ECE175 HW3 Report

Di Guan (A91041815)

Problem 1.

Suppose we have a classification problem with two classes of equal probability,

$$P_Y(0) = P_Y(1) = \frac{1}{2}$$

and class-conditional densities of the form,

$$P_{X|Y}(x|j) = K_j e^{-\dfrac{|x-a_j|}{b_j}}, j = 0, 1, b_j > 0$$

Part (a).

Determine the value of the constant Kj.

- By using the property that the integral of class-conditional densities is equal to 1, that is

$$\int_{-\infty}^{\infty} K_j e^{-\dfrac{|x-a_j|}{b_j}} dx = 1 \qquad (1)$$

- We are able to obtain the following inductions from the equation of (1)

$$\int_{-\infty}^{\infty} K_j e^{-\dfrac{|x-a_j|}{b_j}} dx = 1$$

$$\int_{-\infty}^{a_j} K_j e^{\dfrac{x-a_j}{b_j}} dx \ + \ \int_{a_j}^{\infty} K_j e^{-\dfrac{x-a_j}{b_j}} dx = 1$$

$$Let\ u = \frac{x-a_j}{b_j}, \ \ v = -\frac{x-a_j}{b_j}$$

$$\frac{du}{dx} = \frac{1}{b_j}, \ \ \frac{dv}{dx} = -\frac{1}{b_j}$$

$$dx = b_j\ du, \ \ dx = -b_j\ dv$$

$$then, \quad \int_{-\infty}^{a_j} K_j e^{\dfrac{x-a_j}{b_j}} dx = K_j \int_{-\infty}^{a_j} e^{\dfrac{x-a_j}{b_j}} dx$$

$$= K_j \int_{-\infty}^{0} b_j \; e^u du = K_j \; b_j \int_{-\infty}^{0} e^u du$$

$$= K_j \; b_j \; (e^u|_{-\infty}^{0}) = K_j \; b_j \; (e^0 - e^{-\infty}) = K_j \; b_j (1-0) = K_j \; b_j$$

$$and, \quad \int_{a_j}^{\infty} K_j e^{-\dfrac{x-a_j}{b_j}} dx = K_j \int_{a_j}^{\infty} e^{-\dfrac{x-a_j}{b_j}} dx$$

$$= K_j \int_{0}^{-\infty} -b_j \; e^v dv = -K_j \; b_j \int_{0}^{-\infty} e^v dv$$

$$= -K_j \; b_j \; (e^u|_{0}^{-\infty}) = -K_j \; b_j \; (e^{-\infty} - e^0) = -K_j \; b_j (0-1) = K_j \; b_j$$

$$thus, \quad \int_{-\infty}^{a_j} K_j e^{\dfrac{x-a_j}{b_j}} dx \; + \; \int_{a_j}^{\infty} K_j e^{-\dfrac{x-a_j}{b_j}} dx = 1 \; can \; also \; be \; writen \; as$$

$$K_j \; b_j + K_j \; b_j = 1,$$

$$therefore, \;\; K_j = \frac{1}{2b_j}$$

Part (b)

Calculate the Bayes Decision Rule for this classifier, as a function of the parameter aj and bj

- Given the Bayes Decision Rule and based on the assumption of MAP Rule,

$$i^*(x) = arg \; max_i P_{Y|X}(i|x) = arg \; max_i P_{X|Y}(x|i) P_Y(i)$$

- Therefore, we will **pick 0**,

$$if \;\; P_{X|Y}(x|0)P_Y(0) > P_{X|Y}(x|1)P_Y(1)$$

$$\frac{P_{X|Y}(x|0)}{P_{X|Y}(x|1)} > \frac{P_Y(1)}{P_Y(0)}$$

$$since \;\; P_Y(0) = P_Y(1) = \frac{1}{2},$$

$$\frac{P_{X|Y}(x|0)}{P_{X|Y}(x|1)} > 1$$

$$\frac{K_0 e^{-\frac{|x-a_0|}{b_0}}}{K_1 e^{-\frac{|x-a_1|}{b_1}}} > 1$$

$$e^{-\frac{|x-a_0|}{b_0}+\frac{|x-a_1|}{b_1}} > \frac{K_1}{K_0}$$

$$take\ log\ on\ both\ sides, \quad -\frac{|x-a_0|}{b_0}+\frac{|x-a_1|}{b_1} > ln(\frac{K_1}{K_0})$$

$$-\frac{|x-a_0|}{b_0}+\frac{|x-a_1|}{b_1} > ln(\frac{b_0}{b_1})$$

Part (c)

For the case where

$$a_0 = 0, b_0 = 1, a_1 = 1, b_1 = 2$$

what is the range of x that are satisfied with the label 0?

- Based on the results we obtained from part(b), we will **pick 0**

$$if \quad -\frac{|x-a_0|}{b_0}+\frac{|x-a_1|}{b_1} > ln(\frac{b_0}{b_1})$$
$$-|x|+\frac{|x-1|}{2} > ln(\frac{1}{2})$$
$$|x|-\frac{|x-1|}{2}+ln(\frac{1}{2}) < 0$$
$$2|x|-|x-1|+2\ ln(\frac{1}{2}) < 0$$

- we will find the range of x if x satisfies the inequality above
  - if x>1, then

$$2x-(x-1)+2\ ln(\frac{1}{2}) < 0$$
$$x < -1-2\ ln(1/2)$$
$$x < 0.3863$$

    but, it contradicts the assumption of x>1.

  - if 0<x<1, then

$$2x-(1-x)+2\ ln(\frac{1}{2}) < 0$$
$$3x < 1-2\ ln(1/2)$$
$$x < \frac{1-2\ ln(1/2)}{3}$$
$$x < 0.7954$$

    it does satisfy the assumption of 0<x<1

  - if x<0, then

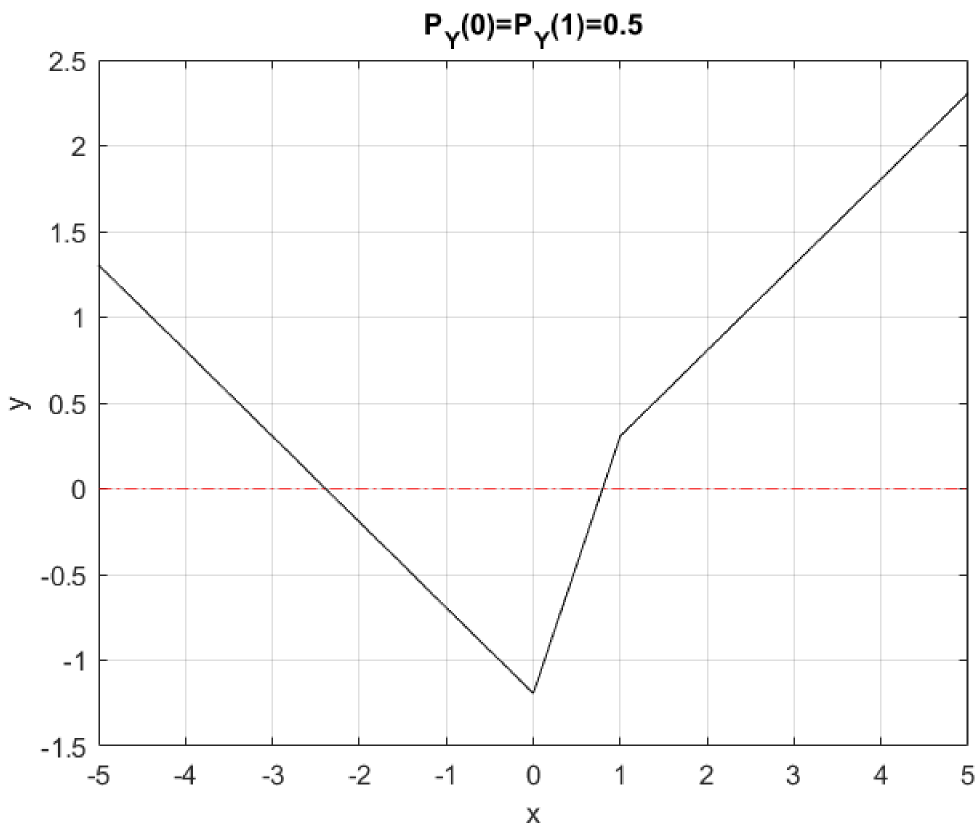
$$-2x-(1-x)+2\ ln(1/2) < 0$$
$$-x-1+2\ ln(1/2) < 0$$
$$x > -1+2\ ln(1/2)$$
$$x > -2.3863$$

    it does satisfy the assumption of x<0

- hence, **the range of x is -2.3863 < x < 0.7954 if we pick the label of 0.**
- the graph of function

$$y = 2|x| - |x-1| + 2\ ln(\frac{1}{2})$$

is shown below:



Code:

```
x=-5:1:5;
y1=abs(x)-abs(x-1)/2+log(1/2);
figure
plot(x,y1,'black')
hold on
yline(0,'-.red');
grid on
xlabel('x')
ylabel('y')
title('P_Y(0)=P_Y(1)=0.5')
```

Part (d)

repeat part (c) for the case where

$$P_Y(0) = 0.75, P_Y(1) = 0.25$$

- repeat the procedure in part (b), we will **pick 0,**

$$if \;\; P_{X|Y}(x|0)P_Y(0) > P_{X|Y}(x|1)P_Y(1)$$

$$\frac{P_{X|Y}(x|0)}{P_{X|Y}(x|1)} > \frac{P_Y(1)}{P_Y(0)}$$

$$\frac{K_0 e^{-\frac{|x-a_0|}{b_0}}}{K_1 e^{-\frac{|x-a_1|}{b_1}}} > \frac{P_Y(1)}{P_Y(0)}$$

$$e^{-\frac{|x-a_0|}{b_0}+\frac{|x-a_1|}{b_1}} > \frac{K_1\ P_Y(1)}{K_0\ P_Y(0)}$$

$$-\frac{|x-a_0|}{b_0}+\frac{|x-a_1|}{b_1} > ln(\frac{K_1\ P_Y(1)}{K_0\ P_Y(0)})$$

$$-\frac{|x-a_0|}{b_0}+\frac{|x-a_1|}{b_1} > ln(\frac{b_0\ P_Y(1)}{b_1\ P_Y(0)})$$

$$since \;\; P_Y(0) = 0.75, P_Y(1) = 0.25; \frac{P_Y(1)}{P_Y(0)} = \frac{1}{3}$$

$$and\ a_0 = 0, b_0 = 1, a_1 = 1, b_1 = 2$$

$$-|x| + \frac{|x-1|}{2} > ln((\frac{1}{2})(\frac{1}{3}))$$

$$2|x| - |x-1| + 2\ ln(\frac{1}{6}) < 0$$

- repeat the procedure of part (d), we can find the range of x that satisfied the inequality above,
  - if x>1,

    $$2x - (x-1) + 2\ ln(\frac{1}{6}) < 0$$
    $$x < 2.5835$$

  - if 0<x<1,

    $$2x - (1-x) + 2\ ln(\frac{1}{6}) < 0$$
    $$x < 1.5278$$

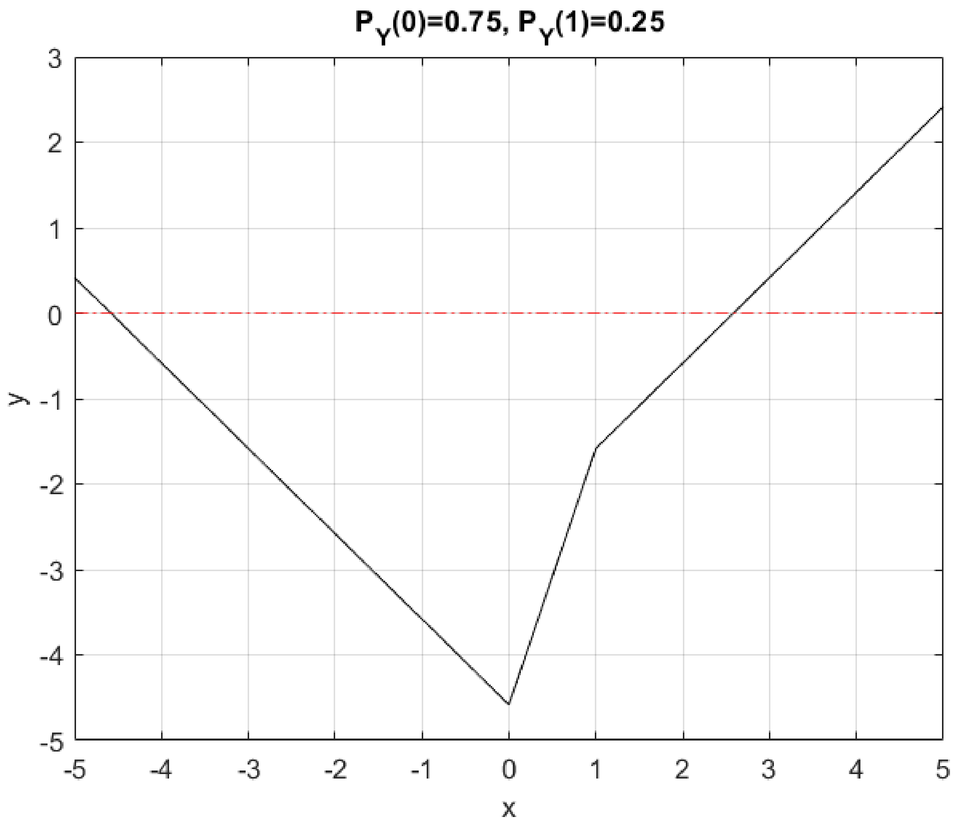
    but, it contradicts the assumption of 0<x<1
  - if x<0,

    $$-2x - (1-x) + 2\ ln(\frac{1}{6}) < 0$$
    $$x > -4.5835$$

- thus, the range of x is **-4.5835<x<2.5835** if we pick the label of 0.
- the graph of function

  $$y = 2|x| - |x-1| + 2\ ln(\frac{1}{6})$$

  is shown below:



Code:

```
%part(d)
x=-5:1:5;
y2=2*abs(x)-abs(x-1)+2*log(1/6);
figure
plot(x,y2,'black')
hold on
yline(0,'-.red');
grid on
xlabel('x')
ylabel('y')
title('P_Y(0)=0.75, P_Y(1)=0.25')
```

Problem 2.

Consider a classification problem with two Gaussian classes of identical covariance

$$\Sigma = \sigma^2 I,$$

$$P_{X|Y}(x|j) = \frac{1}{\sqrt{2\pi|\Sigma|}} e^{-\frac{1}{2\sigma^2}(x-\mu_j)^T(x-\mu_j)}$$

Part (a).

Show that, in this case, the optimal decision boundary is an hyper-plane by showing that the set of points in the boundary, i.e. the points **x** such that

$$P_{X|Y}(x|0)P_Y(0) = P_{X|Y}(x|1)P_Y(1)$$

satisfy the hyper-plane equation

$$\mathrm{w}^T(\mathrm{x}-\mathrm{x}_0)=0$$

$$Determine\ the\ values\ of\ \mathrm{w}\ and\ \mathrm{x}_0\ as\ a\ function\ of$$
$$the\ prior\ P_Y(j),\ the\ Gaussian\ means\ \mu_j\ and\ the\ variance\ \sigma^2$$

- to obtain the optimal decision boundary, we let

  $$P_{X|Y}(x|0)P_Y(0)=P_{X|Y}(x|1)P_Y(1)$$

  then, substitute the class-conditional densities,

  $$\frac{1}{\sqrt{2\pi|\Sigma|}}e^{-\dfrac{1}{2\sigma^2}(x-\mu_0)^T(x-\mu_0)}P_Y(0)=\frac{1}{\sqrt{2\pi|\Sigma|}}e^{-\dfrac{1}{2\sigma^2}(x-\mu_1)^T(x-\mu_1)}P_Y(1)$$

  $$e^{-\dfrac{1}{2\sigma^2}(x-\mu_0)^T(x-\mu_0)}P_Y(0)=e^{-\dfrac{1}{2\sigma^2}(x-\mu_1)^T(x-\mu_1)}P_Y(1)$$

  $$ln(P_Y(0))-\frac{1}{2\sigma^2}(x-\mu_0)^T(x-\mu_0)=ln(P_Y(1))-\frac{1}{2\sigma^2}(x-\mu_1)^T(x-\mu_1)$$

  $$\frac{1}{2\sigma^2}(x-\mu_1)^T(x-\mu_1)-\frac{1}{2\sigma^2}(x-\mu_0)^T(x-\mu_0)+ln(\frac{P_Y(0)}{P_Y(1)})=0$$

  $$(x-\mu_1)^T(x-\mu_1)-(x-\mu_0)^T(x-\mu_0)+2\sigma^2ln(\frac{P_Y(0)}{P_Y(1)})=0$$

  $$(x^Tx-x^T\mu_1-\mu_1^Tx+\mu_1^T\mu)-(x^Tx-x^T\mu_0-\mu_0^Tx+\mu_0^T\mu)+2\sigma^2ln(\frac{P_Y(0)}{P_Y(1)})=0$$

  $$-x^T\mu_1-\mu_1^Tx+\mu_1^T\mu+x^T\mu_0+\mu_0^Tx-\mu_0^T\mu+2\sigma^2ln(\frac{P_Y(0)}{P_Y(1)})=0$$

  $$2\mu_0^Tx-2\mu_1^Tx+||\mu_1||^2-||\mu_0||^2+2\sigma^2ln(\frac{P_Y(0)}{P_Y(1)})=0$$

  $$(\mu_0-\mu_1)^Tx-\frac{||\mu_0||^2-||\mu_1||^2}{2}+\sigma^2ln(\frac{P_Y(0)}{P_Y(1)})=0$$

  now, we obtain the form of

  $$\mathrm{w}^T\mathrm{x}+b=0$$

  $$where\ \mathrm{w}=\mu_0-\mu_1, b=-\frac{||\mu_0||^2-||\mu_1||^2}{2}+\sigma^2ln(\frac{P_Y(0)}{P_Y(1)})$$

  to obtain the form of

  $$\mathrm{w}^T(\mathrm{x}-\mathrm{x}_0)=0$$
  $$which\ is\ \mathrm{w}^T\mathrm{x}-\mathrm{w}^T\mathrm{x}_0=0$$
  $$we\ need\ let\ b=-\mathrm{w}^Tx_0,$$
  $$and\ find\ x_0=-\frac{b}{\mathrm{w}^T}$$

  thus,

$$x_0 = -\frac{-\frac{||\mu_0||^2 - ||\mu_1||^2}{2} + \sigma^2 ln(\frac{P_Y(0)}{P_Y(1)})}{(\mu_0 - \mu_1)^T}$$

$$x_0 = \frac{||\mu_0||^2 - ||\mu_1||^2 - 2\sigma^2 ln(\frac{P_Y(0)}{P_Y(1)})}{2(\mu_0 - \mu_1)^T}$$

$$x_0 = \frac{(\mu_0 - \mu_1)^T(\mu_0 + \mu_1) - 2\sigma^2 ln(\frac{P_Y(0)}{P_Y(1)})}{2(\mu_0 - \mu_1)^T}$$

$$x_0 = \frac{\mu_0 + \mu_1}{2} - \frac{\sigma^2 ln(\frac{P_Y(0)}{P_Y(1)})}{(\mu_0 - \mu_1)^T}$$

$$x_0 = \frac{\mu_0 + \mu_1}{2} - \frac{\sigma^2}{(\mu_0 - \mu_1)^T(\mu_0 - \mu_1)} ln(\frac{P_Y(0)}{P_Y(1)})(\mu_0 - \mu_1)$$

$$x_0 = \frac{\mu_0 + \mu_1}{2} - \frac{\sigma^2}{||\mu_0 - \mu_1||^2} ln(\frac{P_Y(0)}{P_Y(1)})(\mu_0 - \mu_1)$$

$$x_0 = \frac{\mu_0 + \mu_1}{2} + \frac{\sigma^2}{||\mu_0 - \mu_1||^2} ln(\frac{P_Y(0)}{P_Y(1)})(\mu_1 - \mu_0)$$

therefore, we obtain the form of

$$\mathrm{w}^T(\mathrm{x} - \mathrm{x}_0) = 0$$
$$where\ \mathrm{w} = \mu_0 - \mu_1, x_0 = \frac{\mu_0 + \mu_1}{2} + \frac{\sigma^2}{||\mu_0 - \mu_1||^2} ln(\frac{P_Y(0)}{P_Y(1)})(\mu_1 - \mu_0)$$

Part (b).

Consider the 2D case in which

$$\mu_0 = (-1, -1),\ \mu_1 = (1, 1),\ \sigma^2 = \frac{1}{4},\ P_Y(1) = 0.5.$$
$$Repeat\ for\ P_Y(1) = 0.25\ and\ P_Y(1) = 0.75$$

Using MATLAB, make a plot showing a few contours of each Gaussian, and the line corresponding to the optimal decision boundary (*note: to draw the Gaussian contours, create an x, y grid using **meshgrid()**, evaluate the values of Gaussian associated with each class at each point of this grid, multiply by the priors, add the two scaled Gaussians, and make a contour plot using* **contour()**).

- to plot the contour of the Gaussian distribution, we use the function of the multiplication of class-conditional densities and the prior probabilities defined by
$$f(x) = P_{X|Y}(x|j)P_Y(j);$$
$$where\ P_{X|Y}(x|j) = \frac{1}{\sqrt{2\pi|\Sigma|}}\ e^{-\frac{1}{2\sigma^2}(x-\mu_j)^T(x-\mu_j)}$$
- to plot the optimal boundary hyperplane, we use the equation we obtain from part (a), which is

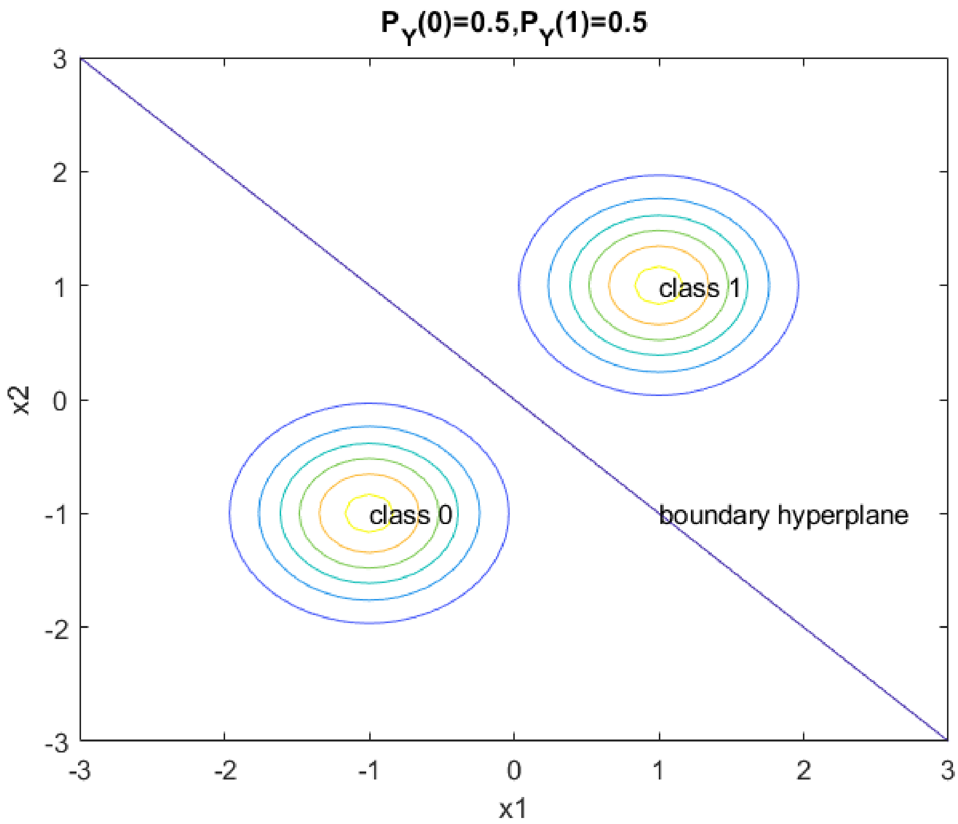
$$(\mu_0 - \mu_1)^T \mathrm{x} - \frac{||\mu_0||^2 - ||\mu_1||^2}{2} + \sigma^2 ln(\frac{P_Y(0)}{P_Y(1)}) = 0$$

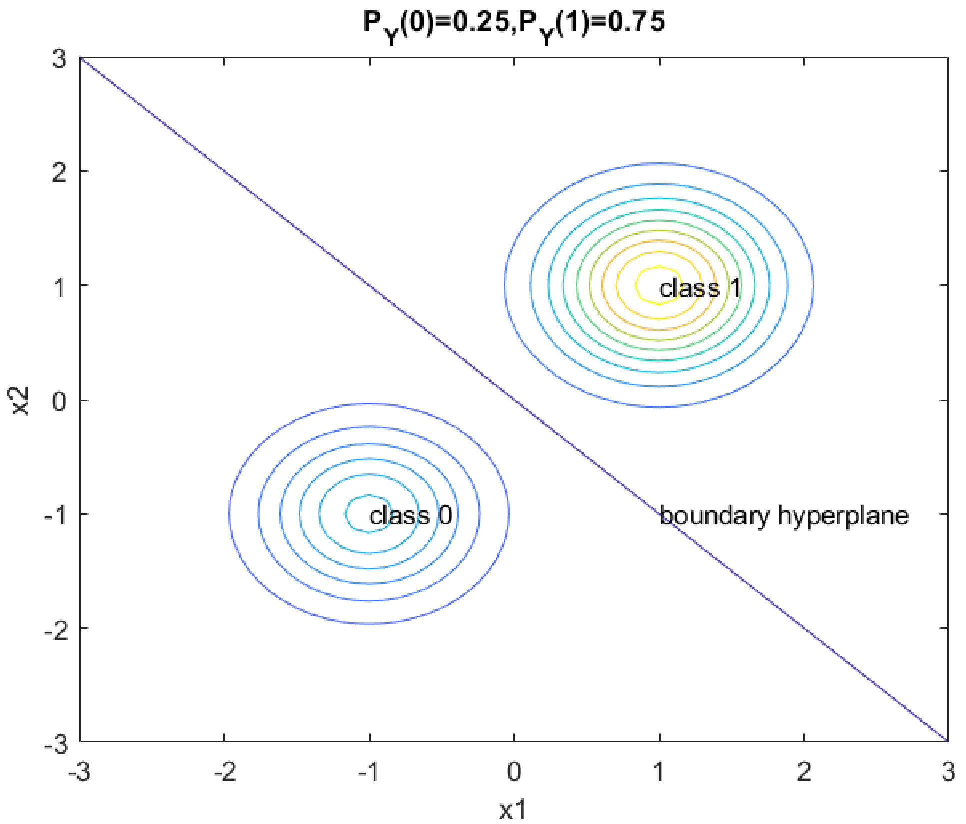
$$where\ \mu_0 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \mu_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \mathrm{x} = \begin{bmatrix} \mathrm{x}_1 \\ \mathrm{x}_2 \end{bmatrix}, \sigma^2 = \frac{1}{4}$$

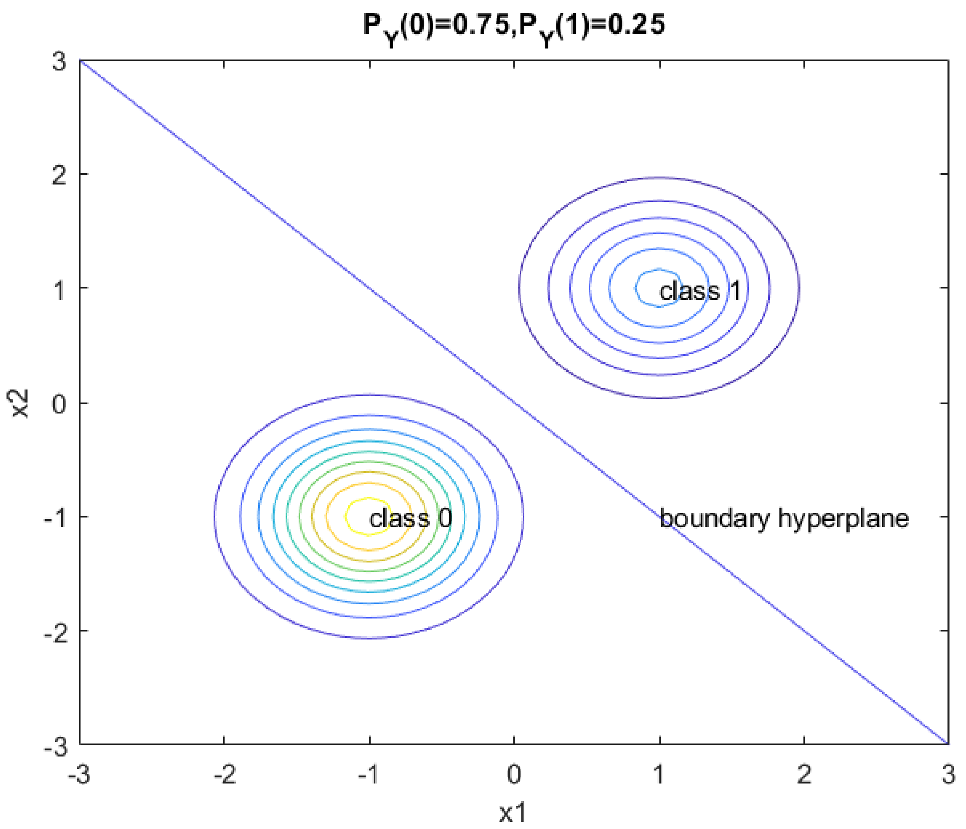
$$that\ is, [-2 \quad -2] \begin{bmatrix} \mathrm{x}_1 \\ \mathrm{x}_2 \end{bmatrix} - \frac{1-1}{2} + \frac{1}{4} ln(\frac{P_Y(0)}{P_Y(1)}) = 0$$

$$-2\mathrm{x}_1 - 2\mathrm{x}_2 + \frac{1}{4} ln(\frac{P_Y(0)}{P_Y(1)}) = 0$$

- three plots are shown below:







code:

```
% %%problem 2
% %part(b)
syms x1 x2
I=[1 0;0 1];
cov=1/4*I;
var=1/4;
det_cov=det(cov);
```

```
constant_1=1./(2*pi*0.25^2);
constant_2=-1/(2*0.25);
mu_0=[-1;-1];
mu_1=[1;1];
P0=0.75;
P1=0.25;
%gaussian distribution
z_0=P0*constant_1*exp(constant_2.*([x1;x2]-mu_0)'*([x1;x2]-mu_0));
z_1=P1*constant_1*exp(constant_2.*([x1;x2]-mu_1)'*([x1;x2]-mu_1));

[x1,x2]=meshgrid(-3:0.1:3);
z_0=eval(z_0);
z_1=eval(z_1);
%optimal boundary hyperplane
[fx1,fx2]=meshgrid(-3:0.1:3);
boundary=-2*fx1-2*fx2+var*log(P0/P1);

figure
contour(x1,x2,z_0);
text(-1,-1,'class 0');
hold on
contour(x1,x2,z_1);
text(1,1,'class 1');
hold on
contour(x1,x2,boundary,1);
text(1,-1,'boundary hyperplane')
xlabel('x1');
ylabel('x2');
title(['P_Y(0)=',num2str(P0),',P_Y(1)=',num2str(P1)]);
```

Part (c).

geometric interpretation to the vector w and the point x0

The vector **w** is orthogonal to the optimal boundary line. The point **x0** is the point of intersecting the optimal boundary line and the line connecting the vectors of μ0 and μ1. If the prior probability for each class is changing, the vector **w** does not change and the point **x0** changes. If prior probability of class 0 is greater than the one of class 1, then the point **x0** is moving toward to μ1(or moving away from μ0) and due to the movement of point **x0**, the optimal decision boundary moves in the direction that is away from μ0 and toward to μ1. Otherwise, x0 and optimal boundary line will perform in an opposite way.

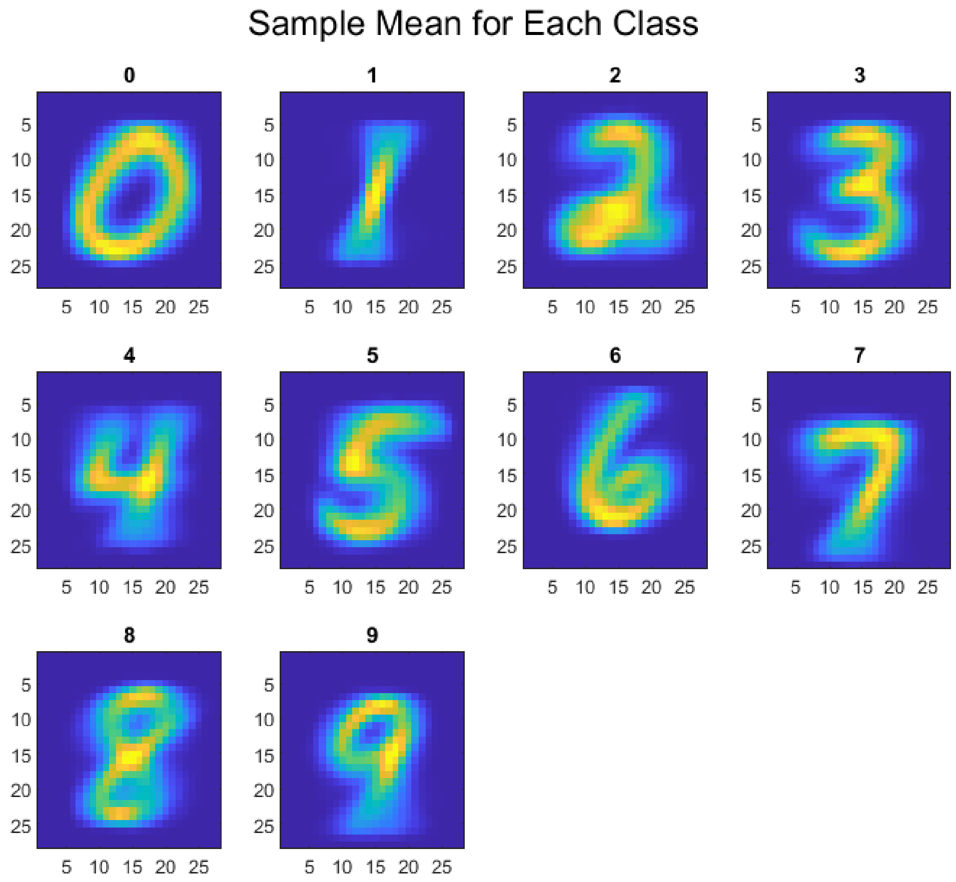
Problem 3.

We assume the class-conditional densities have a multivariate Gaussian distribution of 28 × 28 = 784 dimensions.

$$P_{X|Y}(x|j) = \frac{1}{\sqrt{2\pi|\Sigma|}} e^{-\frac{1}{2\sigma^2}(x-\mu_j)^T\Sigma^{-1}(x-\mu_j)}$$

Moreover, we assume that the covariance matrix for every class is the same and equal to identity covariance, i.e. Σi = Σ = I, ∀ i and also that the prior for the classes are distributed uniformly, i.e PY (i) = 1/N = constant where N is the number of classes.

1. For each digit, calculate and display the sample mean



code:

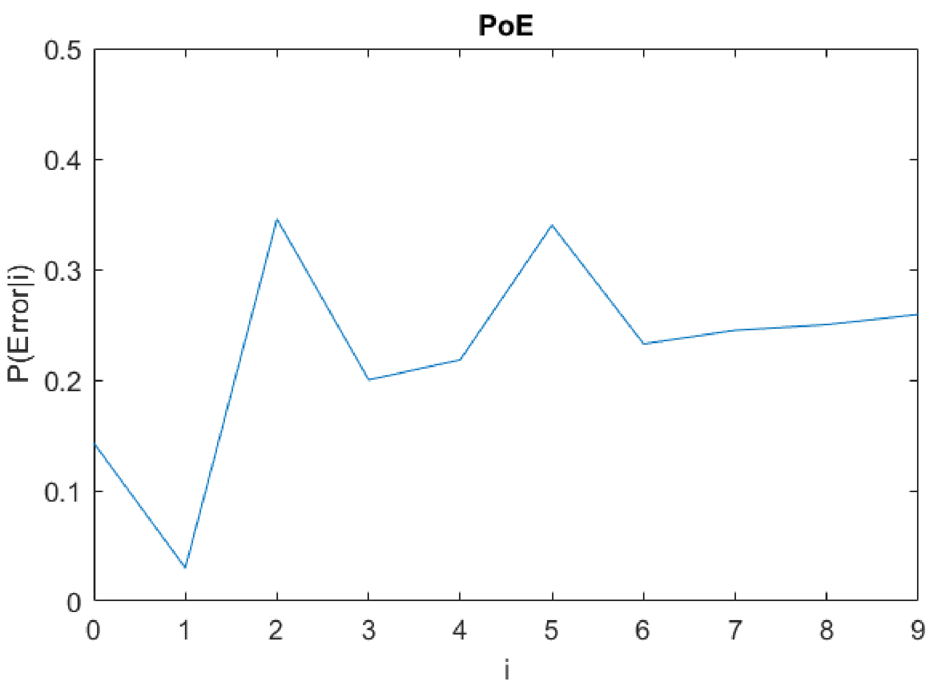
```
%%%problem 3
%%%(1).
data=load('E:\UC Social Dead\2020Winter\ECE175\Homework\hw3\HW3_Data\data.mat');
label=load('E:\UC Social
Dead\2020Winter\ECE175\Homework\hw3\HW3_Data\label.mat');
imageTest=data.imageTest;
imageTrain=data.imageTrain;
labelTest=label.labelTest;
labelTrain=label.labelTrain;

%initialization
all_sample_mean=zeros(784,10);
stat_test=tabulate(labelTest);

%calculate and display the sample mean
figure
for i=0:1:9
    %find the location where labelTrain is equal to the class i,
    loc=find(labelTrain==i);
    %restore the images of the same class into variable class_i
    class_i = imageTrain(:,:,loc);
    %reshape the image dimension from 28*28*size(loc) to 784*size(loc)
    class_i=reshape(class_i,[28*28,size(loc)]);
    %obtain the sample mean for each class
    sample_image=sum(class_i,2)./size(loc,1);
    %restore each class to all_sample_mean with dimension of 784*10
```

```
        all_sample_mean(:,i+1)=sample_image;
        %reshape it back to the dimension of 28*28 for each image
        sample_image_original=reshape(sample_image,[28,28]);
        %plot the sample mean for each class
        subplot(3,4,i+1)
        imagesc(sample_image_original)
        title(i)
    end
    sgtitle('Sample Mean for Each Class')
```

2. Using sample mean as an estimate of the class mean, perform the task of classification by Bayes Decision Rule.

- **part (a)**: The error rate for each class & the plot of PoE

Class	PoE	# error images	# images
0	0.1429	6	42
1	0.0299	2	67
2	0.3455	19	55
3	0.2	9	45
4	0.2182	12	55
5	0.34	17	50
6	0.2326	10	43
7	0.2449	12	49
8	0.25	10	40
9	0.2593	14	54



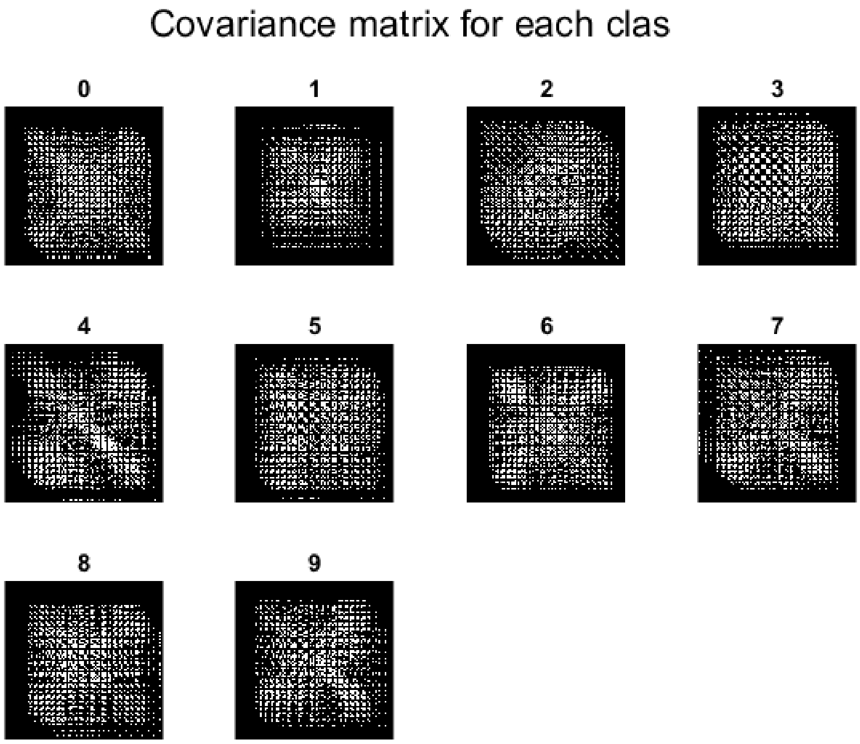
- **part (b)**: The total error rate

$$Total\ Error\ Rate = \frac{111}{500} = 0.222$$

code:

```
%%%(2).perform the classification by BDR
%part(a)
%initialization
i_x=zeros(1,10);
predict_label=zeros(500,1);
error_class=zeros(10,1);

%calculate i_x for each testing image from 1 to 500 wrt those 10 classes, find
%the maximum i_x within those 10 classes and assign the class value and
%restore the value to the repository predict_label
for i=1:500
    for j=0:9
        %reshape each image in testing image to the size of 784*1
        image=reshape(imageTest(:,:,i), [28*28,1]);
        %each class mean from class 0 to 9
        class_mean=all_sample_mean(:,j+1);
        %calculate the value of i_x, each tesing image wrt 10 class sample
        %mean and restore to variable i_x
        i_x(1,j+1)=(image-class_mean)'*(image-class_mean);
        %find the minimum value and location within i_x for each testImage
        [val,loc]=min(i_x);
        %restore the value of class to variable predict_label(500*1)
        predict_label(i,1)=loc-1;
    end
    %calculate the error
    if labelTest(i) ~= predict_label(i)
        error_class(labelTest(i)+1)=error_class(labelTest(i)+1)+1;
    end
end
%calculate error rate for each class
error_rate_class=error_class./stat_test(:,2);

%plot error rate for each class
figure
x=(0:9);
plot(x,error_rate_class);
ylim([0,0.5])
xlabel('i')
ylabel('P(Error|i)')
title('PoE')

%part(b)
count=500-sum(labelTest==predict_label);
total_error_rate=count/500;
```

3. For different digits, calculate and display the covariance matrix. Why were we not able to use this matrix in our distance computations?



this is partly because the dimension of the covariance matrix for each class is too large to calculate the distance.

code:

```
%%%%(3).
imageTrain=reshape(imageTrain,784,5000);

figure
for i=0:9
    a=find(labelTrain==i)';
    b=imageTrain(:,a);
    b=b';
    c=cov(b);
    subplot(3,4,i+1)
    imshow(c)
    title(i)
end
sgtitle('Covariance matrix for each clas')
```