CENG414-Introduction to Data Mining

Programming HW1

Task 4-Support Vector Machine

Zeynepsu Kesim

2292340

Run the classifier with default parameters and report Summary and Detailed Accuracy By Class

I use ‘splitting data with 66% training set’ which is the default value.

```
=== Summary ===

Correctly Classified Instances        3181               93.5588 %
Kappa statistic                          0.9284
Mean absolute error                      0.1605
Root mean squared error                  0.2728
Relative absolute error                 89.1863 %
Root relative squared error             90.9254 %
Total Number of Instances             3400

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0,987    0,005    0,957      0,987    0,972      0,969    0,995     0,955     0
                 0,979    0,005    0,964      0,979    0,971      0,968    0,997     0,963     1
                 0,918    0,009    0,926      0,918    0,922      0,913    0,983     0,906     2
                 0,936    0,011    0,907      0,936    0,921      0,912    0,984     0,879     3
                 0,931    0,007    0,937      0,931    0,934      0,927    0,992     0,908     4
                 0,890    0,011    0,895      0,890    0,893      0,881    0,977     0,846     5
                 0,934    0,005    0,959      0,934    0,946      0,940    0,992     0,935     6
                 0,947    0,005    0,958      0,947    0,953      0,947    0,992     0,932     7
                 0,898    0,005    0,948      0,898    0,922      0,915    0,982     0,892     8
                 0,929    0,011    0,906      0,929    0,917      0,908    0,982     0,865     9
Weighted Avg.    0,936    0,007    0,936      0,936    0,936      0,928    0,988     0,909
```

Support vector machines have a parameter called C. With this parameter, we can control how much we want to trade off a large separation between classified examples and a perfect classification. Because we want actually two main things: a hyperplane with the largest minimum margin, and a hyperplane that correctly separates as many instances as possible. But we cannot take both of them well. Therefore, the question is whether we want to have a large margin between different classes or we want the perfect classification. As much as we increase

the C parameter's value, it means that we want more and more correct values of the points meaning a better classification, and a smaller margin. If we have a medium C, then for having a large separation between classes, we can give up for some points and allow them to be classified in a different group. And as much as we decrease the value of C, it means that we care less about the precise values of the points; on the contrary, we more care about the largest possible distance. Choosing C is significant since we are determining a trade off in our predictions. When C is too large, it can overfit the data which can cause problems in testing the model. For example, when C is too large, it can take the outliers in account, and thus, could have wrong predictions in the test data.

In this dataset, we have the default C value = 1 accuracy = 93.5588%

We can observe accuracy levels by trying different c values (0.0001,0.001,0.01,0.1,1,10,100,1000,10000). As C increases, first, test accuracy increases with the training accuracy; however, after c=1, we see that test accuracy starts to fall very slightly as training accuracy increases. This shows us the overfitting. As I mentioned above, as C value starts to get 'too large', now model tries to separate the data with smaller margins which may lead model to include noises or outliers in the training set, and thus, have slightly worse predictions for the test set. Therefore at c = 1 is relatively the best one.