Question 1:

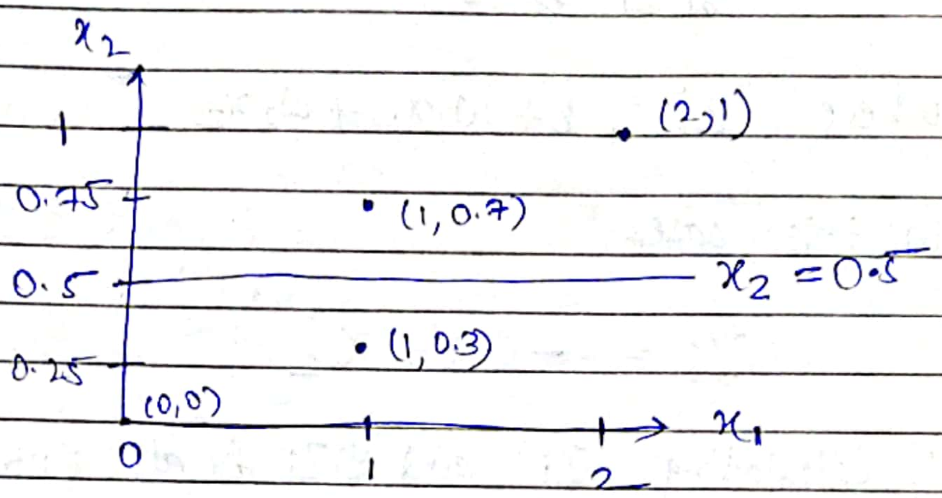
SVM HW6, ML

Sanil Chitnis
N10986573, SSC9983

Date :
Page No.

a)

x_{i1}	0	1	1	2
x_{i2}	0	0.3	0.7	1
y_i	-1	-1	1	1



$\therefore$ $x_2 = 0.5$ seperates the two classes.

$x_2 = 0.5$ can also be written as.

$$0x_1 + 1x_2 - 0.5 = 0.$$

$$\hat{y} = \begin{cases} 1, & 0x_1 + 1x_2 - 0.5 > 0 \\ -1, & 0x_1 + 1x_2 - 0.5 < 0 \end{cases}$$

$\therefore$ Comparing this with $b + w_1x_1 + w_2x_2$ gives us, $\boxed{b = -0.5,\ w_1 = 0,\ w_2 = 1}$

Date :
Page No.

b.) We know,

$$y_i\,(b + w^T x) \geq \gamma$$

$$y_i z_i \geq \gamma$$

where $z_i = b + w_1 x_1 + w_2 x_2$

In this case,

$$z_i = -0.5 + x_2$$

$\therefore$ Calculating z_i and $y_i z_i$ for all entries,

x_{i1}	0	1	1	2
x_{i2}	0	0.3	0.7	1
y_i	−1	−1	1	1
z_i	−0.5	−0.2	0.2	0.5
$y_i z_i$	0.5	0.2	0.2	0.5

since $y_i z_i \geq \gamma$

$\gamma \leq y_i z_i$

$\therefore$ The largest value of γ that we can have is 0.2.

Date :

Page No.

$$M = \frac{\gamma}{\|w\|}$$

$$\|w\| = \sqrt{w_1^2 + w_2^2}$$

Since we know from b, $\gamma = 0.2$

From a, $w_1 = 0$, $w_2 = 1$

$$\therefore \|w\| = \sqrt{w_1^2 + w_2^2} = \sqrt{0^2 + 1^2} = \sqrt{0+1} = \sqrt{1} = 1$$

$$\therefore M = \frac{0.2}{1} = 0.2$$

For points on margin, $y_i z_i = \gamma$ should satisfy.

As per table in b, Points $(1, 0.3)$ and $(1, 0.7)$ satisfy this constraint and thus are the points on margin.

Problem 2 :

x_i	0	1.3	2.1	2.8	4.2	5.7
y_i	-1	-1	-1	1	-1	1

$$\hat{y} = \begin{cases} 1 & z > 0 \\ -1 & z < 0 \end{cases}, \quad z = x - t$$

Date :
Page No.

a.) xi = np.array([0, 1.3, 2.1, 2.8, 4.2, 5.7])
yi = np.array([-1, -1, -1, 1, -1, 1])
t = np.linspace(0, 5, 100)
J = np.zeros(100)

for teach in t:

z = xi - teach
εi = np.maximum(0, 1 - yi z)
J.append(np.sum(εi))

plt.plot(t, J)
plt.xlabel('t')
plt.ylabel('J')
plt.grid()
plt.savefig("pltJt.png")

pltJt.png attached after this page.

Problem 2

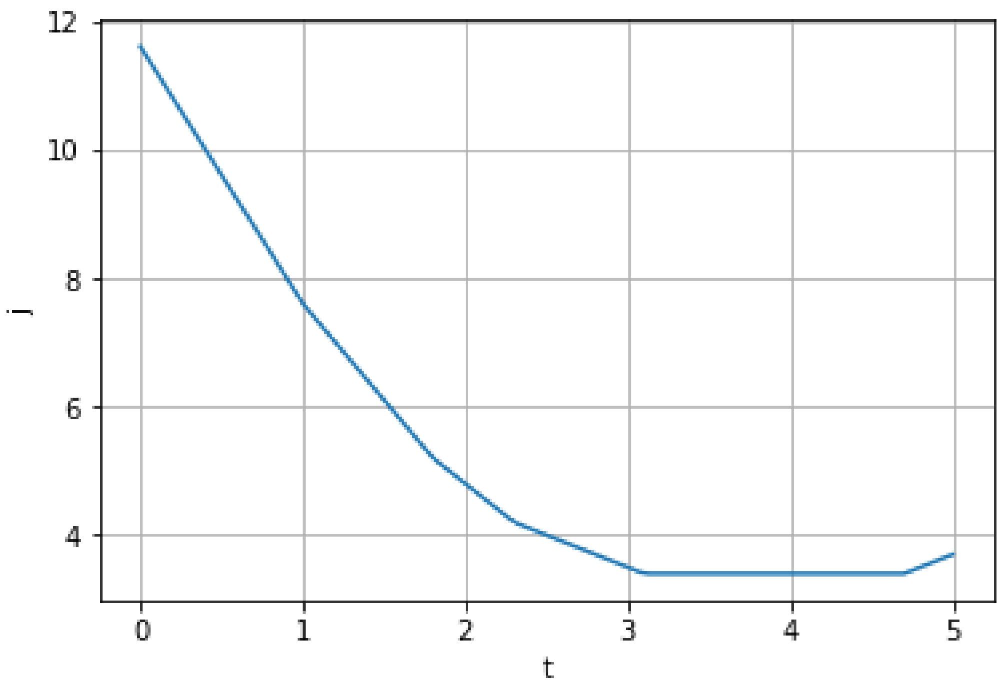
April 4, 2021

[33]:

```
import numpy as np
import matplotlib.pyplot as plt

xi = np.array([0,1.3,2.1,2.8,4.2,5.7])
yi = np.array([-1,-1,-1,1,-1,1])
t = np.linspace(0,5,100)
j = []

for tea in t:
    z = xi - tea
    ei = np.maximum(0, (1 - yi * z))
    j = np.append(j,np.sum(ei))
plt.plot(t,j)
plt.xlabel('t')
plt.ylabel('j')
plt.grid()
plt.savefig("pltJt.png")
```



b.) Min J can be seen from figure at $t = 4$.

c.) $t = 4$
$z = x_i - t$
ε_i = np.maximum(0, 1 - $y_i z$)

Please also find python output attached before this page (previous page.)

Also mentioning ε_i here,

$\varepsilon_i = [0, 0, 0, 2.2, 1.2, 0]$

d.) $\varepsilon_i > 1$ for $\varepsilon_3 = 2.2$ & $\varepsilon_4 = 1.2$ which are misclassified and since $\varepsilon_i > 0$ they also violate the margin.

Problem 3:

a.) $x = vec(X) = [0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0]$

$\omega = vec(W) = [0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0]$

b.) $z = \omega^T x$

Date :
Page No.

$$Z = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} [0,0,0,0,\ 0,0,0,0,\ 0,1,1,1,\ 0,0,0,0]$$

$$Z = 2$$

This shows that since x_i / w_i are only 1 and 0 and there are only 2 pixels where they overlap

c.) $x_{right} = vec(X >> 1)$

$$x_{right} = vec\left(\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}\right)$$

$$x_{right} = [0,0,0,0,\ 0,0,0,0,\ 0,0,0,0,\ 0,1,1,1]$$

Scanned with CamScanner

refore none of the pixels overlap.

$$x_{left} = vec\begin{bmatrix} & & \\ & X << 1 & \\ & & \end{bmatrix}$$

$$x_{left} = vec\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$x_{left} = [0,0,0,0,\ 0,1,1,1,\ 0,0,0,0,\ 0,0,0,0]$$

$$\therefore \quad z = \omega^T x_{left} = 2$$

erefore, 2 pixels overlap in this scenario.

x = np.ravel(xmat)
x_{mat} = np.reshape(x, (H, W), order = 'F')

re order = 'F' is required to put set of 4 elements from x has colums in x_{mat}, without his, it will be put as rows in x_{mat}, which we dont want (following method used in 'a').

Date :
Page No.

Problem 4:

x_i	0	1	2	3
y_i	1	-1	1	-1

$$\hat{y} = \begin{cases} 1, & z > 0 \\ -1, & z < 0 \end{cases}$$

$$z = \sum_i a_i y_i k(x_i, x)$$

$$k(x_i, x) = e^{-\gamma (x - x')^2}$$

a.) $\gamma = 3$, $\alpha = [0, 0, 1, 1]$

Solved PDF attached on next page

b.) $\gamma = 0.3$, $\alpha = [1, 1, 1, 1]$

Solved PDF attached. on next page.

c.) Since 2^{nd} classifier uses higher γ theoratically it is the classifier which makes least error.
Also,

		0	1	2	3
	x_i	0	1	2	3
	y_i	1	-1	1	-1
1 wrong	$\hat{y}_i = \text{sign}(z_i)$ for $\gamma = 3$	1	1	1	-1
0 wrong.	$\hat{y}_i = \text{sign}(z_i)$ for $\gamma = 0.3$	1	-1	1	-1

Problem 4

April 4, 2021

[72]:
```
import numpy as np
import matplotlib.pyplot as plt

x = np.array([0,1,2,3])
y = np.array([1,-1,1,-1])

def support_vector_classifier(x,y,gamma,alpha):
    yhat = np.zeros(100)
    x0 = np.linspace(-2,5,100)
    x0mat, xmat = np.meshgrid(x0,x)
    rbf = np.exp(-gamma * ((x0mat - xmat)**2))
    z = (y * alpha).dot(rbf)

    for i in range(100):
        if z[i] > 0:
            yhat[i] = 1
        else:
            yhat[i] = -1
    plt.plot(x0,z)
    plt.grid()
    plt.xlabel('x0')
    plt.ylabel('z/yhat')

    plt.plot(x0,yhat)
    plt.grid()
    plt.legend(['x vs Z', 'x vs yhat'])

plt.subplots_adjust(hspace=1.5)
plt.subplot(1,2,1)
plt.title('Solution 4a')
alpha = np.array([0,0,1,1])
gamma = 3
support_vector_classifier(x,y,gamma,alpha)

plt.subplot(1,2,2)
plt.title('Solution 4b')
alpha = np.array([1,1,1,1])
```

```
gamma = 0.3
support_vector_classifier(x,y,gamma,alpha)

plt.savefig('Problem4a4b.png')
```