Assignment - 2

Anurag Sarva

cs24mtech14003

Question 1.

Solution...

Given the equation for the margin boundaries:

$$y_i(w \cdot x_i + b) \geq 1, \quad \forall i$$

And the optimality condition:

$$Minimize \frac{1}{2}\|w\|^2,$$

We have to analyze the modified equation:

$$y_i(w' \cdot x_i + b') \geq \gamma, \quad \forall i.$$

To analyze the change due to this modification, we will reparameterize the problem. For that i consider the transformation of the weight vector w and bias b. By scaling the weight and bias, we can write this as follows:

$$w = \frac{w'}{\gamma} \quad and \quad b = \frac{b'}{\gamma}.$$

Substituting this into the original margin boundary equation (i.e. the equation which is given in the question):

$$y_i(w \cdot x_i + b) \geq 1,$$

After substituting it will looks like -

$$y_i \left(\frac{w'}{\gamma} \cdot x_i + \frac{b'}{\gamma} \right) \geq 1.$$

After simplifies the above equation -

$$y_i \left(w' \cdot x_i + b' \right) \geq \gamma.$$

It will looks similar to the modified equation.
Thus, we observe that after scaling down to the orignal equation, we see that it can be transformed into the standard form by scaling down w and b.

Now, let's take a look to the optimality condition, which alos remains the same:

$$Minimize\frac{1}{2}\|w'\|^2.$$

The optimization problem remains unchanged except for the scale factor. Therefore, it is clear that the scaling affects only the magnitude but not the direction of the weight vector.

From all of the above statements, i comes into the conclude that the solutions for

$$y_i(w \cdot x_i + b) \geq 1 \quad and \quad y_i(w' \cdot x_i + b') \geq \gamma$$

are the same.
Therefore, the choice of constants does not affect the decision boundary, and thus the solution remains unchanged.

Question 2.

Solution...

Given:- this is given in the question:

$$\rho = \frac{1}{\|w\|}$$

it is the definition of half-margin of maximum-margin SVM.

Prove:- We have to show that

$$\frac{1}{\rho^2} = \sum_{i=1}^{N} \alpha_i$$

Proof:- where $\rho = \frac{1}{\|w\|}$ and α_i are the Lagrange multipliers from the dual formulation of SVM, we follow these steps:
We know this relation -

$$\rho = \frac{1}{\|w\|}$$

After cross multiplying -

$$\|w\| = \frac{1}{\rho}$$

Taking square on both side -

$$\|w\|^2 = \frac{1}{\rho^2} - - - - - - - - - - - - - - - - - -(1)$$

Optimality condition for the dual SVM is writen as -

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i$$

here, y_i are the labels (+1 or -1), and
x_i are the feature vectors
Taking square on both the side of above equation -

$$\|w\|^2 = \sum_{i=1}^{N} \alpha_i y_i^2 \|x_i\|^2$$

After simplifying little bit, we get this

$$\|w\|^2 = \sum_{i=1}^{N} \alpha_i (since y_i^2 = 1) - - - - - - - - - - - - - - - - - - - (2)$$

Now, comparing equation (1) and (2), then we will get the expected result.

$$\frac{1}{\rho^2} = \sum_{i=1}^{N} \alpha_i$$

Hence, proved the equation

$$\frac{1}{\rho^2} = \sum_{i=1}^{N} \alpha_i$$

Question 3.

Solution...

Validity Analysis of a Kernel function

(a.)

The equation for a kernel is **valid**.

Proof : Kernels are valid here if they are positive semi-definite (PSD). Here k_1 and k_2 are valid PSD matrix. So, there sum is also valid Gram matrix PSD.

(b.)

The equation for a kernel is **valid**.

Proof : Kernels are valid here if they are positive semi-definite (PSD). Here k_1 and k_2 are valid PSD matrix. So, there product is also valid PSD matrix. Basicaly they are belong to valid kernal.

(c.)

The equation for a kernel is **valid**.

Proof : As we know polynomial functions with the positive coefficient of a valid kernel function remain always valid, but it is till when the polynomial function applied entry-wise to the matrix. And here it will happens that's why it is valid.

(d.)

The equation for a kernel is **not always valid**.
So, we considered as invalid.

Counter example : When the exponential function $\exp(k_1(x, z))$ is positive, it will not necessary it will be preserving the PSD property always. There is a few counter example is available, after applying the exponintial function might not remain PSD in the matrix, which will strictly depends on the nature of $k_1(x, z)$.

(e.)

The equation for a kernel is **valid**.

Proof : It is the widely used kernel in the field of ML, and it is known by Gaussian RBF Kernel. And the RBF kernel will be derived from the distance of a vector x and z, and the produced Gram matrix by this is always PSD, that shows it is a valid kernel.

Question 4.

Explanation...

In the programming question number four, for all its subpart, i combined the code, and i properly commented, if you go through my code you can able to see clearly.
For each part i separately created a function for respective parts of question four.

Output...

Programming Question Q4. a)

- Accuracy for the test set: 0.9788
- Number of support vectors by using the soft margin SVM: 28

Programming Question Q4. b)

Training the model with 50 points:

- Accuracy with this test set is: 0.9811
- Number of support vectors in this case is: 2

Training the model with 100 points:

- Accuracy with this test set is: 0.9811
- Number of support vectors in this case is: 4

Training the model with 200 points:

- Accuracy with this test set is: 0.9811
- Number of support vectors in this case is: 8

Training the model with 800 points:

- Accuracy with this test set is: 0.9811
- Number of support vectors in this case is: 14

Programming Question Q4. c)

This is what I get by using hyper parameter C and polynomial degree.

- On polynomial degree: 2, C: 0.0001
  - Training error is: 0.34080717488789236
  - Testing error is: 0.3466981132075472
  - Number of support vectors for this case is: 1112
- On polynomial degree: 5, C: 0.0001
  - Training error is: 0.05188981422165284
  - Testing error is: 0.07547169811320753
  - Number of support vectors for this case is: 374
- On polynomial degree: 2, C: 0.001
  - Training error is: 0.024983984625240208
  - Testing error is: 0.035377358490566
  - Number of support vectors for this case is: 558
- On polynomial degree: 5, C: 0.001
  - Training error is: 0.021140294682895577

  - Testing error is: 0.030660377358490587
  - Number of support vectors for this case is: 158

- On polynomial degree: 2, C: 0.01
  - Training error is: 0.008327994875080007
  - Testing error is: 0.021226415094339646
  - Number of support vectors for this case is: 164

- On polynomial degree: 5, C: 0.01
  - Training error is: 0.008327994875080007
  - Testing error is: 0.021226415094339646
  - Number of support vectors for this case is: 68

- On polynomial degree: 2, C: 1
  - Training error is: 0.004484304932735439
  - Testing error is: 0.018867924528301883
  - Number of support vectors for this case is: 30

- On polynomial degree: 5, C: 1
  - Training error is: 0.004484304932735439
  - Testing error is: 0.01650943396226412
  - Number of support vectors for this case is: 26

Here, I am evaluating the given statements:

- Statement (i): False
- Statement (ii): True
- Statement (iii): False
- Statement (iv): True

Programming Question Q4. d)

C	Testing Error	Training Error
0.01	0.01651	0.00512
1.0	0.02123	0.00448
100.0	0.01887	0.00320
10000.0	0.01887	0.00256
1000000.0	0.02358	0.00256

Lowest testing error at C: 0.01 and the error value: 0.01651
Lowest training error at C: 10000.0 and the error value: 0.00256

Question 5.

Explanation...

In the programming question number five, for all its subpart, i combined the code, and i properly commented, if you go through my code you can able to see clearly.
Here i created a function which will take kernel name and its functionality, then it perform there respective task.

Output...

Q5. a)

Kernel: Linear Kernel

- Number of support vectors: 1084
- Training error: 0.0000
- Testing error: 0.0240

Q5. b)

Kernel: Polynomial

- Number of support vectors: 1332
- Training error: 0.0005
- Testing error: 0.0200

Kernel: RBF

- Number of support vectors: 6000
- Training error: 0.0000
- Testing error: 0.5000