CSCI3230 / ESTR3108 2023-24 First Term Assignment 2

I declare that the assignment here submitted is original except for source material explicitly acknowledged, and that the same or closely related material has not been previously submitted for another course. I also acknowledge that I am aware of University policy and regulations on honesty in academic work, and of the disciplinary guidelines and procedures applicable to breaches of such policy and regulations, as contained in the following websites. University Guideline on Academic Honesty: http://www.cuhk.edu.hk/policy/academichonesty/ Faculty of Engineering Guidelines to Academic Honesty: http://www.erg.cuhk.edu.hk/erg-intra/upload/documents/ENGG_Discipline.pdf

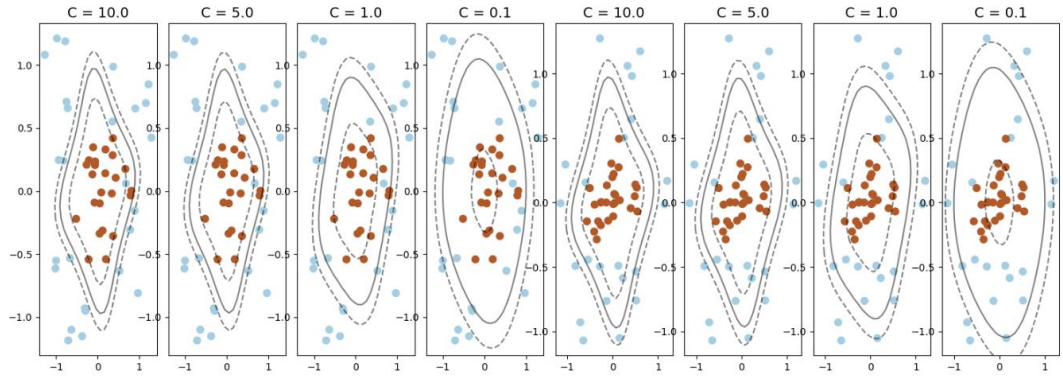
Student Name: Fong Long Wai

Student ID :1155177220

Q2:

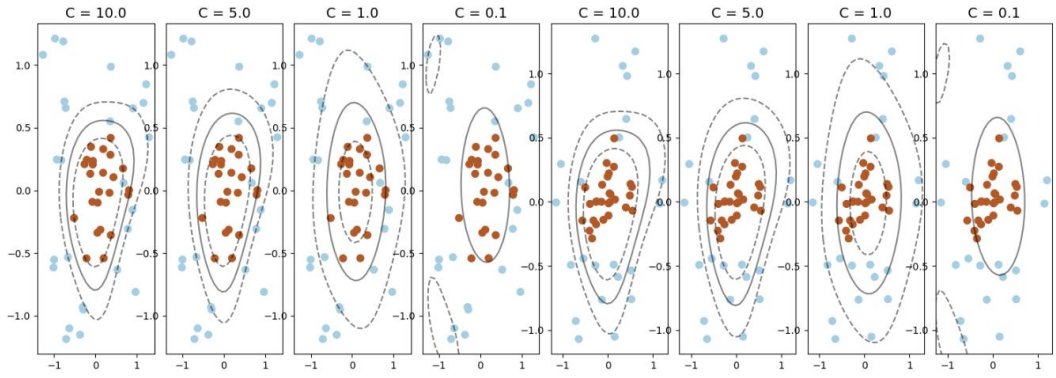
The first model is using Polynomial kernel function with a degree of 4, c is 10. The output accuracy is 0.86. (graph 1 and graph 5)

```
0.86
0.86
0.82
0.78
```



The second model is using Radial basis function kernel with a degree of 1, c is 0.1. The output accuracy is 0.9. (graph 4 and graph 8)

```
0.86
0.88
0.9
0.9
```



Q2 code:

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import make_blobs
from sklearn import svm
def getTest():
    test_csv_file='./test.csv';
    testtemp = np.genfromtxt(test_csv_file, delimiter=',',
skip_header=1)
    testy=testtemp[:,2];
    testX=testtemp[:,:2];
    return testy,testX;

def plot_svc_decision_function(model, ax=None, plot_support=True):
    """Plot the decision function for a 2D SVC"""
    if ax is None:
        ax = plt.gca()
    xlim = ax.get_xlim()
    ylim = ax.get_ylim()

    # create grid to evaluate model
    x = np.linspace(xlim[0], xlim[1], 30)
    y = np.linspace(ylim[0], ylim[1], 30)
    Y, X = np.meshgrid(y, x)
    xy = np.vstack([X.ravel(), Y.ravel()]).T
    P = model.decision_function(xy).reshape(X.shape)

    # plot decision boundary and margins
    ax.contour(X, Y, P, colors='k',
               levels=[-1, 0, 1], alpha=0.5,
               linestyles=['--', '-', '--'])
    """
    # plot support vectors
    if plot_support:
        ax.scatter(model.support_vectors_[:, 0],
                   model.support_vectors_[:, 1],
                   s=100, linewidth=1,
facecolors='none',edgecolors='k')
```

```
    """
    ax.set_xlim(xlim)
    ax.set_ylim(ylim)


"""X, y = make_blobs(n_samples=30, centers=2,
                  random_state=0, cluster_std=1.2)
"""
csv_file='./training.csv';
temp = np.genfromtxt(csv_file, delimiter=',', skip_header=1)
y=temp[:,2];
X=temp[:,:2];
plt.scatter(X[:, 0], X[:, 1], c=y, s=50,cmap=plt.cm.Paired)

# %%
superparameter=[10,5,1,0.1];
fig, ax = plt.subplots(1,len(superparameter)*2, figsize=(16, 6))
fig.subplots_adjust(left=0.0625, right=0.95, wspace=0.1)
model=[]

for C in superparameter:
    model.append(svm.SVC(kernel="poly",degree=4, C=C).fit(X, y))
for axi, m,C in zip(ax, model,superparameter):
    axi.scatter(X[:, 0], X[:, 1], c=y, s=50, cmap=plt.cm.Paired)
    plot_svc_decision_function(m, axi)
    axi.scatter(m.support_vectors_[:, 0],
                m.support_vectors_[:, 1],
                s=300, lw=1, facecolors='none');
    axi.set_title('C = {0:.1f}'.format(C), size=14)

testy,testX=getTest();

for axi, m,C in zip(ax[len(superparameter):], model,superparameter):
    axi.scatter(testX[:, 0], testX[:, 1], c=testy, s=50,
cmap=plt.cm.Paired)
    plot_svc_decision_function(m, axi)
    axi.set_title('C = {0:.1f}'.format(C), size=14)
for m in model:
    result=m.predict(testX);
```

```
accuracy=(testy==result);
print(accuracy.sum()/len(accuracy));
```

a_i as are support vectors

1a. Dual ~~Optimization~~ problem:

$$\max_{\alpha} \sum_{i=1}^{8} a_i - \frac{1}{2} \sum_{i=1}^{8} \sum_{j=1}^{8} a_i a_j y_i y_j x_i^T x_j$$

$$\text{s.t. } \sum_{i=1}^{8} a_i y_i = 0, \quad a_i \geq 0, \quad i = 1 \dots m$$

Optimization problem:

$$\min \frac{1}{2} \|w\|^2 + \sum_{i=1}^{8} a_i (1 - y_i (w^T x_i + b))$$

$$w^* = \sum_{i=1}^{8} a_i y_i x_i = 0.32 \begin{pmatrix} 8 \\ -8 \end{pmatrix} - 0.32 \begin{pmatrix} 8 \\ -5.5 \end{pmatrix}$$

$$= \begin{pmatrix} 0 \\ -0.8 \end{pmatrix}$$

$$b^* = \frac{1}{|S|} \sum_{s \in S} \left(\frac{1}{y_s} - w^T x_s \right)$$

$$= \frac{1}{2} \left[\left(1 - (0 \quad -0.8) \begin{pmatrix} 8 \\ -8 \end{pmatrix} \right) + \left(-1 - (0 \quad -0.8) \begin{pmatrix} 8 \\ -5.5 \end{pmatrix} \right) \right]$$

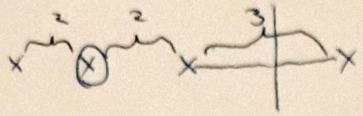
$$= \frac{1}{2} [-5.4 + (-5.4)]$$

$$= -5.4$$

$$\therefore f(x) = \begin{pmatrix} 0 \\ -0.8 \end{pmatrix}^T x - 5.4$$

No.

Date.



1b. Prediction $y_i = \begin{cases} 1 & \text{if } f(x_i) \geq 0 \\ -1 & \text{if } f(x_i) < 0 \end{cases}$

$= \begin{cases} 1 & \text{if } -0.8x_{i2} \geq 5.4 \\ -1 & \text{if } -0.8x_{i2} \leq 5.4 \end{cases}$

$= \begin{cases} 1 & \text{if } x_{i2} \leq -6.75 \\ -1 & \text{if } x_{i2} \geq -6.75 \end{cases}$

i	y_i
1	1
2	1
3	1
4	1
5	-1
6	-1
7	-1
8	-1

Assign case for $f(x_i) = 0$ to the positive class.

1c. No, for removing 2nd since it is not a support vector.
Yes for removing 8th as it is a support vector.

4.

by max distance:

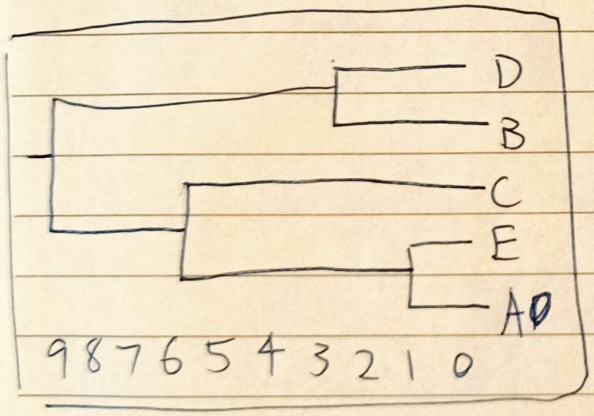
	AB	CD	E
A	0		
B	8	0	
CD	6	7	0
E	1	6	5	0

	AE	B	C	D
AE	0			
B	8	0		
C	6	7	0	
D	4	3	9	0

	AB	CD	E
AB	0		
CD		0	
E			0

	AE	BD	C
AE	0		
BD	8	0	
C	6	9	0

	AEC	BD
AEC	0	
BD	9	0

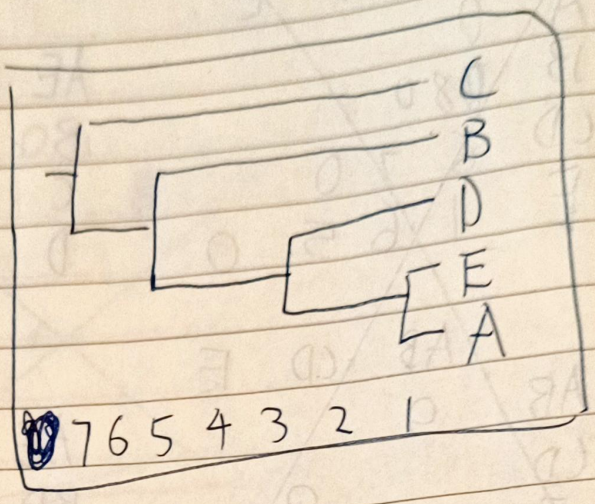


No.

Date.

by group average distance:

	AE	B	C	D
AE	0			
B	7	0		
C	5.5	7	0	
D	3	3	9	0



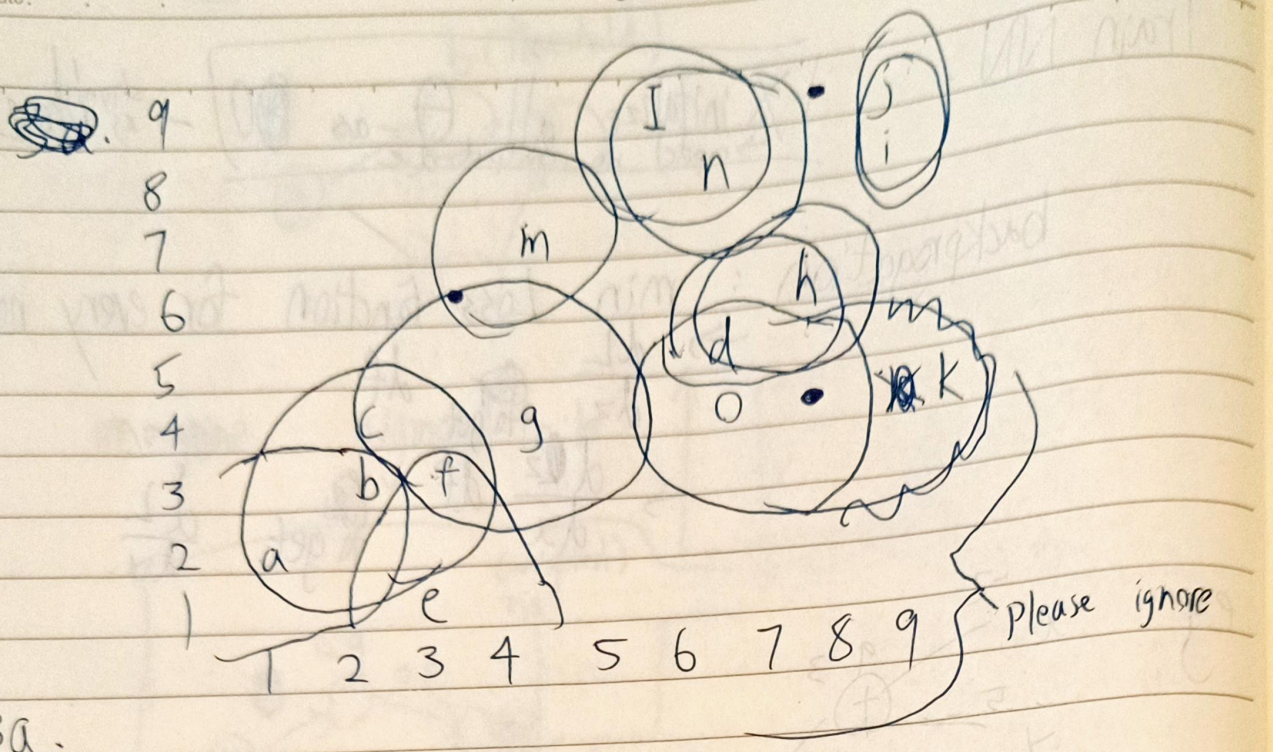
	AED	B	C
AED	0		
B	$\frac{17}{3}$	0	
C	$\frac{20}{3}$	7	0

	AEDB	C
AEDB	0	
C	$\frac{27}{4}$	0

No.

Date.

hw 2



3a.

The cluster are: {I j n i},
{d h o k},
{a b c e f g m}

updated cluster centers

center 1 $= \left(\frac{5+6+8+8}{4}, \frac{9+8+8+8}{4}\right) = (6.75, 8.25)$

center 2 $= \left(\frac{6+6+7+9}{4}, \frac{4+4+5+6}{4}\right) = (7, 4.75)$

center 3 $= \left(\frac{1+2+2+3+3+4+4}{7}, \frac{1+2+3+3+4+4+7}{7}\right)$

$= \left(\frac{19}{7}, \frac{24}{7}\right)$

3b. Core points: b, c, f, d, g, o, n, i
Border points: a, e, h, j, ~~k~~ L
Noise points: k, ~~l~~, m

c. ~~No. The point where chain breaks is a since a is not a core point.~~

Yes. {a, b, f, g, o, d, h}

d. Clusters are:
{a, b, c, f, e, g, o, d, h},
{i, j, n, L}