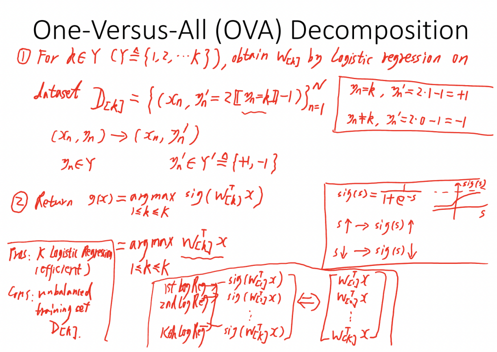
CSE353 - HW5
Yajie Wang

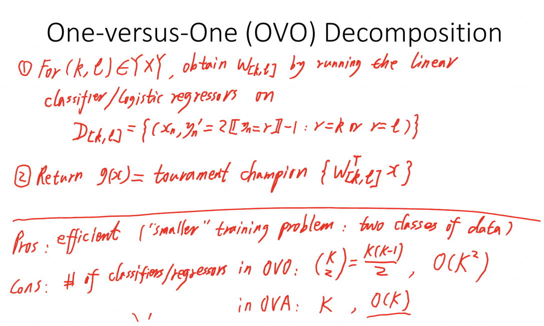
Introduction

In this assignment, we implement the multi-class classifications by using the One-Versus-All (OVA) strategy and the One-Versus-One (OVO) strategy. We plot the data and decision boundaries of the four one-versus-all binary classification, and highlight those samples that are misclassified to visualize the results. Finally, we check the reliability of our algorithms by calculating the training error rates.

Method

The method I used in this assignment is based on what I learned in class. I implement the OVO-based and OVA-based multi-class classifications by following the instructions shown in the lecture slides. In OVA, every class is paired with the remaining classes and the winner takes all. In OVO, every class is paired with other classes one-by-one and each classification is given one vote for the winning class. The class with the highest votes is the winner. I leverage my previous codes on logistic regression because it is easy to implement and very efficient to train. I don't need to use stochastic gradient descent since the dataset is small. I define some get functions to calculate and return the values needed. I mainly use for loops and if-else statements to implement my algorithm.

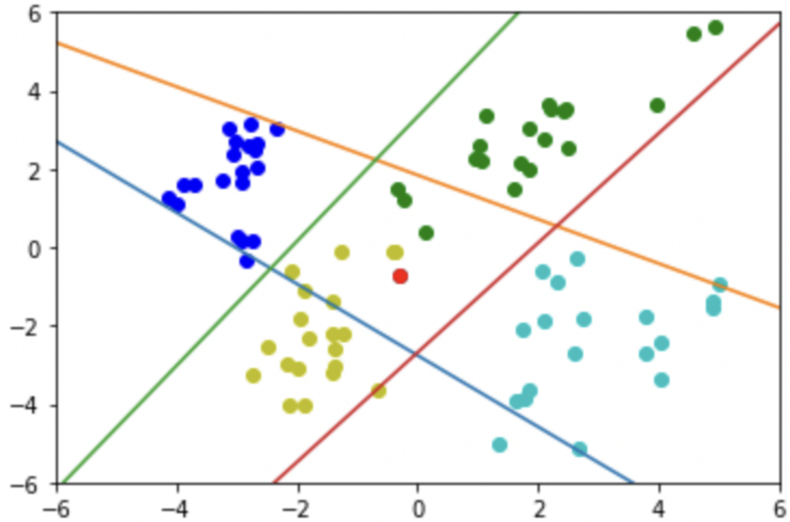




Experiments

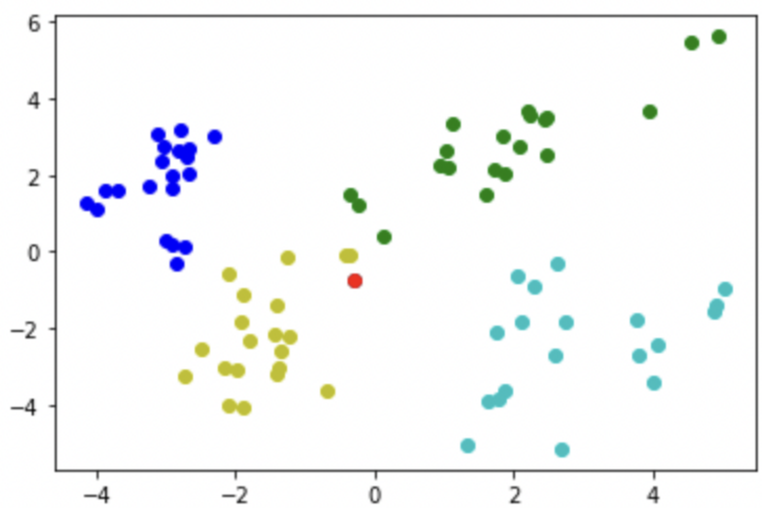
Part 1:

```
Error rate:  0.0125
```



Part 2:

```
Error rate:  0.0125
```



(The red points are misclassified samples.)

Discussions

The result shows that the OVO-based and OVA-based multi-class classifications both work as expected. The error rates are very low because the training data is small and balanced. OVA-based classification may not be a good choice if the dataset is imbalanced. If the training dataset is big, then the OVO-based classification may be a better choice because it is more efficient. The number of data to train for each classifier model is smaller. The number of classifier models depends on the strategy we use. We need N binary classifier models for N-class instances if we use the OVA-based classification. In OVO-based strategy, we need N*(N-1)/2 binary classifier models. Therefore, the number of classifier models in OVO-based strategy may be large if there are many class instances.