CISC 271 Assignment 4 Selena [redacted]

Written Report

Problem 1

a)

<u>Using the k-means to perform unsupervised classification</u>

```
%load data
Yp=load("ypetal.txt");
Z2=load("zpetal.txt");
```

First, the data of y petal and z petal given were loaded in Matlab.

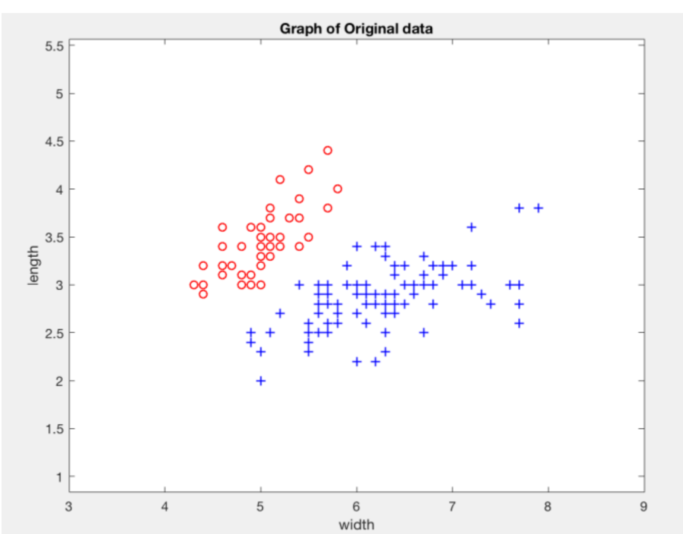
```
%K means
rng(1); % For reproducibility
k=kmeans(Z2',2);
D=[];
```

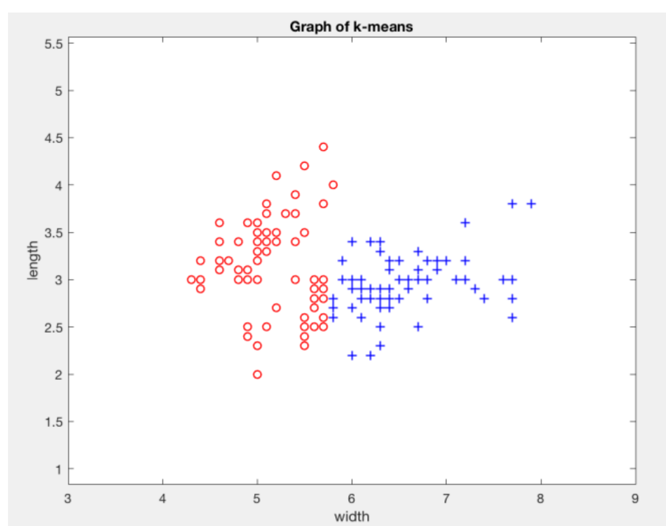
Then, the k-means was find using the kmeans function.

```
%change all 2 in k to -1, so that
%classification matches
for a=1:size(k,1)
    if k(a)==2
        k(a)=-1;
    end
end
```

Because the k-means found was classified in categories 1 and 2, I changed all the 2 to -1 to match the original classification given which was -1 and 1.

I then plotted the graph of k-means, using again the k-means function. With the length of the petals in the y axis and the width of the petals in the x axis.



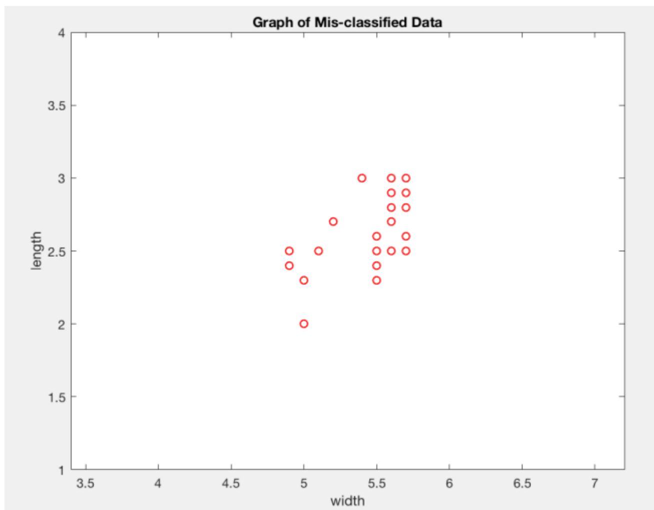


As we can see from above, when comparing the graph of the original data, to the graph of k-means, they have some differences. In the graph of k-means, the data that were originally classified in blue in the original data, are now classified in red.

```
%Check the difference
E=[]
D=k-Yp;
for b=1:size(D,1)
    if D(b)==-2
        E=[E,b];
    end
end

%Mis-classified data
E=E'
```

I then used a for loop to find the mis-classified data and plotted it as a graph.



Those are the misclassified data. These are plotted using the difference between the classification of the original data and the k-means data.

D
149x1 double

	1	2
52	0	
53	-2	
54	0	
55	-2	
56	0	
57	-2	
58	0	
59	-2	
60	-2	
61	0	
62	0	
63	0	
64	-2	
65	0	
66	-2	
67	0	

E
24x1 double

	1
1	53
2	55
3	57
4	59
5	60
6	64
7	66
8	69
9	79
10	80
11	81
12	84
13	88
14	89
15	90
16	93
17	94
18	95
19	96
20	98
21	99
22	106
23	113
24	121

The D vector is the difference between those 2. Whenever there is a -2, it shows that there was an error in the classification. The E vector is the numerical index of those mis-classified data.

b)

<u>Performing the Perceptron Algorithm</u>

To perform the Perceptron Algorithm, I first need to initialize the augmented vector.

```
%Create random 3x1 vector of values
%between 1 to 10
vvec=randi(10,3,1);
```

To do this, I used a random 3x1 vector between the numbers of 1 to 10.

```
% Score the current estimate of the weight vector
%v=randperm(3)
missed = 0;
for jx=1:n
  u=v_est'*zmat(:,jx);
  if((u>0)&((yvec(jx,1))<0))
      v_est=v_est-zmat(:,jx);
      missed = 1;
  elseif ( (u<0) & (yvec(jx,1)>0) )
      v_est=v_est+zmat(:,jx);
      missed = 1;
  %
  end
end

% Stop if the current estimate has converged
if (missed==0)
  v_final = v_est;
  break;
end
```

I then added the pseudocode of the Perception Algorithm that the Professor mentioned in the class notes in the linsepiterate function.

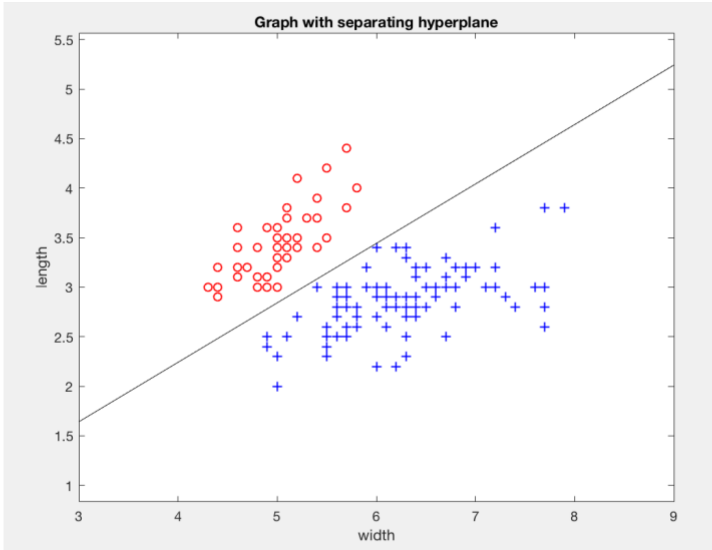
```
[v_final, i_used] = linsepiterate(vvec, Z2, Yp);
```

By calling it, I got the numerical values for the augmented vector of this hyperplane.

```
v_final =

   11.4000
  -19.0000
   -3.0000
```

I used this to plot the graph with a separating hyperplane.



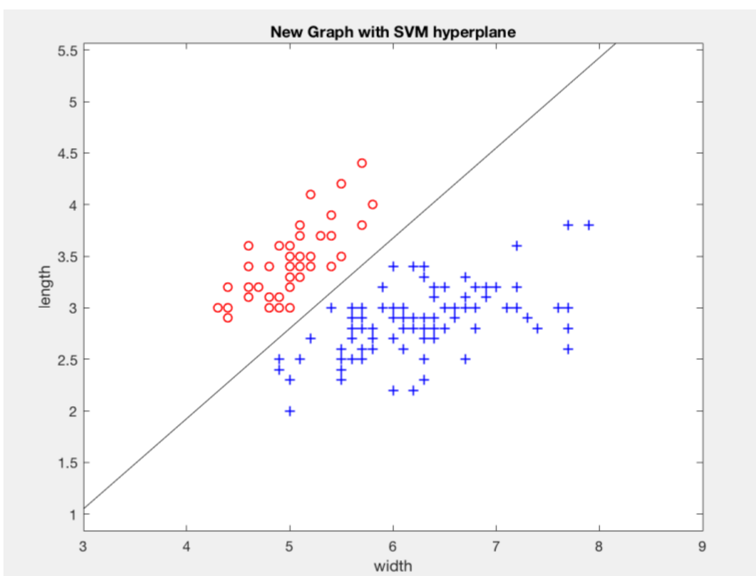
From this graph, we can see a line that separates the 2 clusters of data. However, this line does seem to be closer than the blue cluster of data than the red cluster of data, as we can see that some blue data are almost touching the line.

c)

<u>Finding the SVM</u>

```
vvec = svm271(Z2, Yp);
[v_final, i_used] = linsepiterate(vvec, Z2,Yp);
```

Using the provided svm271 function, I found the support vector machine (SVM). I used it to create another graph with a separating hyperplane for the data.



From this graph, we can see that the hyperplane is more “mid-way” between the 2 clusters. The SVM separating hyperplane is equally distant from the 2 clusters. The SVM finds a hyperplane that has the maximum margin of separation for its support vectors.
Problem 2

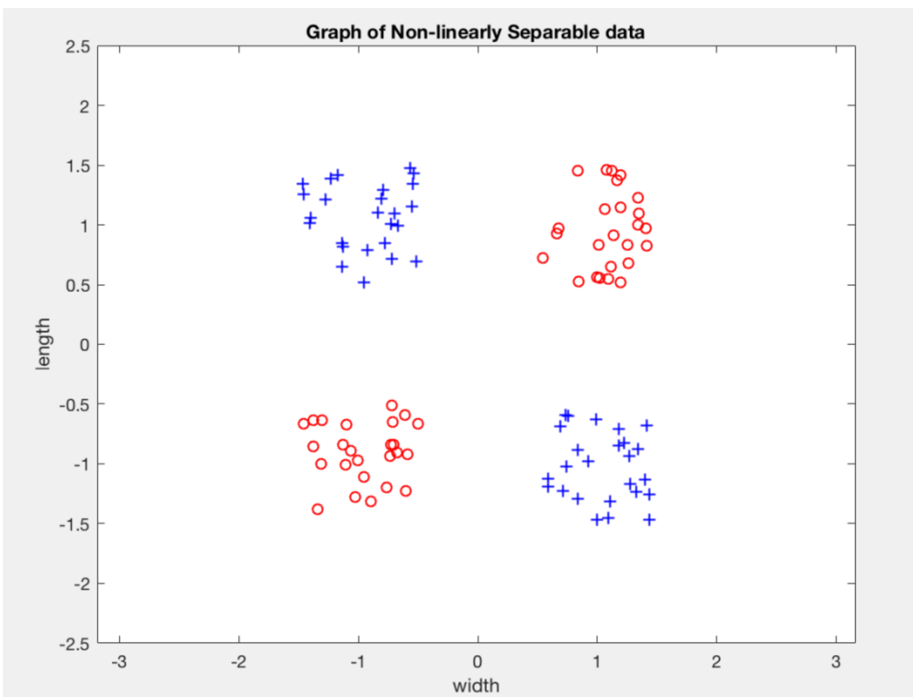
a)

<u>Embedding map into a 3D vector space</u>

In this problem, we are given some classification of data vectors that are non-linearly separable data. We would like to make them separable. One way of doing this is to embedding, which is to map each data vector to a higher-dimensional vector space.

```
%load the datas
zxor=load("zxor.txt");
yxor=load("yxor.txt");
```

We first load the new datas zxor.txt and yxor.txt in Matlab, then we plot the data.



From this graph, we can see that the data are separated into 4 clusters. We can also see that those data are not linearly separable in 2D. So, I’m going to plot them in 3D.

$$\begin{aligned} k(\vec{x}_1, \vec{x}_2) &\stackrel{\text{df}}{=} \vec{y}_1 \cdot \vec{y}_2 \\ &= x_{x1}x_{x2} + x_{y1}x_{y2} + \|\vec{x}_1\|^2\|\vec{x}_2\|^2 \\ &= \vec{x}_1 \cdot \vec{x}_2 + \|\vec{x}_1\|^2\|\vec{x}_2\|^2 \end{aligned}$$

To do this, with the given 2

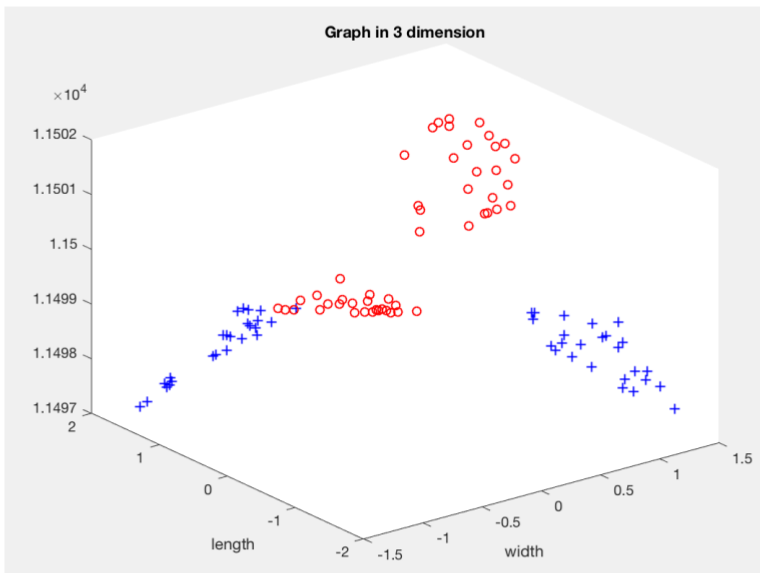
```
%Find the 3rd row of the z data
x1=zxor(1,:);
x2=zxor(2,:);
dP=x1 .* x2;
x1N= (norm(x1))^2;
x2N= (norm(x2))^2;
kernal=dP+(x1N*x2N);
```

data by using this equation.

Then, I combined the 3rd row with the previous 2 rows data to make a 3xN matrix.

```
%Combine into a 3xN vector
zxor=[zxor; kernal];
```

With these, I plotted the graph using the function plotclass3d and this gave me a graph in 3D.



This is the graph of the data given in 3d.

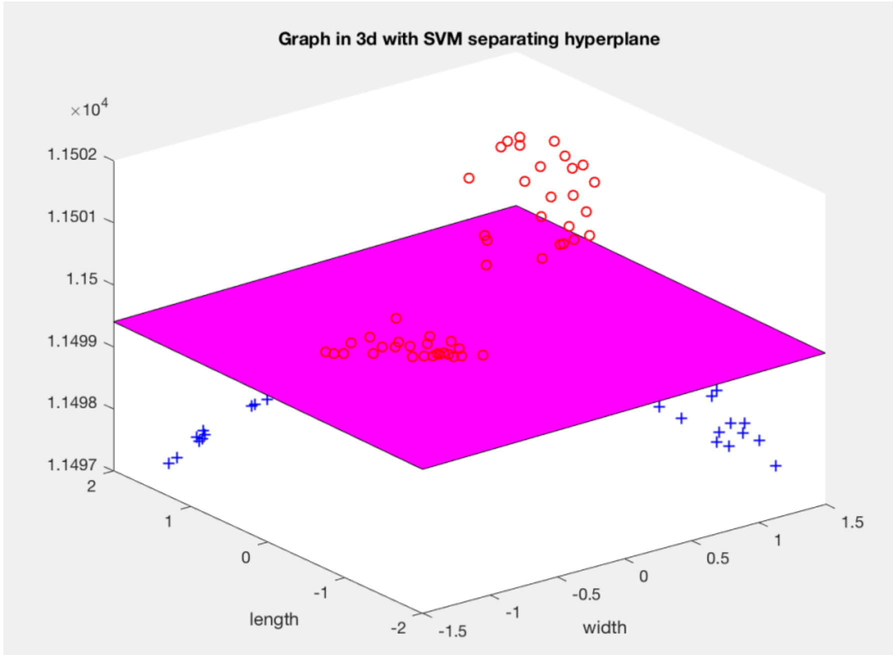
b)

<u>Find separating hyperplane for the 3D data</u>

Now that we have embedded the graph in a higher dimension, we can separate the clusters with a hyperplane.

```
%Compute a separating hyperplane
%for the 3D data
vvec = svm271(zxor, yxor);
```

I first calculated the svm for the 3D data. Then I used the function plotclass3d to plot the separating hyperplane.



This graph shows a separating hyperplane separating the data. We can see from the graph that the once non-linearly separable data in 2D can now be separated in a higher 3D dimension. If we rotate the plot, we could see more clearly that the separating hyperplane separated the red datas from the blue datas. Above the hyperplane are the red data and below are the blue datas.