

CMPE 462 - Project 2

Implementing an SVM Classifier

Student ID1: 2019700099

Student ID2: 2019705069

Student ID3: 2019700057

Task 1: Train and test accuracy of linear a hard margin linear SVM.

In this part, we applied linear hard margin SVM by choosing a large C value. The training and test accuracies are found as 74.6667% and 77.5% ,respectively. The test accuracy is found greater than the train accuracy. This observation contradicts the theory and may have occurred due to the distribution of the data.

Task 2: Performance for different C values for a fixed kernel.

In this part, we trained the training data with SVM using different kernels and C values. The tested kernels are linear, RBF, polynomial, and sigmoid. The results of linear kernel are illustrated in Table 1.

Value of C	Train Accuracy	Test Accuracy
1.024e-07	53.33333333333336	58.33333333333336
5.12e-07	53.33333333333336	58.33333333333336
2.56e-06	53.33333333333336	58.33333333333336
1.28e-05	53.33333333333336	58.33333333333336
6.4e-05	53.33333333333336	58.33333333333336
0.00032	53.33333333333336	58.33333333333336
0.0016	53.33333333333336	58.33333333333336
0.008	82.66666666666667	84.16666666666667
0.04	84.0	84.16666666666667
0.2	85.33333333333334	84.16666666666667
1	86.66666666666667	85.0
5	88.66666666666667	81.66666666666667
25	88.66666666666667	81.66666666666667
125	88.66666666666667	81.66666666666667
625	90.0	81.66666666666667
3125	90.0	81.66666666666667
15625	90.0	81.66666666666667
78125	89.33333333333333	82.5
390625	88.66666666666667	84.16666666666667
1953125	80.0	83.33333333333334

Table 1: Performance of the linear kernel for different C values

Table 2 shows the results of RBF kernel when gamma value is selected as 0.0032.

Value of C	Train Accuracy	Test Accuracy
0.00032	53.33333333333336	58.33333333333336
0.0016	53.33333333333336	58.33333333333336
0.008	53.33333333333336	58.33333333333336
0.04	53.33333333333336	58.33333333333336
0.2	53.33333333333336	58.33333333333336
1	53.33333333333336	58.33333333333336
5	74.66666666666667	79.16666666666666
25	84.0	84.16666666666667
125	84.66666666666667	83.33333333333334
625	86.0	85.0
3125	89.33333333333333	85.0
15625	90.0	82.5
78125	93.33333333333333	83.33333333333334
390625	95.33333333333334	76.66666666666667
1953125	97.33333333333334	78.33333333333333

Table 2: Performance of the RBF kernel for different C values

Table 3 shows the results of polynomial kernel when gamma, coef0 and degree values are fixed as 0.00032,0.00032 and 2 respectively.

Value of C	Train Accuracy	Test Accuracy
0.00032	53.33333333333336	58.33333333333336
0.0016	53.33333333333336	58.33333333333336
0.008	53.33333333333336	58.33333333333336
0.04	53.33333333333336	58.33333333333336
0.2	53.33333333333336	58.33333333333336
1	53.33333333333336	58.33333333333336
5	53.33333333333336	58.33333333333336
25	53.33333333333336	58.33333333333336
125	53.33333333333336	58.33333333333336
625	53.33333333333336	58.33333333333336
3125	53.33333333333336	58.33333333333336
15625	83.33333333333334	79.16666666666666
78125	87.33333333333333	79.16666666666666
390625	89.33333333333333	80.0
1953125	95.33333333333334	76.66666666666667

Table 3: Performance of the polynomial kernel for different C values

Table 4 shows the results of polynomial kernel when gamma and coef0 values are fixed as 0.00032 and 0.00032 respectively.

Value of C	Train Accuracy	Test Accuracy
0.00032	53.33333333333336	58.33333333333336
0.0016	53.33333333333336	58.33333333333336
0.008	53.33333333333336	58.33333333333336
0.04	53.33333333333336	58.33333333333336
0.2	53.33333333333336	58.33333333333336
1	53.33333333333336	58.33333333333336
5	74.66666666666667	79.16666666666666
25	84.0	84.16666666666667
125	84.66666666666667	83.33333333333334
625	86.0	85.0
3125	89.33333333333333	85.0
15625	90.0	82.5
78125	93.33333333333333	83.33333333333334
390625	95.33333333333334	76.66666666666667
1953125	97.33333333333334	78.33333333333333

Table 4: Performance of the sigmoid kernel for different C values

Task 3: Observation the number of support vectors changes as the value of C increases.

The number of support vectors are listed in Table 5, 6, 7 and 8 when SVM with linear, polynomial, RBF and sigmoid kernel is applied respectively. In general, the number of support vectors increases when C the value gets bigger. For small C values the margin is wider. Therefore, the number of support vectors becomes more since the data points in the margin are also considered as support vectors. This observation matches the theory.

Value of C	Nummer of support vectors
0.00032	140
0.0016	140
0.008	124
0.04	87
0.2	68
1	58
5	54
25	50
125	50
625	49
3125	49
15625	49
78125	53
390625	67
1953125	83

Table 5: Number of support vectors of the linear kernel for different C values

Value of C	Nummer of support vectors
0.00032	140
0.0016	140
0.008	140
0.04	140
0.2	140
1	140
5	140
25	140
125	140

625	132
3125	91
15625	71
78125	62
390625	56
1953125	54

Table 6: Number of support vectors of the polynomial kernel for different C values

Value of C	Nunmer of support vectors
0.00032	140
0.0016	140
0.008	140
0.04	140
0.2	140
1	140
5	140
25	140
125	140
625	140
3125	106
15625	77
78125	64
390625	55
1953125	51

Table 7: Number of support vectors of the RBF kernel for different C values

Value of C	Nunmer of support vectors
0.00032	140
0.0016	140
0.008	140
0.04	140
0.2	140
1	140
5	140
25	140

125	140
625	140
3125	140
15625	140
78125	140
390625	140
...	...
...	...
...	...
59604644775390625	140
298023223876953125	136
1490116119384765625	28
7450580596923828125	6
37252902984619140625	2
186264514923095703125	2

Table 8: Number of support vectors of the sigmoid kernel for different C values

Task 4: Investigation of the changes in the hyperplane when a support vector removes and a data point that is not a support vector.

According to the theory, the features of the decision boundary are expected to stay the same after retraining when a datapoint that is not a support vector is removed from the training set. However, when a support vector is removed from the training set, the decision boundary after retraining is expected to show different characteristics such as possibly consisting of more support vectors than before and having a different shape etc.

In our application, we removed a random support vector and a non support vector from the training set. After that, we computed weight vectors of each case by multiplying the support vector coefficient matrix and support vectors. Testing the fact that is given above did not occur all the time in the sense of decision boundary staying the same. We suspect that this is because training data is not linearly separable. Nonetheless, in our tests we reached this fact most of the time. An example that supports the theory can be seen in our Jupyter notebook under Task 4.

Bonus Task: Implementation of hard margin SVM with QP solver.

$$Q = \begin{bmatrix} 0 & 0_{13}^T \\ 0_{13} & I_{13} \end{bmatrix}$$

$$p = 0_{14}$$

$$A = \begin{bmatrix} \text{trainY}[0] & \text{trainY}[0] * \text{trainX}[0]^T \\ \vdots & \vdots \\ \vdots & \vdots \\ \text{trainY}[149] & \text{trainY}[149] * \text{trainX}[149]^T \end{bmatrix}$$

$$c = 1_{150}$$

$$\text{sol} = \text{solvers.qp}(Q, p, A, c) = \begin{bmatrix} 6.93e-02 \\ -2.00e-10 \\ -3.24e-09 \\ -2.18e-09 \\ -3.01e-10 \\ -1.20e-10 \\ -2.65e-10 \\ -1.68e-09 \\ 1.32e-09 \\ -3.36e-09 \\ -1.11e-09 \\ -1.88e-09 \\ -1.91e-09 \\ -4.65e-09 \end{bmatrix} = \begin{bmatrix} b^* \\ w^* \end{bmatrix}$$

$$\Rightarrow \quad b^* = 6.93e - 02, \quad w^* = \begin{bmatrix} -2.00e - 10 \\ -3.24e - 09 \\ -2.18e - 09 \\ -3.01e - 10 \\ -1.20e - 10 \\ -2.65e - 10 \\ -1.68e - 09 \\ 1.32e - 09 \\ -3.36e - 09 \\ -1.11e - 09 \\ -1.88e - 09 \\ -1.91e - 09 \\ -4.65e - 09 \end{bmatrix}$$