NYCU Introduction to Machine Learning, Homework 2

110550088 李杰穎

1 Part. 1, Coding

1.1 Logistic Regression

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

   - Learning rate: 0.01
   - Iteration: 1000

   Noted that I normalize the training data before training.

2. Show the weights and intercept of your model.

   Figure 1 shows the weights and intercept of the model.

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

   Figure 1 shows the accuracy of the model. The accuracy is 0.7541, which is greater than 0.75.

```
Part 1: Logistic Regression
Weights: [-0.0358968  -1.31643396  0.80233053 -0.1577323   0.03048323 -0.57369561], Intercept: -0.9019473887248837
Accuracy: 0.7540983606557377
```

Figure 1: The weights, intercept and accuracy of logistic regression

1.2 Fisher's Linear Discriminant (FLD)

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

   Figure 2 shows the mean vector m_0 and m_1.

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

   Figure 2 shows the within-class scatter matrix S_W of the training set.

6. Show the between-class scatter matrix S_B of the training set.

   Figure 2 shows the between-class scatter matrix S_B of the training set.

7. Show the Fisher's linear discriminant w of the training set.

   Figure 2 shows the Fisher's linear discriminant w.

8. 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.

   Figure 2 shows the accuracy on the testing set. The accuracy is 0.657, which is greater than 0.65.

9. Plot the projection line (x-axis: age, y-axis: thalach).

   Figure 3 shown the projection line.

```
Part 2: Fisher's Linear Discriminant
Class Mean 0: [ 56.75925926 137.7962963 ], Class Mean 1: [ 52.63432836 158.97761194]
With-in class scatter matrix:
[[ 19184.82283029 -16006.39331122]
 [-16006.39331122 106946.45135434]]
Between class scatter matrix:
[[ 17.01505494 -87.37146342]
 [-87.37146342 448.64813241]]
w:
[-0.28737344  0.95781862]
Accuracy of FLD: 0.6557377049180327
```

Figure 2: The mean vectors, within-class scatter matrix, between-class scatter matrix, Fisher's linear discriminant and accuracy of FLD method

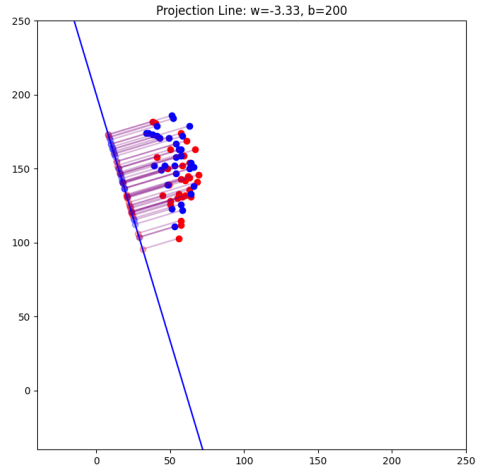

Figure 3: The projection line given by Fisher's Linear Discriminant. Blue point is positive samples and red is negative samples (x-axis: age, y-axis: thalach)

2 Part. 2, Questions

1. 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.

   First, let's discuss the difference between these two functions from few perspectives:

   - **Formula**

     The sigmoid function is given by $\sigma(x) = \frac{1}{1+e^{-x}}$. In the contrast, the softmax function, for an n dimensional vector x, is given by $\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}, i = 1,2,...,n$. Although, the two functions use exponential function, but the mathematical form is quite different.

   - **Input and output**

     The sigmoid function takes any input value x to a value between 0 and 1. This function is a non-decreasing and 1-to-1 function, make it useful for model where the output is binary.

     And the softmax function converts a vector x to a probability distribution, every value in the converted vector is between 0 and 1, and the sum of all elements is 1.

   As for the scenarios the two function be used. The sigmoid function is often used in the final layer of a **binary classification** model to map the single output to probability. For the softmax function, it often used in the final layer of a **multi-class classification** model to map the output value into a probability for every class.

   In summary, the sigmoid function is primarily used for binary classification or scenarios requiring probability estimation for a single event, as for the softmax function, it used in multi-class classification problems where a probabilistic distribution over multiple classes is required.

2. 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.

   First, logistic regression models the probability of a binary outcome. The output of logistic regression, which constraints between 0 and 1, represents the likelihood of a data point belonging to a particular class. However, the MSE loss is designed to measure the average square distance between estimated and actual value, this makes it more suitable for continuous value predictions.

   To be more concise, when we apply MSE loss to logistic regression, MSE doesn't reflect the model's performance accurately. The squaring error can lead to small deviation in probabilities are penalized too lightly, and larger deviation too seriously. This make MSE loss often has slower convergence or convergence to local minima when applying to logistic regression.

   In the contrast, the cross-entropy function is designed to measure the difference between two probability distributions (e.g. two models whose output is between 0 and 1). Therefore, cross-entropy is a better fit for logistic regression.

   Finally, cross-entropy provides a more direct measurement of the model's performance in classification task, compared to MSE. It changes more significantly with

small changes in the predicted probabilities, leading to more effective and efficient learning process.

3. 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?

   Before answering the three questions below, I assume that P1, P2, ..., Pc are the results after apply softmax function to the original output. This assumption guarantees that the value of P_i will between 0 and 1, and $\sum_{i=1}^{c} P_i = 1$.

   (a) What are the metrics that are commonly used to evaluate the performance of the classifier? Please at least list three of them.

   - Accuracy
     This method measure the percentage of correctly predicted instance out of all predictions. It's straightforward but may not be informative in case of class imbalance. We can also calculate the in-class accuracy.
   - Precision, recall and F1-score
     Precision is the ratio of correctly predicted observations to the total predicted observations in that class.
     Recall is the ratio of correctly predicted observations to the total true observations in that class.
     And F1-score is a metric that combined precision and recall, one model need to achieve high precision and recall at the same time to achieve high F1-score. The formula of F1-score is $2 \times \text{Precision} \times \text{Recall} / (\text{Precision} + \text{Recall})$.
   - Confusion matrix
     Confusion matrix is a matrix that provides a detail view for classifier's performance across all the classes, each entry correspond a predicted class and a true class. The number of that entry C_{ij} show that number of samples that being predicted as class j and its true class is class i. By observing the confusion matrix, we can know that the model performs worse when classifying which class.

   (b) Based on the previous question, how do you determine the predicted class of each sample?

   Predicted class is the corresponding i whose P_i is the highest among all P.

   (c) 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?

   For example, for a imbalance dataset, the classifier can always output class-1 to achieve 90% of accuracy. This make it unfair to evaluate the model by accuracy.

   Therefore, we can need to calculate the in-class accuracy or other metrics, and observe the metrics of every class. And for overall model performance, we can use macro-average, micro-average and weighted-average. Marco-average is a straightforward method, it just takes average of the metrics. As for micro-average, it considers the number of samples in every class, i.e. $\text{Precision}_{\text{micro}} = \frac{\sum \text{TP}_i}{\sum (\text{TP}_i + \text{FP}_i)}$, $\text{Recall}_{\text{micro}} = \frac{\sum \text{TP}_i}{\sum (\text{TP}_i + \text{FN}_i)}$. It's easy to check that the total number of false positve sample and the total number of false negative sample is actually the same, they both eqaul to the total number of wrong predictions. Therefore, for micro-average, accuracy, precision, recall and F1-score are all the same. Lastly, for weighted-average, it weights each metrics using the percentages of that class.

   By observing different metrics and confusion matrix, we can evaluate the model in a more fair manner.

4. 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.)

   Before calculating the below results, let's first prove that $\sigma'(x) = \sigma(x)\,(1 - \sigma(x))$

$$
\begin{aligned}
\sigma'(x) &= \frac{\mathrm{d}\sigma(x)}{\mathrm{d}x} \\
&= \frac{\mathrm{d}(\frac{1}{1+e^{-x}})}{\mathrm{d}x} \\
&= e^{-x}(1+e^{-x})^{-2} \\
&= \frac{e^{-x}}{1+e^{-x}} \; \frac{1}{1+e^{-x}} \\
&= (1-\frac{1}{1+e^{-x}}) \; \frac{1}{1+e^{-x}} \\
&= (1-\sigma(x)) \; \sigma(x)
\end{aligned}
$$

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

$$
\begin{aligned}
&= -t \; \sigma'(x) \; \frac{1}{\sigma(x)} + (1-t) \; \sigma'(x) \; \frac{1}{1-\sigma(x)} \\
&= -t\sigma(x)(1-\sigma(x)) \; \frac{1}{\sigma(x)} + (1-t)\sigma(x)(1-\sigma(x)) \; \frac{1}{1-\sigma(x)} \\
&= -t(1-\sigma(x)) + (1-t)\sigma(x) \\
&= \sigma(x) - t
\end{aligned}
$$

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

$$
\begin{aligned}
&= -2 \; \sigma'(x) \; (t-\sigma(x)) \\
&= -2 \; \sigma(x)(1-\sigma(x))(t-\sigma(x))
\end{aligned}
$$