Syed Ali   Cosc 750   Assignment due 4/19/2020

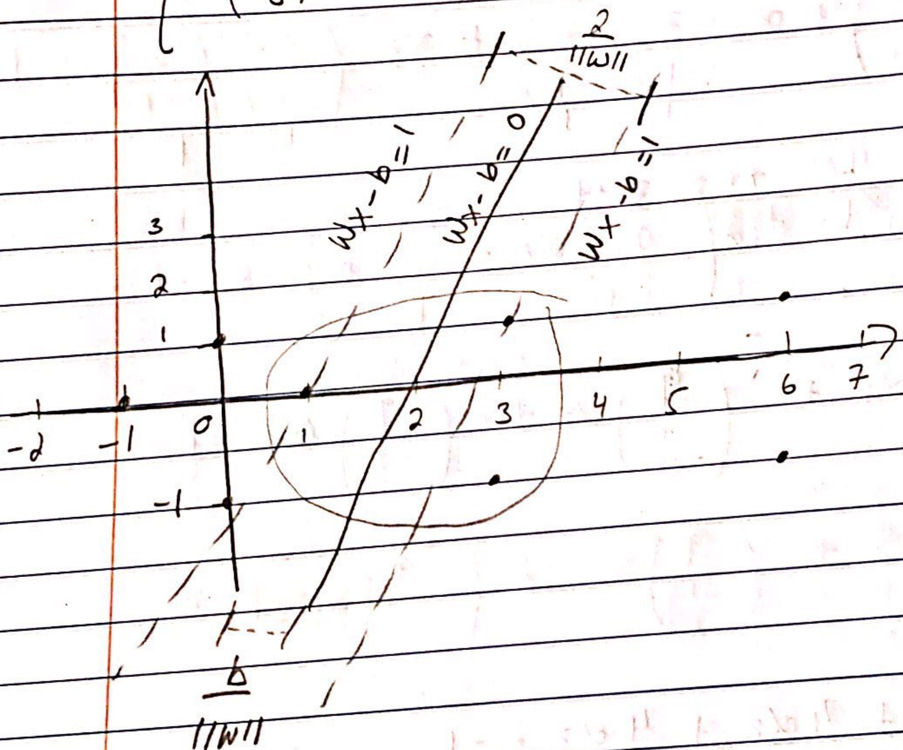
Linearly Separable case +vely

ⓐ Given the following labeled data points R^2

$$\left\{ \begin{pmatrix} 3 \\ 1 \end{pmatrix}, \begin{pmatrix} 3 \\ -1 \end{pmatrix}, \begin{pmatrix} 6 \\ 1 \end{pmatrix}, \begin{pmatrix} 6 \\ -1 \end{pmatrix} \right\}$$

And the following -vely labeled data points R^2

$$\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix} \right\}$$



∴ the three support vectors are :-

$$\left\{ S_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, S_2 = \begin{pmatrix} 3 \\ 1 \end{pmatrix}, S_3 = \begin{pmatrix} 3 \\ -1 \end{pmatrix} \right\}$$

$$\alpha_1 \bar{\Phi}(s_1)\cdot\bar{\Phi}(s_1) + \alpha_2 \bar{\Phi}(s_2)\cdot\bar{\Phi}(s_1) + \alpha_3 \bar{\Phi}(s_3)\cdot\bar{\Phi}(s_1) = -1$$

$$\alpha_1 \bar{\Phi}(s_1)\cdot\bar{\Phi}(s_2) + \alpha_2 \bar{\Phi}(s_2)\cdot\bar{\Phi}(s_2) + \alpha_3 \bar{\Phi}(s_3)\cdot\bar{\Phi}(s_2) = +1$$

$$\alpha_1 \bar{\Phi}(s_1)\cdot\bar{\Phi}(s_3) + \alpha_2 \bar{\Phi}(s_2)\cdot\bar{\Phi}(s_3) + \alpha_3 \bar{\Phi}(s_3)\cdot\bar{\Phi}(s_3) = +1$$

Syed Ali

For now we have $\phi() = I$, this reduces to:-

$\alpha \tilde{S}_1 \cdot \tilde{S}_1 + \alpha_2 \tilde{S}_2 \cdot \tilde{S}_1 + \alpha_3 \tilde{S}_3 \cdot \tilde{S}_1 = 1$

$\alpha_1 \tilde{S}_1 \cdot \tilde{S}_2 + \alpha_2 \tilde{S}_2 \cdot \tilde{S}_2 + \alpha_3 \tilde{S}_3 \cdot \tilde{S}_2 = +1$

$\alpha_1 \tilde{S}_1 \cdot \tilde{S}_3 + \alpha_2 \tilde{S}_2 \cdot \tilde{S}_3 + \alpha_3 \tilde{S}_3 \cdot \tilde{S}_3 = +1$

$$\alpha_1 \begin{pmatrix}1\\0\\1\end{pmatrix}\begin{pmatrix}1\\0\\1\end{pmatrix} + \alpha_2 \begin{pmatrix}3\\1\\1\end{pmatrix}\begin{pmatrix}1\\0\\1\end{pmatrix} + \alpha_3 \begin{pmatrix}3\\-1\\1\end{pmatrix}\begin{pmatrix}1\\0\\1\end{pmatrix} = -1$$

$$\alpha_1 \begin{pmatrix}1\\0\\1\end{pmatrix}\begin{pmatrix}3\\1\\1\end{pmatrix} + \alpha_2 \begin{pmatrix}3\\1\\1\end{pmatrix}\begin{pmatrix}3\\1\\1\end{pmatrix} + \alpha_3 \begin{pmatrix}3\\-1\\1\end{pmatrix}\begin{pmatrix}3\\1\\1\end{pmatrix} = 1$$

$$\alpha_1 \begin{pmatrix}1\\0\\1\end{pmatrix}\begin{pmatrix}3\\-1\\1\end{pmatrix} + \alpha_2 \begin{pmatrix}3\\1\\1\end{pmatrix}\begin{pmatrix}3\\-1\\1\end{pmatrix} + \alpha_3 \begin{pmatrix}3\\-1\\1\end{pmatrix}\begin{pmatrix}3\\-1\\1\end{pmatrix} = 1$$

$$\begin{pmatrix}1\\0\\1\end{pmatrix}\alpha_1 + \begin{pmatrix}3\\0\\1\end{pmatrix}\alpha_2 + \begin{pmatrix}3\\0\\1\end{pmatrix}\alpha_3 = -1$$

$$\begin{pmatrix}3\\0\\1\end{pmatrix}\alpha_1 + \begin{pmatrix}9\\1\\1\end{pmatrix}\alpha_2 + \begin{pmatrix}9\\-1\\1\end{pmatrix}\alpha_3 = 1$$

$$\begin{pmatrix}3\\0\\1\end{pmatrix}\alpha_1 + \begin{pmatrix}9\\-1\\1\end{pmatrix}\alpha_2 + \begin{pmatrix}9\\1\\1\end{pmatrix}\alpha_3 = 1$$

$2\alpha_1 + 4\alpha_2 + 4\alpha_3 = -1$

$4\alpha_1 + 11\alpha_2 + 9\alpha_3 = 1$

$4\alpha_1 + 9\alpha_2 + 11\alpha_3 = 1$

$\alpha_1 = \dfrac{-1 - 4\alpha_3 - 4\alpha_2}{2}$ $\qquad \alpha_2 = \dfrac{2 - 5\alpha_3}{7}$

$\cancel{4}\left(\dfrac{-1 - 4\alpha_3 - 4\alpha_2}{\cancel{2}}\right) + 11\alpha_2 + 9\alpha_3 = 1$ $\qquad 14\alpha_1 = -1 - 28\alpha_3 - 56 + 14\alpha$

$-1 - 4\alpha_3 - 4\alpha_2 + 11\alpha_2 + 9\alpha_3 = 1$ $\qquad 2\alpha_1 = -1 - 4\alpha_3 - 4\left(\dfrac{2-5\alpha}{7}\right.$

$5\alpha_3 + 7\alpha_2 = 2$ $\qquad 2\alpha_1 = -1 - 4\alpha_3 - 8 + 20\alpha_3 / 7$

Syed Ali

$$\boxed{\alpha_1 = -3.5\ ,\ \alpha_2 = 0.75,\ \alpha_3 = 0.75}$$

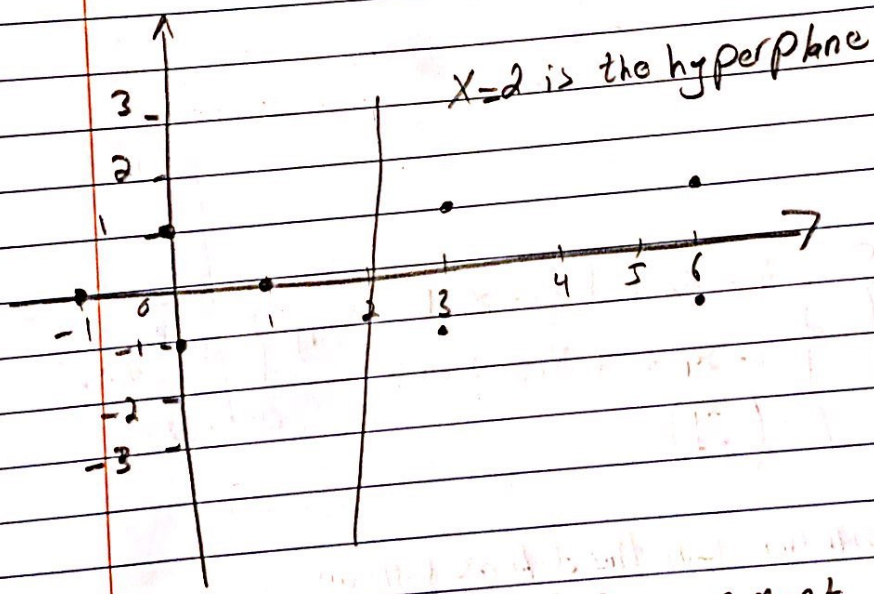
$$\tilde{w} = \sum_i \alpha_i s_i$$

$$\tilde{w} = -3.5\begin{pmatrix}1\\0\\1\end{pmatrix} + 0.75\begin{pmatrix}3\\1\\1\end{pmatrix} + 0.75\begin{pmatrix}3\\-1\\1\end{pmatrix}$$

$$w \Rightarrow \boxed{\begin{pmatrix}1\\0\\-2\end{pmatrix}} \qquad y = \begin{pmatrix}1\\0\end{pmatrix}x - 2$$

$$y = x - 2$$

So, the hyperplane is :-



See final plot at end of assignment

Scanned with CamScanner

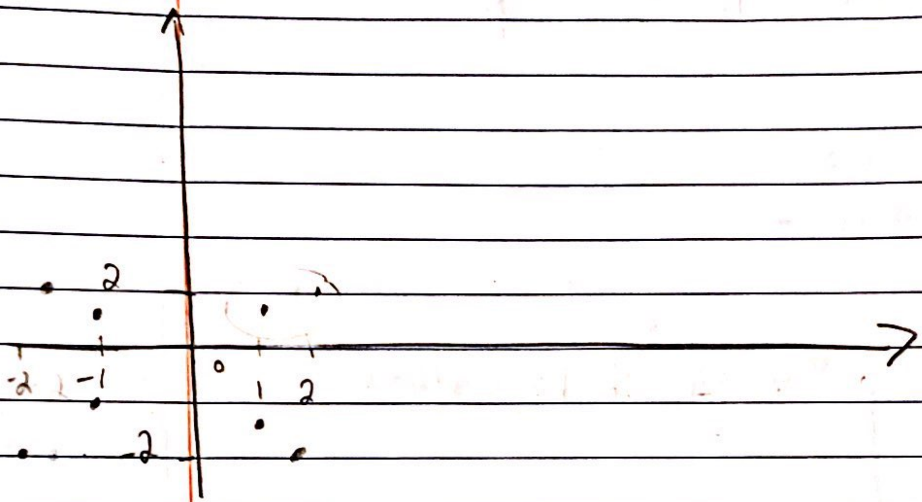
Syed Ali

(b) Non-linearly Separable case

(i) Given the following labeled data points:-

$$\left\{ \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ -2 \end{pmatrix} \right\}$$

$$\left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix} \right\}$$



Now Define

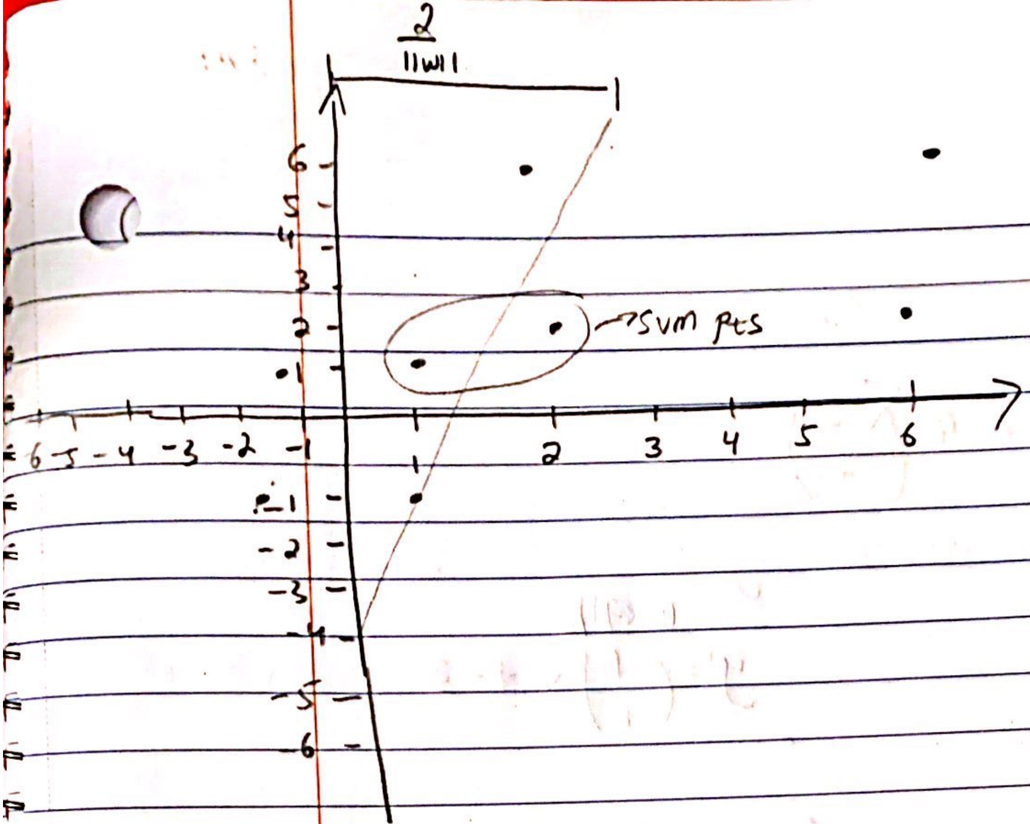
$$\Phi \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{cases} \begin{pmatrix} 4 - x_2 + |x_1 - x_2| \\ 4 - x_1 + |x_1 - x_2| \end{pmatrix} & \text{if } \sqrt{x_1^2 + x_2^2} > 2 \\ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} & \text{otherwise} \end{cases}$$

So, now this will transform the data as follows

$$\left\{ \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 6 \\ 2 \end{pmatrix}, \begin{pmatrix} 6 \\ 6 \end{pmatrix}, \begin{pmatrix} 6 \\ 6 \end{pmatrix} \right\} \text{ +ves}$$

$$\left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix} \right\} \text{ -ves}$$

Syed Ali



$$\left\{ S_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, S_2 = \begin{pmatrix} 2 \\ 2 \end{pmatrix} \right\}$$

$$\alpha_1 \phi(s_1) \cdot \phi(s_1) + \alpha_2 \phi(s_2) \cdot \phi(s_1) = -1$$
$$\alpha_1 \phi(s_1) \cdot \phi(s_2) + \alpha_2 \phi(s_2) \cdot \phi(s_1) = 1$$

this reduces to

$$\alpha_1 \tilde{s}_1 \cdot \tilde{s}_1 + \alpha_2 \tilde{s}_2 \cdot \tilde{s}_1 = -1$$
$$\alpha_1 \tilde{s}_1 \cdot \tilde{s}_2 + \alpha_2 \tilde{s}_2 \cdot \tilde{s}_2 = +1$$

$$\alpha_1 \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = -1$$

$$\alpha_1 \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix} \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix} = 1$$

$$\alpha_1 \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} = -1 \qquad 3\alpha_1 + 5\alpha_2 = -1$$

$$\Rightarrow \quad 5\alpha_1 + 9\alpha_2 = 1$$

$$\alpha_1 \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 4 \\ 4 \\ 1 \end{pmatrix} = 1 \qquad \alpha_1 = \frac{1 - 9\alpha_2}{5}$$

$$\left(3\left(\frac{1 - 9\alpha_2}{5}\right) + 5\alpha_2 = -1\right)5$$

$$3 - 27\alpha_2 + 25\alpha_2 = -5$$

Syed Al

$$\boxed{\begin{aligned}\alpha_2 &= 4\\ \alpha_1 &= -7\end{aligned}}$$

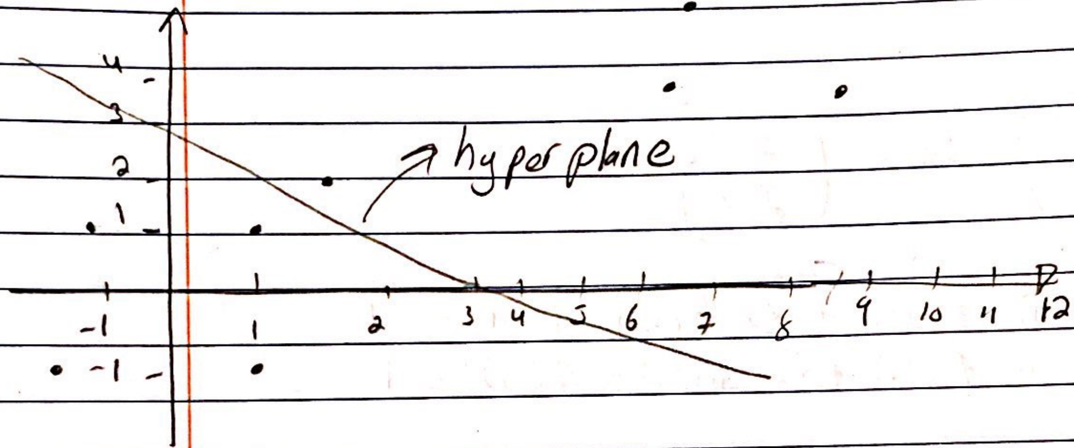
$$\bar{w} = \sum_i \alpha_i \tilde{s}_i$$

$$\Rightarrow -7\begin{pmatrix}1\\1\\1\end{pmatrix} + 4\begin{pmatrix}2\\2\\1\end{pmatrix}$$

$$w \Rightarrow \begin{pmatrix}1\\1\\-3\end{pmatrix}$$

$$y = wx + b$$

$$y = \begin{pmatrix}1\\1\end{pmatrix}x \cancel{+} -3$$



(ii) <u>Kernel trick</u>

Here we will use a new mapping fcn

$$\Phi_2\begin{pmatrix}x_1\\x_2\end{pmatrix} = \begin{pmatrix}x_1\\x_2\\ \dfrac{(x_1^2 + x_2^2) - 5}{3}\end{pmatrix}$$

$\frac{4+4-5}{3}$, $8-5$

This will transform it from a 2D - 3D space

$$\left\{\begin{pmatrix}2\\2\\1\end{pmatrix}, \begin{pmatrix}2\\-2\\1\end{pmatrix}, \begin{pmatrix}-2\\-2\\1\end{pmatrix}, \begin{pmatrix}-2\\2\\1\end{pmatrix}\right\} \text{+ves}$$

$$\left\{\begin{pmatrix}1\\1\\-1\end{pmatrix}, \begin{pmatrix}1\\-1\\-1\end{pmatrix}, \begin{pmatrix}-1\\-1\\-1\end{pmatrix}, \begin{pmatrix}-1\\1\\-1\end{pmatrix}\right\} \text{-ves}$$

See plot attached

$$\left\{ s_1 = \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix}, s_2 \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix} \right\}$$

$$\alpha_1 \tilde{s}_1 \cdot \tilde{s}_1 + \alpha_2 \tilde{s}_2 \cdot \tilde{s}_1 = -1$$
$$\alpha_1 \tilde{s}_1 \cdot \tilde{s}_2 + \alpha_2 \tilde{s}_2 \cdot \tilde{s}_1 = +1$$

$$\alpha_1 \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix} \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix} \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix} = -1$$

$$\alpha_1 \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix} \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix} + \alpha_2 \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix} \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix} = 1$$

$$\begin{pmatrix} 4 \\ 4 \\ 1 \end{pmatrix} \alpha_1 - \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix} \alpha_2 = -1$$

$$-\alpha_1 \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \alpha_2 = 1$$

$$9\alpha_1 - 5\alpha_2 = -1$$
$$-5\alpha_1 + 3\alpha_2 = 1$$

$$\alpha_2 = \frac{1+5\alpha_1}{3}$$

$32\alpha_1 = -2$

$$9\alpha_1 - 5\left(\frac{1+5\alpha_1}{3}\right) = -1$$

$$-\frac{7}{46} \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix} + \frac{1}{46} \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix}$$

$$21\alpha_1 + 5 + 25\alpha_1 = -2$$

$$\boxed{\alpha_2 = \frac{1}{46}} \quad \boxed{\alpha_1 = \frac{-2}{46}}$$

$$\begin{pmatrix} -\frac{14}{46} \\ -\frac{14}{46} \\ -\frac{7}{46} \end{pmatrix} + \begin{pmatrix} -\frac{1}{46} \\ -\frac{1}{46} \\ -\frac{1}{46} \end{pmatrix} \Rightarrow \begin{pmatrix} -\frac{15}{46} \\ -\frac{15}{46} \\ -\frac{8}{46} \end{pmatrix}$$

Syed Ali

$\bar{w} = \sum_i \alpha_i \bar{s}_i$ → passes through the origin

$$\boxed{\bar{w} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}}$$

$y = wx + b \;=\; \therefore y = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} x + \overset{b}{0}$

$b = 0$

See final plots attached at the end of this assignment

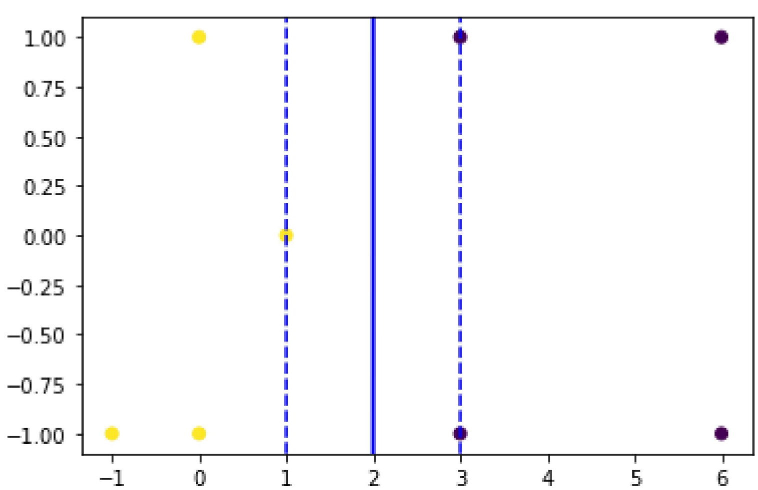
SVM Example Assignment 6

Part a using Linearly Separable Case

In [1]:

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

def __intersect(rect, line):
    l = []
    xmin,xmax,ymin,ymax = rect
    a,b,c = line

    assert a!=0 or b!=0

    if a == 0:
        y = -c/b
        if y<=ymax and y>=ymin:
            l.append((xmin, y))
            l.append((xmax, y))
        return l
    if b == 0:
        x = -c/a
        if x<=xmax and x>=xmin:
            l.append((x, ymin))
            l.append((x, ymax))
        return l

    k = -a/b
    m = -c/b
    for x in (xmin, xmax):
        y = k*x+m
        if y<=ymax and y>= ymin:
            l.append((x,y))

    k = -b/a
    m = -c/a
    for y in (ymin, ymax):
        x = k*y+m
        if x<xmax and x> xmin:
            l.append((x,y))
    return l


def plotline(coef, *args, **kwargs):
    '''plot line: y=a*x+b or a*x+b*y+c=0'''
    coef = np.float64(coef[:])
    assert len(coef)==2 or len(coef)==3
    if len(coef) == 2:
        a, b, c = coef[0], -1., coef[1]
    elif len(coef) == 3:
        a, b, c = coef
    ax = plt.gca()

    limits = ax.axis()
    points = __intersect(limits, (a,b,c))
    if len(points) == 2:
```

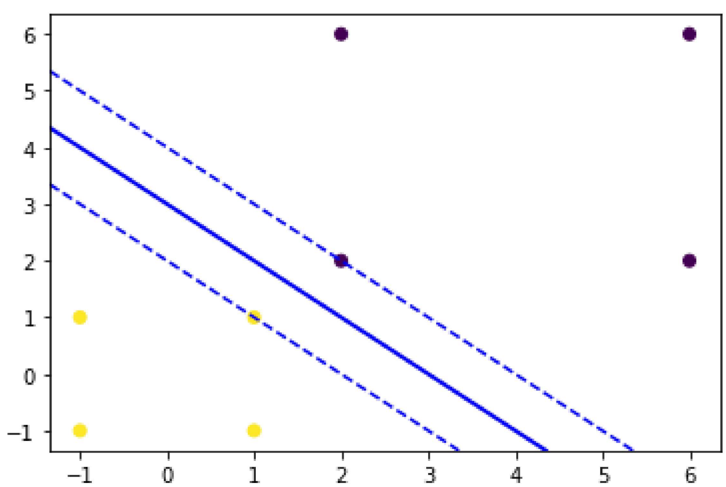
```
        pts = np.array(points)
        ax.plot(pts[:,0], pts[:,1], *args, **kwargs)
        ax.axis(limits)

def circle_out(x, y, s=20, *args, **kwargs):
    '''Circle out points with size 's' and edgecolors'''
    ax = plt.gca()
    if 'edgecolors' not in kwargs:
        kwargs['edgecolors'] = 'g'
    ax.scatter(x, y, s, facecolors='none', *args, **kwargs)


def plotSVM(coef, support_vectors=None):
    coef1 = coef[:]
    coef2 = coef[:]
    coef1[2] += 1
    coef2[2] -= 1
    plotline(coef, 'b', lw=2)
    plotline(coef1, 'b', ls='dashed')
    plotline(coef2, 'b', ls='dashed')



from pylab import *

X = array([[3.0,  3.0, 6.0, 6.0],
              [1.0, -1.0, 1.0, -1.0]])

Y = array([[1.0, 0.0,  0.0, -1.0],
              [0.0, 1.0, -1.0, -1.0]])

data = hstack((X,Y)).T
label = hstack((zeros(X.shape[1]), ones(Y.shape[1])))

from sklearn.svm import SVC

clf = SVC(kernel='linear')
clf.fit(data, label)

coef = [clf.coef_[0,0], clf.coef_[0,1], clf.intercept_[0]]
scatter(data[:,0], data[:,1], c=label)
plotSVM(coef, clf.support_vectors_)
show()
```



Part b (i) using Non-Linearly Separable Case

In [2]:
```
import numpy as np
import matplotlib.pyplot as plt

def __intersect(rect, line):
    l = []
    xmin,xmax,ymin,ymax = rect
    a,b,c = line

    assert a!=0 or b!=0

    if a == 0:
        y = -c/b
        if y<=ymax and y>=ymin:
            l.append((xmin, y))
            l.append((xmax, y))
        return l
    if b == 0:
        x = -c/a
        if x<=xmax and x>=xmin:
            l.append((x, ymin))
            l.append((x, ymax))
        return l

    k = -a/b
    m = -c/b
    for x in (xmin, xmax):
        y = k*x+m
        if y<=ymax and y>= ymin:
            l.append((x,y))

    k = -b/a
    m = -c/a
    for y in (ymin, ymax):
        x = k*y+m
        if x<xmax and x> xmin:
            l.append((x,y))
    return l


def plotline(coef, *args, **kwargs):
    '''plot line: y=a*x+b or a*x+b*y+c=0'''
    coef = np.float64(coef[:])
    assert len(coef)==2 or len(coef)==3
    if len(coef) == 2:
        a, b, c = coef[0], -1., coef[1]
    elif len(coef) == 3:
        a, b, c = coef
    ax = plt.gca()

    limits = ax.axis()
    points = __intersect(limits, (a,b,c))
    if len(points) == 2:
        pts = np.array(points)
        ax.plot(pts[:,0], pts[:,1], *args, **kwargs)
        ax.axis(limits)

def circle_out(x, y, s=20, *args, **kwargs):
    '''Circle out points with size 's' and edgecolors'''
```

```
    ax = plt.gca()
    if 'edgecolors' not in kwargs:
        kwargs['edgecolors'] = 'g'
    ax.scatter(x, y, s, facecolors='none', *args, **kwargs)


def plotSVM(coef, support_vectors=None):
    coef1 = coef[:]
    coef2 = coef[:]
    coef1[2] += 1
    coef2[2] -= 1
    plotline(coef, 'b', lw=2)
    plotline(coef1, 'b', ls='dashed')
    plotline(coef2, 'b', ls='dashed')



from pylab import *

X = array([[2.0,  6.0, 6.0, 2.0],
              [2.0, 2.0, 6.0, 6.0]])

Y = array([[1.0, 1.0,  -1.0, -1.0],
              [1.0, -1.0, -1.0, 1.0]])

data = hstack((X,Y)).T
label = hstack((zeros(X.shape[1]), ones(Y.shape[1])))

from sklearn.svm import SVC

clf = SVC(kernel='linear')
clf.fit(data, label)

coef = [clf.coef_[0,0], clf.coef_[0,1], clf.intercept_[0]]
scatter(data[:,0], data[:,1], c=label)
plotSVM(coef, clf.support_vectors_)
show()
```



Part b (ii) using Kernel Trick

In [3]:
```
import numpy as np
import matplotlib.pyplot as plt
```

```python
def __intersect(rect, line):
    l = []
    xmin,xmax,ymin,ymax = rect
    a,b,c = line

    assert a!=0 or b!=0

    if a == 0:
        y = -c/b
        if y<=ymax and y>=ymin:
            l.append((xmin, y))
            l.append((xmax, y))
        return l
    if b == 0:
        x = -c/a
        if x<=xmax and x>=xmin:
            l.append((x, ymin))
            l.append((x, ymax))
        return l

    k = -a/b
    m = -c/b
    for x in (xmin, xmax):
        y = k*x+m
        if y<=ymax and y>= ymin:
            l.append((x,y))

    k = -b/a
    m = -c/a
    for y in (ymin, ymax):
        x = k*y+m
        if x<xmax and x> xmin:
            l.append((x,y))
    return l


def plotline(coef, *args, **kwargs):
    '''plot line: y=a*x+b or a*x+b*y+c=0'''
    coef = np.float64(coef[:])
    assert len(coef)==2 or len(coef)==3
    if len(coef) == 2:
        a, b, c = coef[0], -1., coef[1]
    elif len(coef) == 3:
        a, b, c = coef
    ax = plt.gca()

    limits = ax.axis()
    points = __intersect(limits, (a,b,c))
    if len(points) == 2:
        pts = np.array(points)
        ax.plot(pts[:,0], pts[:,1], *args, **kwargs)
        ax.axis(limits)

def circle_out(x, y, s=20, *args, **kwargs):
    '''Circle out points with size 's' and edgecolors'''
    ax = plt.gca()
    if 'edgecolors' not in kwargs:
        kwargs['edgecolors'] = 'g'
    ax.scatter(x, y, s, facecolors='none', *args, **kwargs)
```

```python
def plotSVM(coef, support_vectors=None):
    coef1 = coef[:]
    coef2 = coef[:]
    coef1[2] += 1
    coef2[2] -= 1
    plotline(coef, 'b', lw=2)
    plotline(coef1, 'b', ls='dashed')
    plotline(coef2, 'b', ls='dashed')



from pylab import *

X = array([[2.0,  2.0, -2.0, -2.0],
              [2.0, -2.0, -2.0, 2.0],[1.0,1.0,1.0,1.0]])

Y = array([[1.0, 1.0,  -1.0, -1.0],
              [1.0, -1.0, -1.0, 1.0],[-1.0,-1.0,-1.0,-1.0]])

data = hstack((X,Y)).T
label = hstack((zeros(X.shape[1]), ones(Y.shape[1])))

from sklearn.svm import SVC

clf = SVC(kernel='linear')
clf.fit(data, label)

coef = [clf.coef_[0,0], clf.coef_[0,1], clf.intercept_[0]]
scatter(data[:,0], data[:,1], c=label)
plotSVM(coef, clf.support_vectors_)
show()
```