CS498AML HW2

Songwei Feng(songwei3)
Yuxi Gu(yuxigu2)
Xuanyi Zhu(xzhu42)

Problem 1

We used **Python** as the programming language for this problem.

We wrote the SVM classifier from scratch. First, we read data from "adult.data.txt" and "adult.test.txt" and combined the data from these two txt files to become our raw data. Then we split our data into train data, test data and validation data randomly, with percentages of 80%, 10%, 10% respectively. After that, we selected the variables that are continuous and used these variables to compose our feature vector. And we dropped examples where there were missing values of the continuous variables and scaled the variables inside the feature vector by using the StandardScaler in Python so that each has unit variance. Finally, we tried each regularization constant (either 0.001, 0.01, 0.1 or 1) with 50 epochs, and each epoch has 300 steps. And in each epoch, we chose 50 training examples at random for evaluation.

1. Plot of the accuracy every 30 steps, for each value of the regularization constant(lambda). For the x-axis, one batch is 30 steps. One thing to notice here is that the range of our y-axis is from 0.68-0.82, therefore, our plot may have more fluctuations than plots whose y-axis's starting point is from 0.

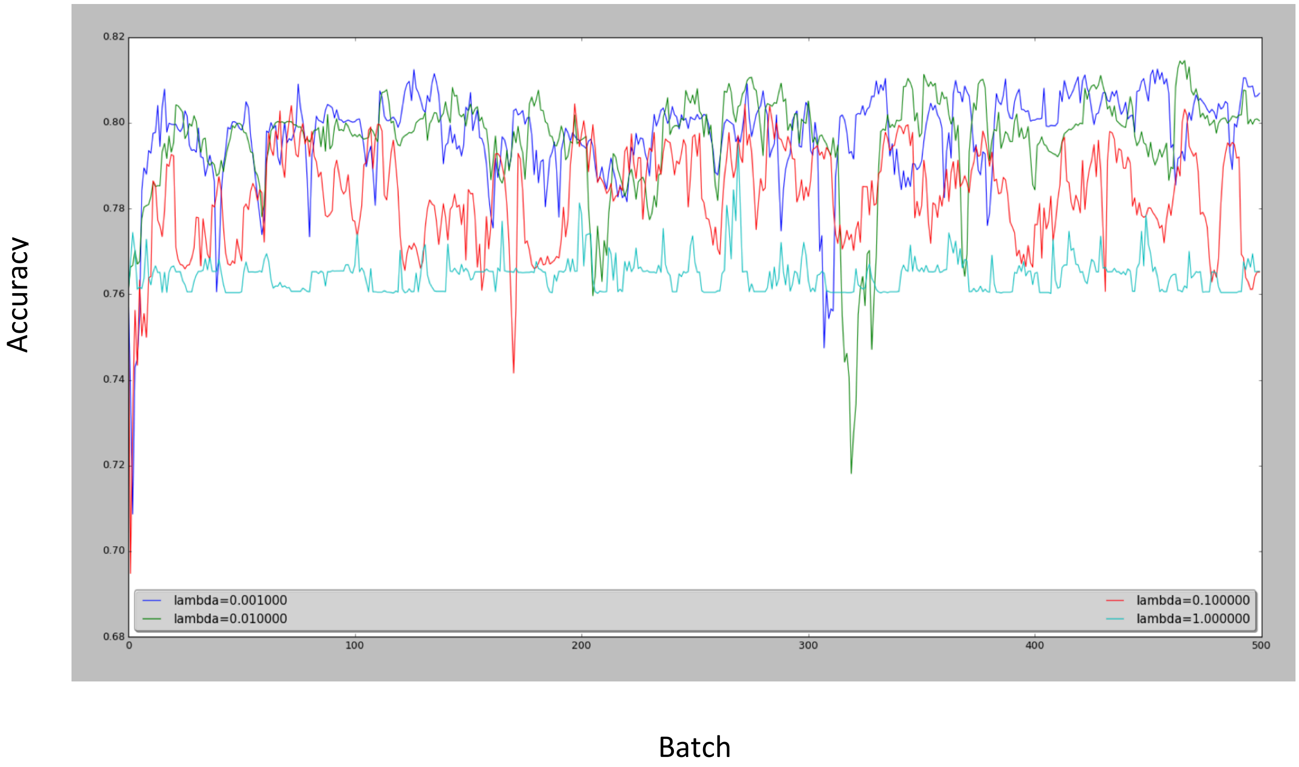

Figure. Plot of The Accuracy

2. Plot of the magnitude of the coefficient vector every 30 steps, for each value of the regularization constant.

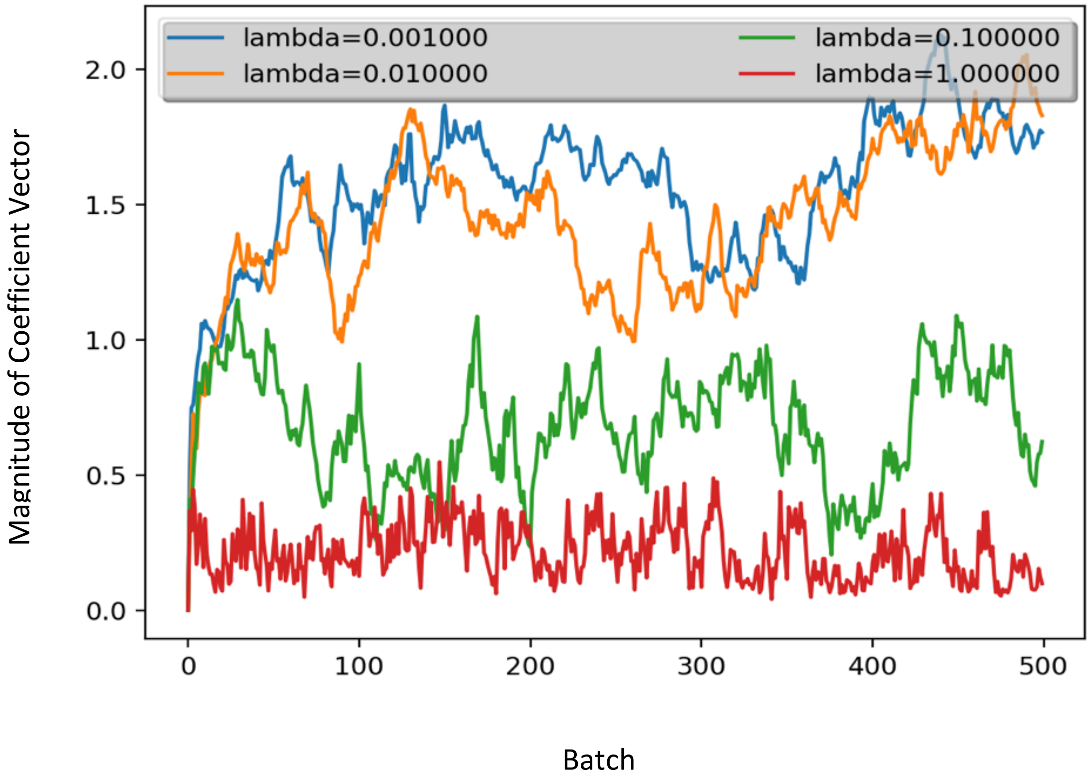

Figure. Plot of The Magnitude of Coefficient Vector

From the plot, we can see that the overall trend is that the magnitude of coefficient vector(a) will be smaller if the regularization constant(lambda) becomes larger.

3. Estimate of the best value of the regularization constant: **0.01**
From the Figure. Plot of The Accuracy, we can see that when the regularization constant get smaller, the accuracy tends to converge slower in the beginning, but it will eventually end up with a higher accuracy. In other words, the SVM classifier with the smaller regularization constant will have a higher accuracy in the overall trend.

Also, as the regularization constant get smaller, the difference between the magnitude of coefficient vector becomes much smaller, since it's obvious that the difference between the curve for lambda= 0.001 and the curve for lambda=0.01 after batch= 400 is very small in the Figure. Plot of The Magnitude of Coefficient Vector. And the magnitude of coefficient vector for small lambda can exceeds the magnitude of coefficient vector of smaller lambda at some point due to the fluctuations, such as the case that the magnitude for lambda=0.01 exceed the magnitude for lambda=0.001 at batch= 500. Based on the above observation, we believe 0.01 is a good regularization constant.

4. The accuracy of the best classifier on the 10% test dataset: **81.1016%**