0 + x_1 \times w_1 + x_2 \times w_2 + \cdots + x_k \times w_k = w^T x$$

$$\frac{\partial L(x)}{\partial x} = \frac{\partial L(x)}{\partial \sigma(x)} \frac{\partial \sigma(x)}{\partial x} \frac{\partial x}{\partial x}$$

$$\frac{\partial L(x)}{\partial \sigma(x)} = -\left(\frac{t}{\sigma(x)} - \frac{1-t}{1-\sigma(x)}\right) = \frac{\sigma(x)-t}{\sigma(x)(1-\sigma(x))}$$

$$\frac{\partial \sigma(x)}{\partial x} = -(1+e^{-x})^{-2}(-e^{-x}) = \frac{e^{-x}}{(1+e^{-x})^2} = \frac{1 \cdot e^{-x}}{(1+e^{-x})(1+e^{-x})}$$

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

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

$$\therefore \frac{\partial L(x)}{\partial x} = \frac{\sigma(x)-t}{\sigma(x)(1-\sigma(x))} \cdot \sigma(x)(1-\sigma(x))$$

$$= \sigma(x) - t \quad //$$

4.2. (10%)

$\frac{\partial}{\partial x}((t-\sigma(x))^2)$

$$\frac{\partial}{\partial x}\left((t-\sigma(x))^2\right) = 2(t-\sigma(x))\frac{\partial}{\partial x}(t-\sigma(x))$$

$$\frac{\partial}{\partial x}(t-\sigma(x)) = \frac{\partial t}{\partial x} - \frac{\partial \sigma(x)}{\partial x}$$

$$= 0 - \sigma(x)(1-\sigma(x)) \text{ 代入}$$

$$\Rightarrow -2(t-\sigma(x))\sigma(x)(1-\sigma(x))$$