CSE353 Assignment 5 Report

Chengzhi Dong
ID 112890166

Due November 24 2021

1 Introduction

For Assignment 5, I need to use the One-Versus-All (OVA) strategy and the One-Versus-One (OVO) strategy to implement the multi-class classification. I will use logistic regression algorithm to softly classify the data, so I can use the probability to classify the data in the overlapping region of different classes. For both One-Versus-All and One-Versus-One strategies, I need to plot the data, highlight the classified samples, and report the error rates. For the OVA strategy, I also need to draw the decision boundaries of the four OVA classifications.

2 Method

My algorithm implemented the logistic regression algorithm, and used the One-Versus-All (OVA) strategy and the One-Versus-One (OVO) strategy to implement the multi-class classification.

First, I imported the numpy and matplotlib.pyplot libraries.

Then, I leveraged my previous codes on logistic regression. I defined a sigmoid function and the logistic regression algorithm function. The logistic regression algorithm function takes a initial w_0, a learning rate (step) α, a set of data x_{data}, the corresponding ground truth y_{data}, and the max iteration. The logistic regression algorithm uses gradient descent to calculate the optimal $w_{LogisticReg}$. The function returns the $w_{LogisticReg}$ and the number of iteration it takes as a dictionary.

I also defined a function to count error rate. The count_error_rate function takes an array of result test_classification from the OVA or OVO multiclass classification and an array of ground truth (y_{data}). The function iterate through the two array and compare the classification of each data. If a test classification does not match the ground truth, then the error number would increase by 1. Last, the function return the error rate, which is obtained by the error number divided by the data size.

For the One-Versus-All strategy, I defined a function called "OVA_classify" and a OVA_visualization function. The OVA_classify function takes a dictionary of decision boundaries (w_{dict}, in this case, it would be the decision boundaries of the four OVA classifications), a data x (only one data point $[x_0, x_1, x_2]$), and the class set ($\{1, 2, 3, 4\}$). For each decision boundaries in w_{dict}, the function calculated the $sig(w_c^T x)$ to find the probability that this data x belong to class c. The function returns the most likely class that the data x belong to by OVA strategy. The OVA_visualization function plot the plot the data, draw the decision boundaries of the four OVA classifications, highlight the classified samples, colored background region for each classification, and report the error rates for visualization.

For the One-Versus-One strategy, I defined a function called "OVO_classify" and a OVO_visualization function. The OVO_classify function takes a matrix of decision boundaries (w_{matrix}, in this case, it would contain the six pairwise classifier decision boundaries $w_{k,l}$), a data x (only one data point $[x_0, x_1, x_2]$), and the class set ($\{1, 2, 3, 4\}$). The function would sums up all the probability of the data x belong to the class k and class l by $sig(w_{k,l}^T x)$. Then, it find the most likely class that the data x belong to by compare each class sum probability for the data x. The function returns the most likely class that the data x belong to by OVO strategy. The OVO_visualization function plot the plot the data, highlight the classified samples, colored background region for each classification, and report the error rates for visualization.

After I defined previous functions, I loaded the data into arrays of x_{data} and $ydata$. Also, I created a set of classes from the ground truth y_{data}. Then, I started use the One-Versus-All (OVA) strategy and the One-Versus-One (OVO) strategy to implement the multi-class classification.

For part 1, I use the One-Versus-All (OVA) strategy to implement the multi-class classification. First, I reformat the data to $D_{[k]} = \{(x_n, y_n' = \{+1 \quad if \quad y_n = k; \quad -1 \quad if \quad y_n \neq k\})\}_{n=1}^{N}$ for each $k \in class\_set$. Then I use the logistic regression algorithm function with $w_0 = [0, 0, 0]$, $\alpha = 5$, and $max\_iteration = 1000$ to calculate the $w_{LogisticReg}$ on each $D_{[k]}$ and store each $w_{LogisticReg}$ into a dictionary w_{ova} as $\{k : w_k\}$. Once, I obtained the decision boundaries dictionary w_{ova}, I loop over the data set x_{data}, I called the OVA_classify function with w_{ova} and each data point x to calculate an array ova_classification for the data. Then, I calculate the error rate of the ova_classification and called the OVA_visualization to plot the graph.

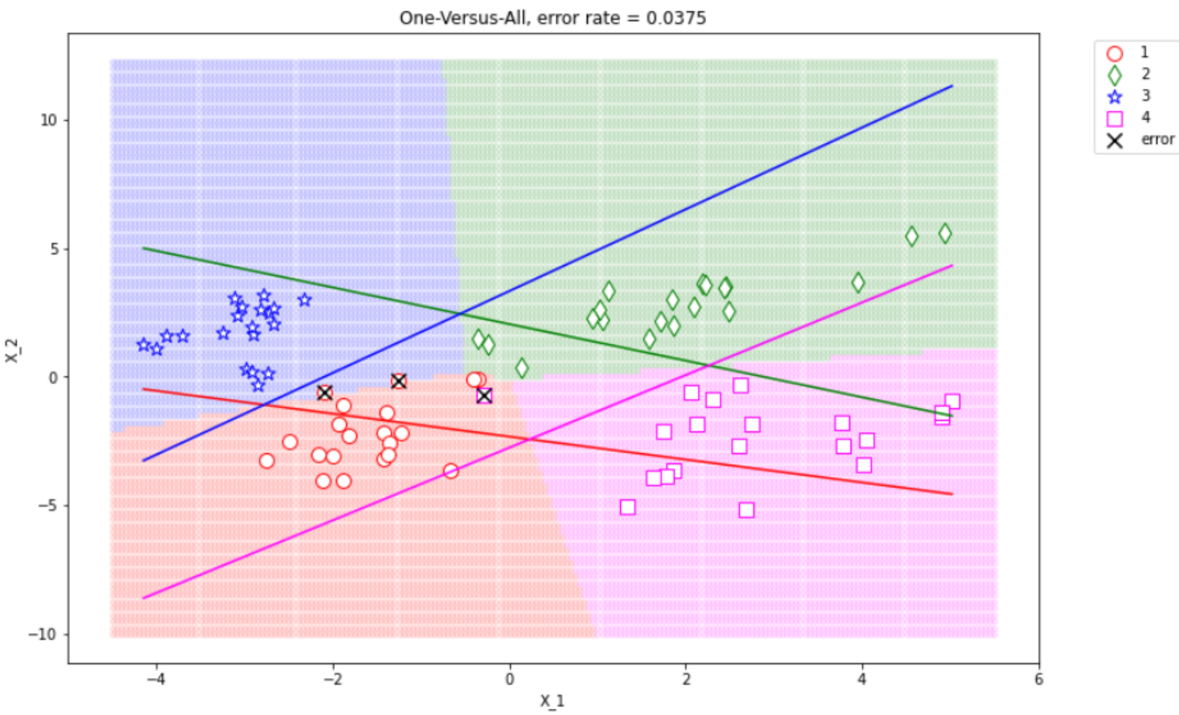
For part 2, I use the One-Versus-One (OVO) strategy to implement the multi-class classification. First, I reformat the data to $D_{[k,l]} = \{(x_n' = x_n \text{ if } y_n = k \text{ or } y_n = l,\ y_n' = \{+1 \quad if \quad y_n = k; \quad -1 \quad if \quad y_n = l\})\}_{n=1}^{N}$ for each $k, l \in class\_set * class\_set$. Then I use the logistic regression algorithm function with $w_0 = [0, 0, 0]$, $\alpha = 5$, and $max\_iteration = 1000$ to calculate the $w_{LogisticReg}$ on each $D_{[k,l]}$ and store each $w_{LogisticReg}$ into a matrix w_{ovo} as $w[k+1][l+1] = w_{[k,l]}$. The algorithm ignore if $k \leq l$ because $w_{[k,l]}$ has been calculated and there is no need to calculate $w_{[l,k]}$ again. If $k = l$, we are classify within only one class so also ignore that. Once, I obtained the decision boundaries matrix w_{ovo} which in this case contain six $w_{[k,l]}$, I loop over the data set x_{data}, I called the OVO_classify function with w_{ovo} and each data point x to calculate an array ovo_classification for the data. Then, I calculate the error rate of the ovo_classification and called the OVO_visualization to plot the graph.

3 Experiment

Part 1: One-Versus-All (OVA) strategy

Below are my results for Part 1:

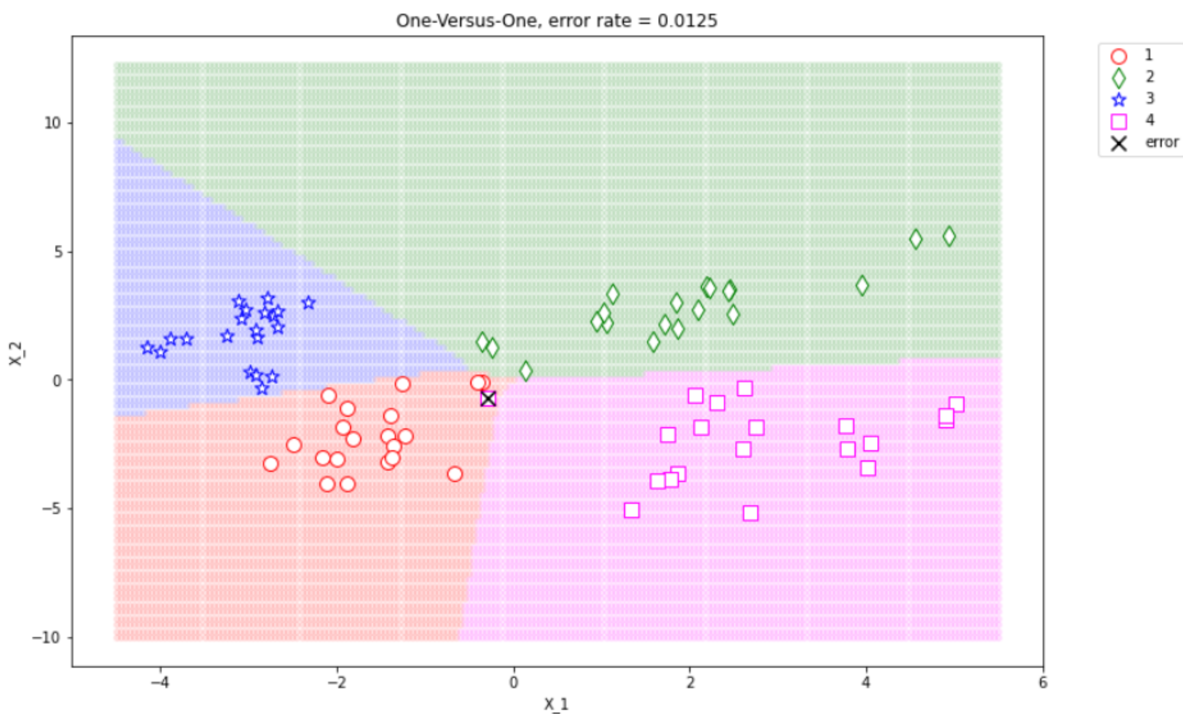
```
Part1: One-Versus-All (OVA)
Class 1 vs. All w: [-4.92619274 -0.94330895 -2.11885676]
Class 2 vs. All w: [-4.53849068  1.57431481  2.21450014]
Class 3 vs. All w: [-7.02013749 -3.35275379  2.11018085]
Class 4 vs. All w: [-4.88952082  2.5015903  -1.77331577]
```



As the results shown, the classification of One-Versus-All multi-class classification has an error rate of 0.0375. Marked as X, it misclassified two points of class 1 (red circle) as class 3 (blue region), and one point of class 4 (magenta square) as class 1 (red region). One of the factors that causes these errors maybe the w_0 and α I chose when using the logistic regression algorithm. Overall, the One-Versus-All multi-class classification algorithm did a decent job on classifying the data with only 3 mistakes out of 80 samples.

Part 2: One-Versus-One (OVO) strategy
Below are my results for Part 2:

```
Part2: One-Versus-One (OVO)
Class 1 vs. Class 2 w: [ 0.20151258 -4.49715707 -6.35558736]
Class 1 vs. Class 3 w: [ 4.84421509  3.03537893 -6.28064987]
Class 1 vs. Class 4 w: [-0.77381141 -5.55931964  0.26627215]
Class 2 vs. Class 3 w: [1.29697055 6.02341705 2.74427773]
Class 2 vs. Class 4 w: [ 0.         -1.10105625  6.44841875]
Class 3 vs. Class 4 w: [-0.41163935 -7.20274974  5.42487085]
```

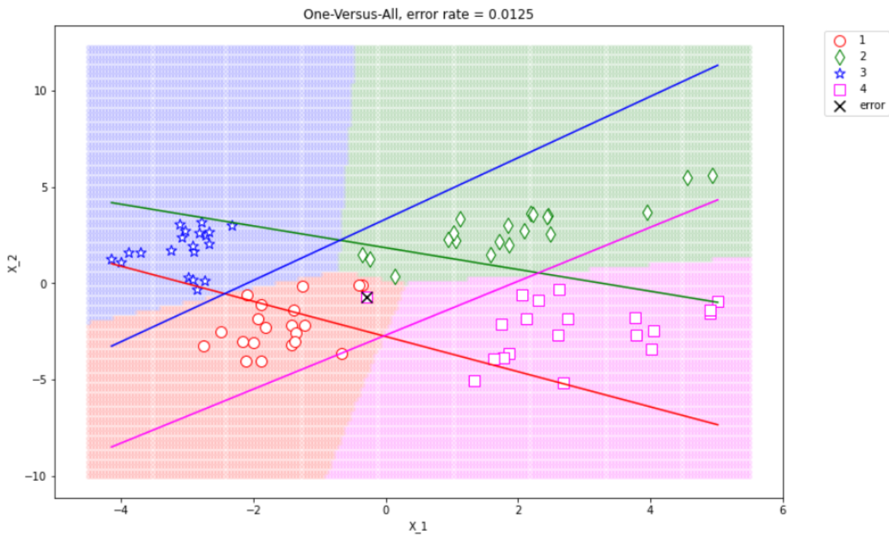


As the results shown, the classification of One-Versus-One multi-class classification has an error rate of 0.0125. Marked as X, it misclassified one point of class 4 (magenta square) as class 1 (red region). This result shows that the One-Versus-One strategy is more accurate than the One-Versus-All strategy because it has a lower error rate. With only 1 mistake out of 80 samples, the One-Versus-One multi-class classification algorithm did a great job.

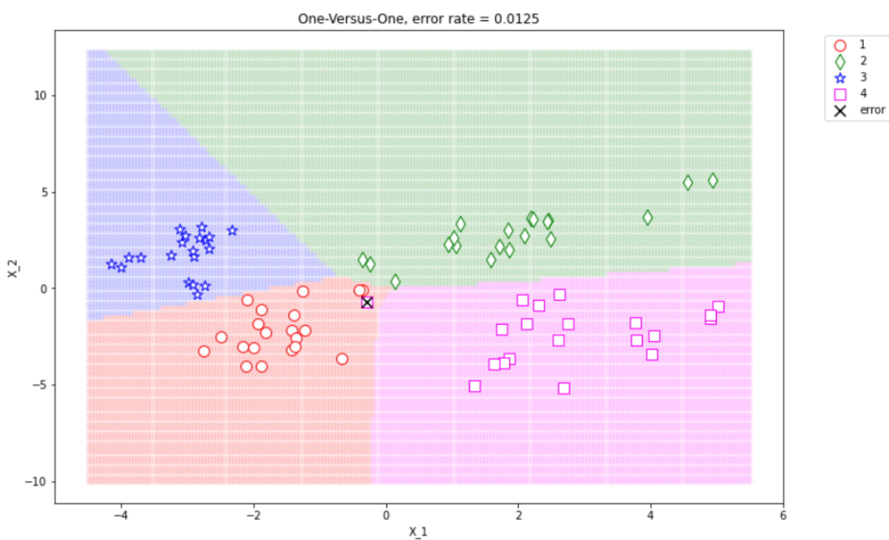
Other Test: try with different learning rate of Logistic Regression

Some differences occur after I change the learning rate α from 5 to 1:

```
Part1: One-Versus-All (OVA)
Class 1 vs. All w: [-3.07663658 -1.01758681 -1.11662615]
Class 2 vs. All w: [-3.70025868  1.13112598  2.00588706]
Class 3 vs. All w: [-7.01775004 -3.35164974  2.10947276]
Class 4 vs. All w: [-4.01019707  2.08450855 -1.49098961]
```




```
Part2: One-Versus-One (OVO)
Class 1 vs. Class 2 w: [ 0.59429137 -3.58271732 -4.79465616]
Class 1 vs. Class 3 w: [ 5.25188837  3.0621785  -5.18183896]
Class 1 vs. Class 4 w: [-0.51387211 -4.35244284  0.05441562]
Class 2 vs. Class 3 w: [2.44310084 4.23180657 1.26207268]
Class 2 vs. Class 4 w: [ 0.41503504 -1.16359679  4.5670898 ]
Class 3 vs. Class 4 w: [-1.18364131 -2.13083284  2.16048253]
```



The error rate of the One-Versus-All strategy decreased from 0.0375 to 0.0125, it made two less mistakes when $\alpha = 1$ compared to $\alpha = 5$. The error rate of the One-Versus-One strategy is still 0.0125, but we can observe that the boundary of classified magenta region (class 4) and red region (class 1) has changed under the misclassified sample.

4 Discussion

I choose to use logistic regression algorithm instead of PLA and linear regression because logistic regression gives a soft classification with the probability of each class the data x is belong to. For the overlapping class regions, I can decide which class the data x belong to by compared the probabilities and choose the highest one using the soft classification. On the other hand, PLA and linear regression gives hard classification, which is either 0 or 1, so for the overlapping class regions where all the values are 1, I cannot decide which class the data x belong to.

When choose the learning rate α of the logistic regression, I tried different learning rates and decided $\alpha = 5$ would be the most suitable one because when $\alpha = 5$ both OVA and OVO strategy did a great job on classify the multi-class data, and there is an obvious difference between the results of OVA and OVO in term of error rate.

I wanted to color the background region for each class of the OVA and OVO classification because it would gives a better visualization of the classification compared to just draw the decision boundary lines. I don't know how to color the different background region the official way. Therefore, what I did was I plot many of background points that covers all the region of the graph. For each background points, I use the OVA or OVO strategy to classified which class they belong to, then plot them with a bit transparent version of the colors.