CSCI3230 / ESTR3108: Fundamentals of Artificial Intelligence 2022-2023 Term 1
The Chinese University of Hong Kong

Assignment 2

Due date: 24 October 2022 (Tue) 23:59

Full mark: 100
Expected time spent: 3-5 hours

Aims: 1. Understand the knowledge about Support Vector Machine Hands-on practice of the optimization problem in SVM.
2. and Clustering.
3. Hands-on practice of implementation processes of K-means, DBSCAN and Hierarchical clustering.
4. Get familiar with how to use some tools (e.g., scikit-learn) to implement SVM.

Description:

In Assignment 2, you will practice how to build a SVM classifier on a training set and evaluate it on a test set. Here, you will know how to use some toolboxes to implement SVM. You will practice on using concepts of K-means, DBSCAN and Hierarchical clustering to solve problems.

For some calculations, you can use the toolbox in Python or MATLAB or any other programming languages you are familiar with.

Questions:

1. Consider the following training and test data set:

Training set

Index	x_1	x_2	y
1	8	-8	1
2	6	-9.5	1
3	9	-8.5	1
4	6.5	-9	1
5	5.5	-2.5	-1
6	7	-3	-1
7	9.5	-3.5	-1
8	8	-5.5	-1

test set

Index	x_1	x_2
1	5	-7
2	5.5	-9
3	8	-8
4	9	-8
5	7	-2.5
6	8	-5
7	5	-5
8	6.5	-4

(a) Set up the optimization problem using $(\alpha_1, \alpha_2, \dots, \alpha_8)$ and write down the dual problem of optimization. Then, given that the optimal $\alpha_1 = \alpha_8 = 0.32$, $\alpha_2 = \alpha_3 = \alpha_4 = \alpha_5 = \alpha_6 = \alpha_7 = 0$. Based on these $\boldsymbol{\alpha}$, which are the support vectors? Then, calculate the function of optimal hyperplane of this model. (10%)

Solution:

The optimization problem is:

$$\max_{\alpha} \sum_{i=1}^{8} \alpha_i - \frac{1}{2} \sum_{i=1}^{8} \sum_{j=1}^{8} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j \text{ (plug the values of x1 and x2)}$$

$$s.t. \sum_{i=8}^{8} \alpha_i y_i = 0, \alpha_i \geq 0$$ .... (2%)

The support vectors are the 1st, 8th data points .... (2%)

The optimal $w^* = \sum_{i=1}^{8} \alpha_i y_i x_i = \binom{0}{-0.8}$ .... (3%)

The optimal $b^* = \frac{1}{|S|} \sum_{s \in S} \left(\frac{1}{y_s} - w^T \boldsymbol{x}_s \right) = -5.4$ .... (3%)

The hyperplane is $w^* \boldsymbol{x} + b^* = 0$

(b) Given the test data points, how can we use the optimal hyperplane to predict them? Please write corresponding formulas and then get the predictions. (10%)

Solution:

$$\hat{y_i} = \begin{cases} 1, & if\ w^* \boldsymbol{x} + b^* > 0 \\ -1, & if\ w^* \boldsymbol{x} + b^* < 0 \end{cases} \quad \cdots\cdots (5\%)$$

The values of $w^* \boldsymbol{x} + b^*$ on the test data points are:

$$(0.2, 1.8, 1\ , 1, -3.4, -1.4, -1.4, -2.2)$$

So the predictions are (set the threshold as 0.5):

$$(1,1,1,1,-1,-1,-1,-1) \cdots\cdots (5\%)$$

(c) If we remove the 2nd data point in training set and use the remaining 7 points to train the SVM model, will the prediction of test data change? How about removing the 8th data point? (5%)

Solution:

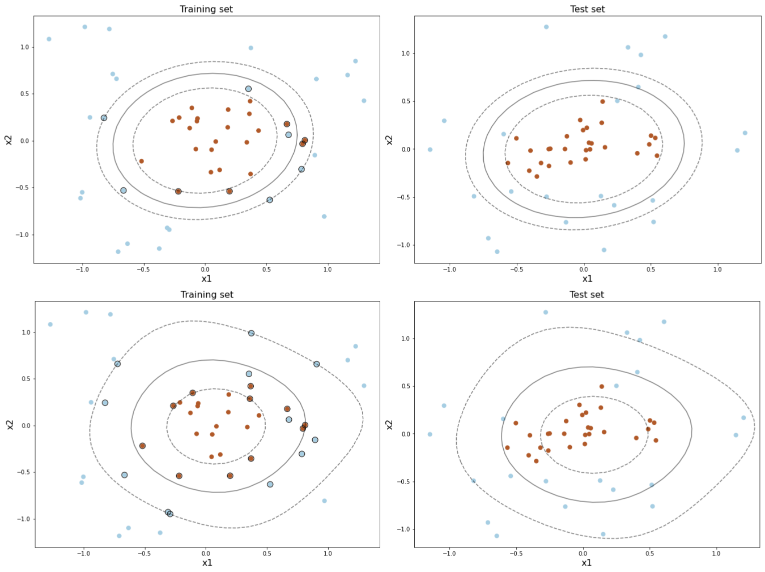
As we have three support vectors (the 1st, 8th data points) and the hyperplane is only optimized by the support vectors, removing the 2nd will not change the optimal hyperplane. For the 8th data point is the only negative support vector, which means removing it must influence the optimal hyperplane. However, if you use this new optimal hyperplane, the prediction of test data will not change. .... (5%)

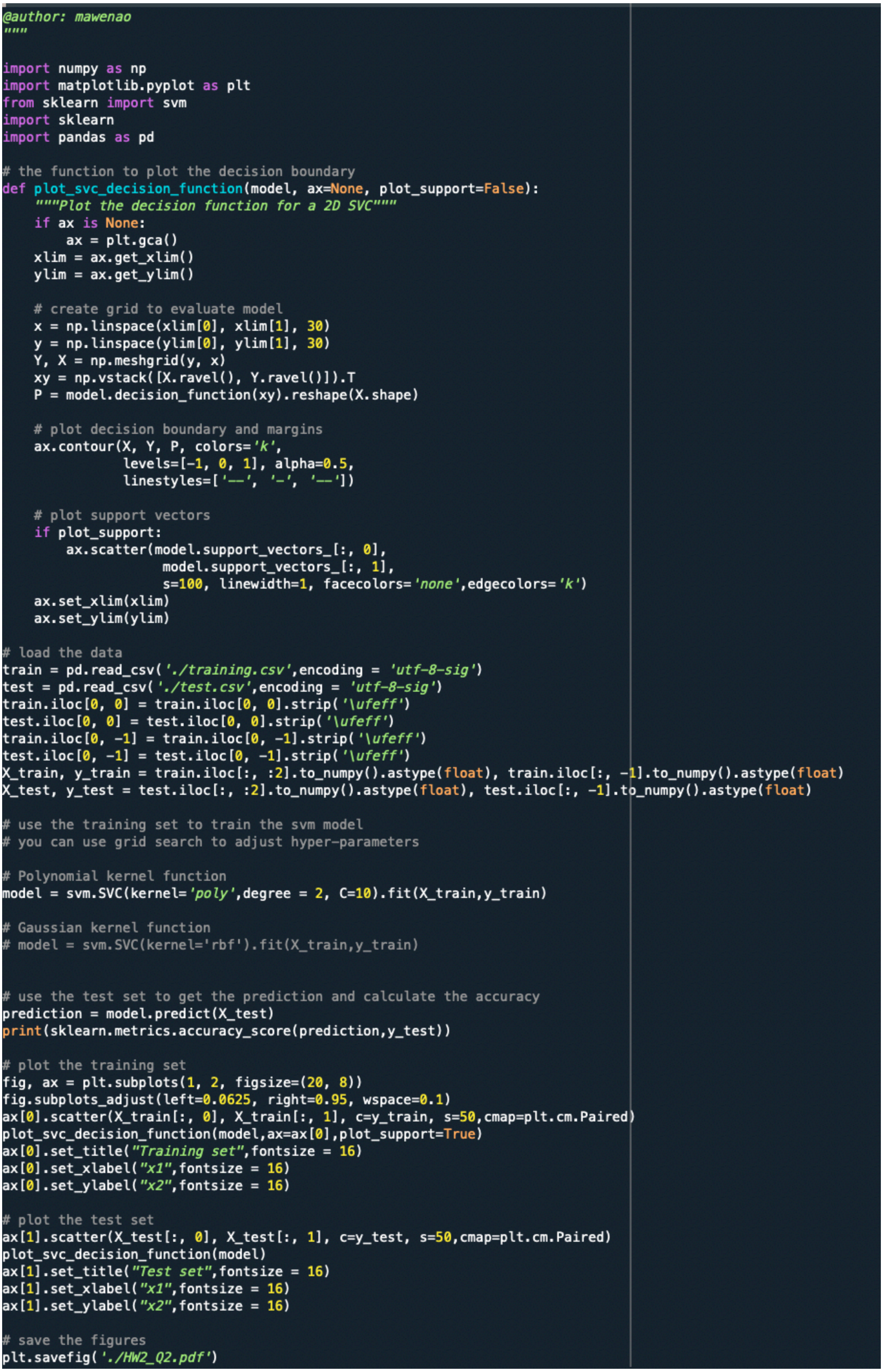
2. Use the training data attached in the blackboard to train a SVM model. Try to use what we have learned from the lecture (kernel function and soft-margin method) to optimize your model and make the accuracy of your model $\geq 86\%$ on the test set. Please try **at least two models** and each model should use different kernel function. You should give a brief introduction about your models (less than 50 words) and plot your decision boundaries with training set and test set in 2D figure (x−axis expresses x_1 while y−axis expresses x_2), respectively. Then, submit your code (Python, MATLAB or other programming languages you like). (25%)

Solution:

1) Use Polynomial kernel function with degree of 2 and set the C (soft-margin SVM) to be 10. Then the accuracy can reach to 88%.

2) Use Gaussian kernel function and set the C (soft-margin SVM) to be 22. Then the accuracy can reach to 90%.



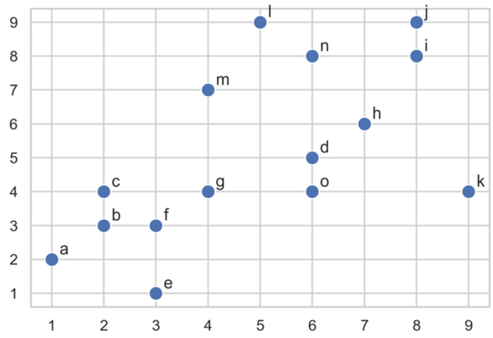


```
@author: mawenao
"""

import numpy as np
import matplotlib.pyplot as plt
from sklearn import svm
import sklearn
import pandas as pd

# the function to plot the decision boundary
def plot_svc_decision_function(model, ax=None, plot_support=False):
    """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)

# load the data
train = pd.read_csv('./training.csv',encoding = 'utf-8-sig')
test = pd.read_csv('./test.csv',encoding = 'utf-8-sig')
train.iloc[0, 0] = train.iloc[0, 0].strip('\ufeff')
test.iloc[0, 0] = test.iloc[0, 0].strip('\ufeff')
train.iloc[0, -1] = train.iloc[0, -1].strip('\ufeff')
test.iloc[0, -1] = test.iloc[0, -1].strip('\ufeff')
X_train, y_train = train.iloc[:, :2].to_numpy().astype(float), train.iloc[:, -1].to_numpy().astype(float)
X_test, y_test = test.iloc[:, :2].to_numpy().astype(float), test.iloc[:, -1].to_numpy().astype(float)

# use the training set to train the svm model
# you can use grid search to adjust hyper-parameters

# Polynomial kernel function
model = svm.SVC(kernel='poly',degree = 2, C=10).fit(X_train,y_train)

# Gaussian kernel function
# model = svm.SVC(kernel='rbf').fit(X_train,y_train)


# use the test set to get the prediction and calculate the accuracy
prediction = model.predict(X_test)
print(sklearn.metrics.accuracy_score(prediction,y_test))

# plot the training set
fig, ax = plt.subplots(1, 2, figsize=(20, 8))
fig.subplots_adjust(left=0.0625, right=0.95, wspace=0.1)
ax[0].scatter(X_train[:, 0], X_train[:, 1], c=y_train, s=50,cmap=plt.cm.Paired)
plot_svc_decision_function(model,ax=ax[0],plot_support=True)
ax[0].set_title("Training set",fontsize = 16)
ax[0].set_xlabel("x1",fontsize = 16)
ax[0].set_ylabel("x2",fontsize = 16)

# plot the test set
ax[1].scatter(X_test[:, 0], X_test[:, 1], c=y_test, s=50,cmap=plt.cm.Paired)
plot_svc_decision_function(model)
ax[1].set_title("Test set",fontsize = 16)
ax[1].set_xlabel("x1",fontsize = 16)
ax[1].set_ylabel("x2",fontsize = 16)

# save the figures
plt.savefig('./HW2_Q2.pdf')
```

Example (attached in the blackboard):

1) Introduction······ (5%)
2) Plot the figure correctly (and accuracy≥ 86%); if you has not plot the training set and test set respectively, you will be deducted 2 marks ······ (5%)
3) Build a SVM model on the given data set successfully······ (5%)
4) Use some tricks to make the accuracy≥ 86% (e.g., kernel function and soft-margin method)······ (10%)

3. Consider the following data and perform the K-means and DBSCAN algorithms using the Euclidean distance between points:



(a) Use K-means algorithm to perform three-class classification task. Assume that the initial cluster centers are (7, 4), (3, 6), (7,9) respectively. Calculate the updated cluster centers after the first iteration. (10%)

Solution:
First, we compute the distance between each sample and the initial cluster centers. For example, the distance between sample a and cluster center (3,6) is $\sqrt{20}$ while the distance between a and cluster center (7,4) is $\sqrt{40}$ and cluster center (7,9) is $\sqrt{85}$. As $\sqrt{20} < \sqrt{40} < \sqrt{85}$, the sample a should be assigned to cluster 1.

Cluster 0 (cluster center is (7, 4)): d, h, k, o
Cluster 1 (cluster center is (3, 6)): a, b, c, e, f, g, m
Cluster 2 (cluster center is (7, 9)): i, j, l, n······ (4%)

The updated cluster center of cluster 0:

$$\left(\frac{6+7+9+6}{4}, \frac{5+6+4+4}{4}\right) = (7, 4.8)$$

······(2%)

The updated cluster center of cluster 1:

$$\left(\frac{1+2+2+3+3+4+4}{7}, \frac{2+3+4+1+3+4+7}{7}\right) = (2.7, 3.4)$$

······(2%)

The updated cluster center of cluster 2:

$$\left(\frac{8+8+5+6}{4}, \frac{8+9+9+8}{4}\right) = (6.8, 8.5)$$

······(2%)

(b) Use DBSCAN to perform the clustering. Assume that $\epsilon = 2$ and *minpts*=2. List all core points, border point, noise point. (10%)

Solution:
Core points: b, c, d, f, g, n, o···(4%)
Border points: a, e, h, i, l, j···(3%)
Noise points: k, m··· (3%)

(c) Is *h* density reachable from *a*? Show the intermediate points on the chain or the point where the chain breaks of DBSCAN model. (5%)

Solution:
No. *a* is not a core point···(5%)

(d) Show the density-based clusters of DBSCAN model. (5%)

Solution:
Cluster 0: a, b, c, d, e, f, g, h, o
Cluster 1: i, l, n, j···(5%)

4. Use the distance matrix in the following table to perform the hierarchical clustering with the **group average distance** and **max distance** respectively. Show your results by listing all intermediate updated table and drawing the final dendrogram. The dendrogram should clearly show the order in which the points are merged. (20%)

	A	B	C	D	E
A	0				
B	8	0			
C	6	7	0		
D	2	3	9	0	
E	1	6	5	4	0

Solution:
Group average distance:

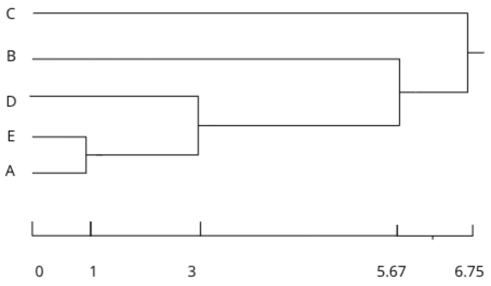
	A	B	C	D	E
A	0				
B	8	0			
C	6	7	0		
D	2	3	9	0	
E	1	6	5	4	0

	(A,E)	B	C	D
（A,E）	0			
B	7	0		
C	5.5	7	0	
D	3	3	9	0

	(A,E,D)	B	C
（A,E,D）	0		
B	5.67	0	
C	6.67	7	0

	(A,E,D,B)	C
（A,E,D,B）	0	
C	6.75	0

····························(5%)



……………………………………………………………(5%)

Max distance:

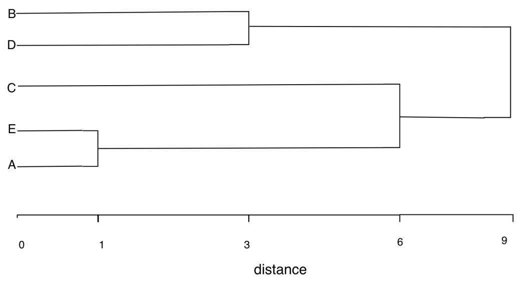
	A	B	C	D	E
A	0				
B	8	0			
C	6	5	0		
D	2	3	9	0	
E	1	7	5	4	0

	(A,E)	B	C	D
(A,E)	0			
B	8	0		
C	6	5	0	
D	4	3	9	0

	(A,E)	(B,D)	C
（A,E）	0		
(B,D)	8	0	
C	6	9	0

	(A,C,E)	(B,D)
（A,C,E）	0	
(B,D)	9	0

…………………………(5%)



……………………………………………………………(5%)

Submission:

Submit a single file named <ID>_asmt2.pdf, where <ID> is your student ID.
Your file should contain the following header. Contact Professor Dou before submitting the assignment if you have anything unclear about the guidelines on academic honesty.

```
CSCI3230 / ESTR3108 2022-23 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: <fill in your name>
Student ID  : <fill in your ID>
```

Submit your files using the Blackboard online system.

Notes:

1. Remember to submit your assignment by 23:59pm of the due date. We may not accept late submissions.
2. If you submit multiple times, **ONLY** the content and time-stamp of the **latest** one would be considered.

University Guideline for Plagiarism

Please pay attention to the university policy and regulations on honesty in academic work, and the disciplinary guidelines and procedures applicable to breaches of such policy and regulations. Details can be found at http://www.cuhk.edu.hk/policy/academichonesty/. With each assignment, students will be required to submit a statement that they are aware of these policies, regulations, guidelines and procedures.