Kerem Anar

ENGR 421: Intro. To Machine Learning

HW\_06 Report

Part 2 and 3)

I took the given data and divide it into training (first 1000) and test (remaining 4000) part.

Part 4)

I defined the Gaussian Kernel function as we did in LAB 8.

Text

Description automatically generated

Figure : Gaussian Kernel Function.

s, the width parameter of Gaussian Kernel, is given 10 in this project. I set s = 10 and calculated Kernel functions of training data and test data with

*K\_train = gaussian\_kernel(data\_train, data\_train, s)*

*K\_test = gaussian\_kernel(data\_test, data\_train, s)*

For the remaining part, I defined OneVersusAll(C) function to calculate predictions for training and test data points for given C variable with one-vs-all approach.

For this purpose, initially I rearranged the label values. For each class, I constructed new label array which gives 1 if it is in that class, otherwise returns -1.

If we write our dual problem in matrix-vector form:

Text, letter

Description automatically generated

In Lab 8, we learned how to solve this problem with cvxopt library in phyton. I used same method to obtain alpha values and w0.

Text

Description automatically generated

Figure : Solving QP problem by using cvxopt library.

Then I predicted training labels:

*f\_predicted\_train.append( np.matmul(K\_train, label\_train\_new[:,None] \* alpha[:,None]) + w0 )*

by using K\_train, label\_train\_new and alpha for each class (one) versus to all.

For test labels, I used the following:

*f\_predicted\_test.append( np.matmul(K\_test, label\_train\_new[:,None] \* alpha[:,None]) + w0 )*

Especially at the beginning I used label\_test\_new to predict test labels but it leaded to dimensionality issue and also I realized that it doesn’t make sense that using labels to predict that labels. Then I corrected my mistake by using labels\_train\_new in prediction line. Now it makes sense because basically Gaussian Kernel gives me the relationship between training data points and test data points. I know the relationship between training data points and training labels through the alpha values. Now, by using Kernel of training and test data points, training labels and alpha values, I should be able to figure out the relationship between test data points and their labels. Finally, this OneversusAll(C) function returns the training predictions and test predictions.

Part 5)

In this part I simply compare the corresponding label values for each data point and pick the maximum one as its class as we did in previous homework and labs.

The confusion matrix for training data points can be seen in Figure 3.

Text

Description automatically generated with medium confidence

Figure : Confusion Matrix for training data.

Part 6)

I did same things with part 5 to construct confusion matrix for test data points which can be seen in Figure 4.

Text

Description automatically generated

Figure : Confusion Matrix for test data.

Part 7)

For classification accuracy, I basically looked at how many samples out of total data I predicted correctly. I calculated predictions of training and test data for different C values in a for loop. Then for each C value, for both training and test data, I counted correctly-predicted samples and divided it to total data in training and test cases and multiplied by 100. This gives me percentage accuracy for different C values of the algorithm for both training and test data.

Chart, line chart

Description automatically generatedAs we can see in Figure 5, larger C values provides better prediction performance in training data set. Because when we increase the regularization parameter C, we penalize the slack variables which can be seen in Figure 6, primal problem.

Figure : Training and Test Accuracy of Gaussian Kernel Machine OVA algorithm.

Text, letter

Description automatically generated

Figure : Primal problem, penalty term.

Since we minimize the primal problem, for larger C, we force the slack variables be much smaller indirectly and misclassified data is decreased in training data set. However, much larger C values lead to construct a specific alpha values for this training set. It actually leads to overfitting. Therefore, much larger C values decrease the test data performance of the algorithm as we can see in Figure 5.