Assignment 1

Thursday, February 6, 2020 8:18 PM

Question 1 :

Part 1 , Sub - Question 1

Answer 1 .)

As we know from likelihood ratio for a general case that , w1 : class label 1 , w2 : class label 2 , x : feature , $\lambda : loss\ variable$.

$$P(w1 \mid x\ ) \,/\, P(w2 \mid x) > \frac{(\ \lambda 12\ - \lambda 22\ )\ \ P(w2)}{(\ \lambda 21\ -\ \lambda 11\ )\ \ P(w1)} == threshold \quad \ldots . (1)$$

For above question , w1 : L=0 while , w2 : L=1
And we would assume that it is a zero-one loss , where we would
Get a loss matrix as below .

$$\lambda = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad \ldots \ldots (2)$$

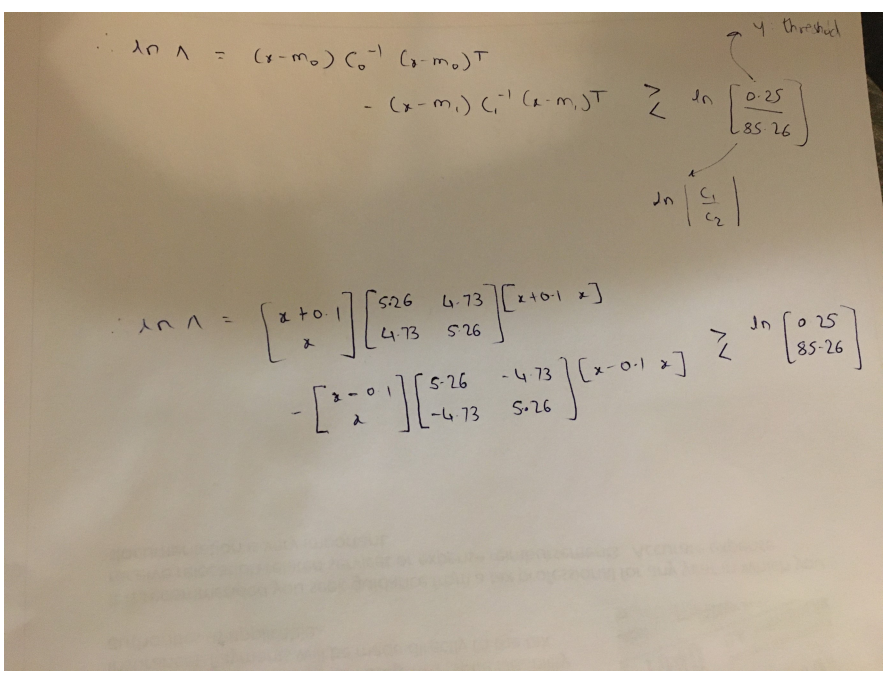
From equation (1) and (2) , we have

$$P(x \mid w1\ ) \,/\, P(x \mid w2) > \frac{P(w2)}{P(w1)}$$

As per the class prior's in question , P(w1) = 0.8 and P(w2) = 0.2

=> $P(x \mid w1\ ) \,/\, P(x \mid w2) > 0.25$.

Since both the distributions are gaussians , thus we can plug-in the distribution equations of them , take a logarithm on both sides and solve the equations (on paper , posting the pictures below) :



Part 1 , Sub - Question 2

Answer 2.)

Documented and commented source code :
https://github.com/vikas753/eece5644/blob/master/hw1/question1part1.m

We have generated two class labels with an approximate priority of 0.8 for **label 0** and 0.2 for **label 1 .** Logic is to generate 10000 random numbers and randomnly distribute them an approximate ration of 0.8:0.2 by comparison against 0.2 and then recalculate the number of samples in both classes .

Once the numbers are recalculated in both class labels , a gaussian PDF plot with a likelihood parameters (mean , variance) (in this case it is mentioned in question) is devised for both classes and the plot looks like below :

Code reference :

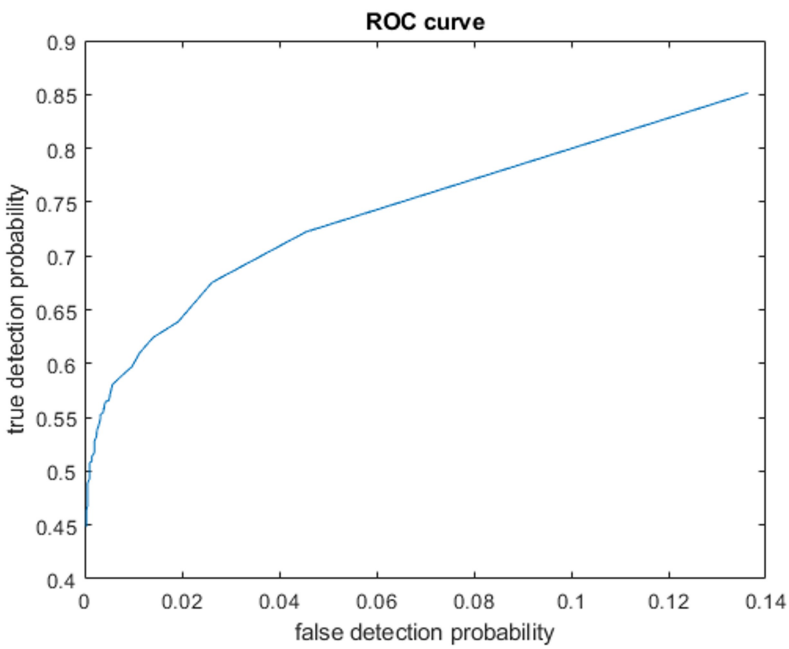
```
% number of iid samples for the data distribution
N = 10000;

mu(:,1) = [-0.1;0]; mu(:,2) = [0.1;0];
Sigma(:,:,1) = [1 -0.9;-0.9 1]; Sigma(:,:,2) = [1 0.9;0.9 1];

% class priors for labels 0 and 1 respectively
p = [0.8,0.2];
label = rand(1,N) >= p(1);
Nc = [length(find(label==0)),length(find(label==1))]; % number of samples from each class
x = zeros(n,N); % save up space
% Draw samples from each class pdf
for l = 0:1
   x(:,label==l) = mvnrnd(mu(:,l+1),Sigma(:,:,l+1),Nc(l+1))';
end
```

There is narrow but thick region of coincidence between the PDFs which usually contribute a lot to false as well as true probability rates .

We then perform a likelihood ratio test on above combined PDF data by varying threshold from 1 to 100 in steps of 1 to study the sensitivity and specifity of the data , which is also known as the ROC curve . ROC plot is as below :

Code reference :

```
for iterator = 1:1:10000
  gamma = iterator;

  discriminantScore = log(evalGaussian(x,mu(:,2),Sigma(:,:,2)))-log(evalGaussian(x,mu(:,1),Sigma(:,:,1)));
  decision = (discriminantScore >= log(gamma));

  % probability of true negative
  ind00 = find(decision==0 & label==0);
  p00 = length(ind00)/Nc(1);

  % probability of false positive
  ind10 = find(decision==1 & label==0);
  p10 = length(ind10)/Nc(1);
  falsePositiveRateArray = [p10;falsePositiveRateArray];

  % probability of false negative
  ind01 = find(decision==0 & label==1);
  p01 = length(ind01)/Nc(2);


  % probability of true positive
  ind11 = find(decision==1 & label==1);
  p11 = length(ind11)/Nc(2);
  truePositiveRateArray = [p11;truePositiveRateArray];

  % Probability of error would be sum of all incorrect probabilities
  perror = p10 + p01;
  if perror < minPerror
    minPerror = perror;
    BestThreshold = iterator;
    rocMarkerTruePositive = p11;
    rocMarkerfalsePositive = p10;
  end

end
```



Part 1 , Sub - Question 3

Please find the Probability error characteristics for this data distribution as below :

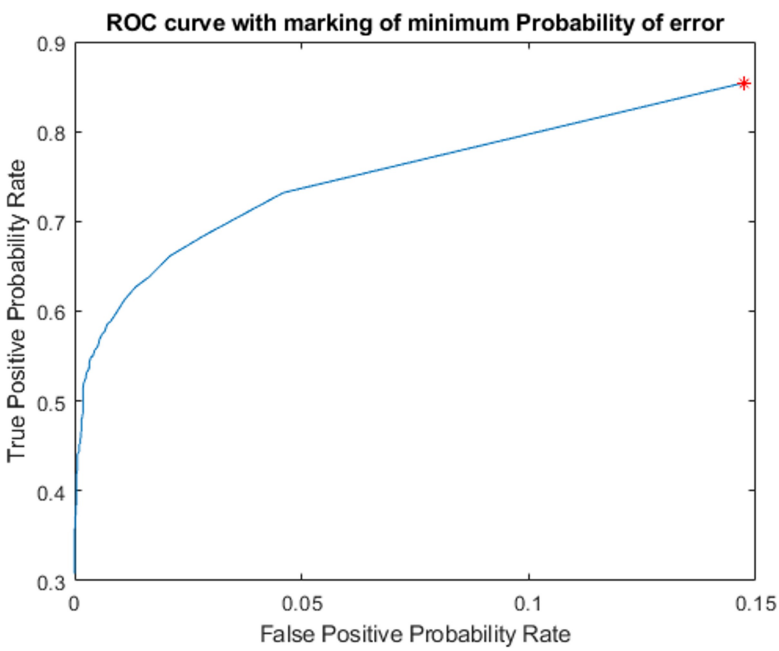
Source code : https://github.com/vikas753/eece5644/blob/master/hw1/question1part1.m

Algorithm :

Vary the threshold from 1 to 1000
Compute the likelihood ratio for the combined data distribution
Compare it with threshold and make a decision for all the values in data set .
Compute the probabilities : true positive , true negative , false positive , false negative
Compute error probability : true negative + false positive
Check the same against a previous error probability if it is less than the same then update the minimum probability of error

Characteristics	Value
Min P-error	**0.2862**
Threshold	**1**
ROC Marker - True Positive	**0.8597**
ROC Marker - False Positive	**0.1459**

And it's operating characteristic marking on the ROC curve is as below :



< Avoiding the mathematical computation on paper as it would certainly be a very long equation to represent and we don't know the boundary clearly as well >

Part 2 , Sub - Question 1

Answer 1 .)

As we know from likelihood ratio for a general case that , w1 : class label 1 , w2 : class label 2 , x : feature , $\lambda : loss\ variable$.

$$P(w1 \mid x\ ) \ / \ P(w2 \mid x) > \frac{(\ \lambda 12\ - \lambda 22\ )\ \ P(w2)}{(\ \lambda 21\ -\ \lambda 11\ )\ \ P(w1)} == threshold \quad \ldots.(1)$$

For above question , w1 : L=0 while , w2 : L=1

And we would assume that it is a zero-one loss , where we would
Get a loss matrix as below .

$$\lambda = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad \dots\dots (2)$$

From equation (1) and (2) , we have

$$P(x \mid w1\,) \,/\, P(x \mid w2) > \frac{P(w2)}{P(w1)}$$

As per the class prior's in question , P(w1) = 0.8 and P(w2) = 0.2

=> $P(x \mid w1\,) \,/\, P(x \mid w2) > 0.25$.

Since both the distributions are gaussians , thus we can plug-in the distribution equations of them , take a logarithm on both sides and solve the equations (on paper , posting the pictures below) :

$$\Lambda = \frac{\frac{1}{\sqrt{2\pi}\,|C_1|^2}\, e^{-(x-m_0)C_0^{-1}(x-m_0)^T}}{e^{-(x-m_1)C_1^{-1}(x-m_1)^T}} \gtrless$$

As we know that $C_0 = C_1 = I$ [as per the incorrect assumption in question]

$$\therefore \Lambda = \frac{e^{-(x-m_0)C_0^{-1}(x-m_0)^T}}{e^{-(x-m_1)C_1^{-1}(x-m_1)^T}} \gtrless y$$

$$\therefore \ln \Lambda = (x-m_1)C_1^{-1}(x-m_1)^T - (x-m_0)C_0^{-1}(x-m_0)^T \gtrless \ln y$$

$$\therefore \ln \Lambda = I^{-1}\left[(x-m_1)(x-m_1)^T - (x-m_0)(x-m_0)^T\right] \gtrless \ln y$$

$$\therefore \ln \Lambda = \begin{bmatrix} x-0.1 \\ x \end{bmatrix}\left[x-0.1 \;\; x\right] - \begin{bmatrix} x+0.1 \\ x \end{bmatrix}\left(x+0.1 \;\; x\right) \gtrless \ln y$$

$$\therefore \ln \Lambda = (x-0.1)^2 + x^2 - \left[(x+0.1)^2 + x^2\right] \gtrless \ln y$$

$$\therefore \ln \Lambda = 0.2\,(2x) \gtrless \ln y$$

$$\boxed{\therefore \ln \Lambda = -0.4x \gtrless \ln y}$$

Likelihood function reduces itself to be a linear classifier for an assumption that variance matrix is an identity .

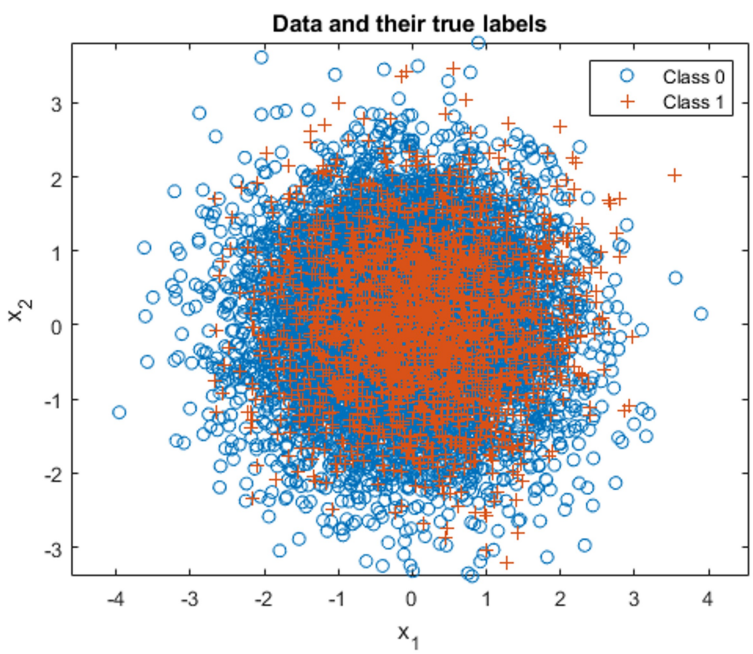
Part 2 , Sub - Question 2

Source code : https://github.com/vikas753/eece5644/blob/master/hw1/question1part2.m

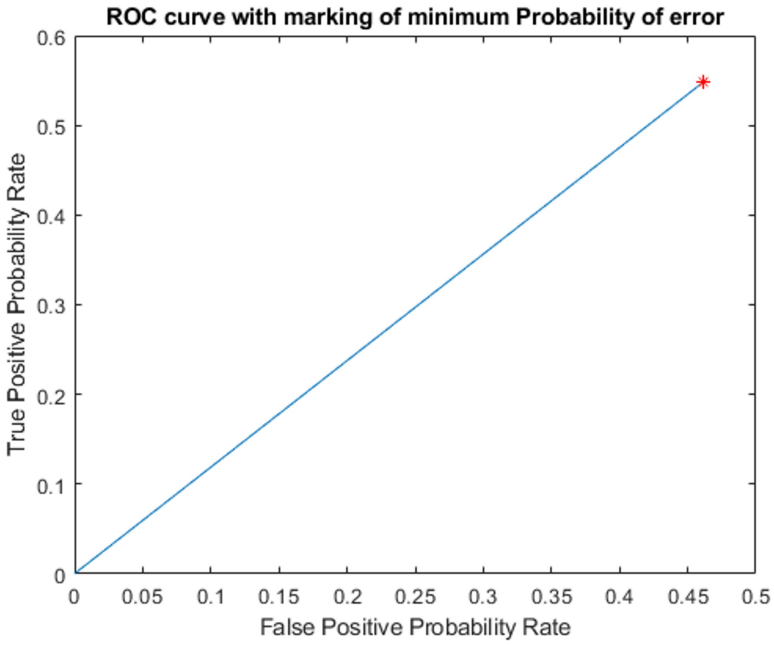
We have generated two class labels with an approximate priority of 0.8 for **label 0** and 0.2 for **label 1 .** Logic is to generate 10000 random numbers and randomnly distribute them an approximate ration of 0.8:0.2 by comparison against 0.2 and then recalculate the number of samples in both classes .
Once the numbers are recalculated in both class labels , a gaussian PDF plot with a likelihood parameters (mean , variance) (in this case it is mentioned in question) is devised for both classes and the plot looks like below .
Now the catch is , variance modified has been modified to be **Identity matrices** , as the result of the same both datasets are coinciding which could lead to more error's in probability and a bad operating characteristics .

Code reference :

```
mu(:,1) = [-0.1;0]; mu(:,2) = [0.1;0];

% Initialize and assume the variances to be identity matrices
Sigma(:,:,1) = [1 0;0 1]; Sigma(:,:,2) = [1 0;0 1];

% class priors for labels 0 and 1 respectively
p = [0.8,0.2];
label = rand(1,N) >= p(1);
Nc = [length(find(label==0)),length(find(label==1))]; % number of samples from each class
x = zeros(n,N); % save up space
% Draw samples from each class pdf
for l = 0:1
   x(:,label==l) = mvnrnd(mu(:,l+1),Sigma(:,:,l+1),Nc(l+1))';
end
```



ROC curve (as expected) is linear , since the likelihood ratio test in previous solution gave a linear classifier or mathematical equation . A linear classifier would be more inclined to have a linear ROC . It would eventually have a worst performance , as it certainly doesn't perform any prediction .



Part 2 , Sub - Question 3 : (algorithm same as that of part 1 sub-question 3)

Source code : https://github.com/vikas753/eece5644/blob/master/hw1/question1part2.m

Code reference :

```
for iterator = 1:1:10000

  gamma = iterator;
  disp(gamma);

  discriminantScore = log(evalGaussian(x,mu(:,2),Sigma(:,:,2)))-log(evalGaussian(x,mu(:,1),Sigma(:,:,1)));
  decision = (discriminantScore >= log(gamma));

  % probability of true negative
  ind00 = find(decision==0 & label==0);
  p00 = length(ind00)/Nc(1);

  % probability of false positive
  ind10 = find(decision==1 & label==0);
  p10 = length(ind10)/Nc(1);
  falsePositiveRateArray = [p10;falsePositiveRateArray];

  % probability of false negative
  ind01 = find(decision==0 & label==1);
  p01 = length(ind01)/Nc(2);


  % probability of true positive
  ind11 = find(decision==1 & label==1);
  p11 = length(ind11)/Nc(2);
  truePositiveRateArray = [p11;truePositiveRateArray];

  % Probability of error would be sum of all incorrect probabilities
  perror = p10 + p01;
  if perror < minPerror
    minPerror = perror;
    BestThreshold = iterator;
    rocMarkerTruePositive = p11;
    rocMarkerfalsePositive = p10;
  end

end
```

Characteristics	Value
Min P-error	**0.9132**
Threshold	**1**
ROC Marker - True Positive	**0.5485**
ROC Marker - False Positive	**0.4616**

Please find the mathematical computation of a rough estimate of error probability as below :

Average probability of error would be

Label = 1, for feature, $x = 0$ .. $p10$

Label = 0, for feature, $x = 1$ .. $p01$

Note

$$p10 = \int_{-\infty}^{0} p(x \mid L=1)\,dx$$

$$p01 = \int_{0}^{\infty} p(x \mid L=0)\,dx$$

$$p\,Error = p10 + p01$$

$$\therefore p\,Error = \int_{-\infty}^{0} \frac{1}{\sqrt{2\pi}} e^{(x-m_1)(x-m_1)^T} + \int_{0}^{\infty} \frac{1}{\sqrt{2\pi}} e^{(x-m_0)(x-m_0)^T}$$

$$\therefore p\,Error = \frac{1}{\sqrt{2\pi}} \left[\int^{\infty} \left(e^{0.01 - 0.2x + 2x^2} - e^{0.01 + 0.2x + 2x^2} \right) dx \right]$$

$$\therefore p\,Error = \frac{e^{0.01}}{\sqrt{2\pi}} \left[\int_{0}^{\infty} e^{2x^2} \left[e^{-0.2x} - e^{0.2x} \right] dx \right]$$

Approximating above equation by assuming that >1, integral is zero and using sample counting

$$\boxed{p\,Error \approx 0.92}$$

Comments on both classifiers :

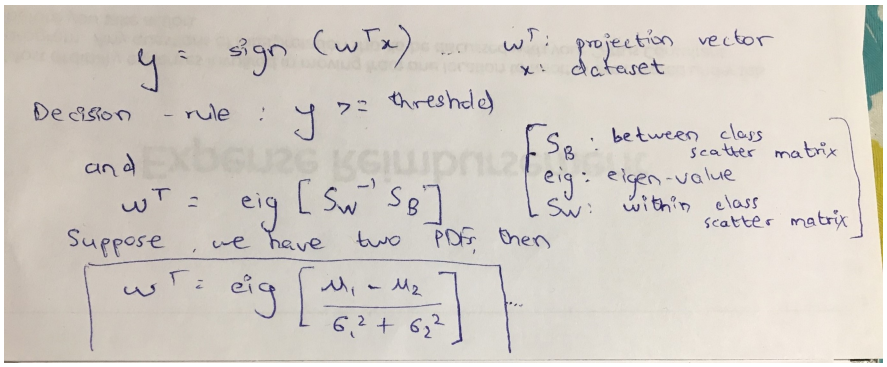
For the dataset whose label class conditional PDFs have same mean but different variances not equal to an identity matrix chose in such a manner that there is a thick region of intersection has a better performance when it comes to using probability as a threshold . While having variances same as that of an identity matrix yields a linear classifier that has a poor performance as evident the characteristics above , with a high Probability of error .

Part 3 : In the third part of this exercise, repeat the same steps as in the previous two cases for the Fisher Linear Discriminant Analysis based classifier. Using the 10000 available samples, with sample average based estimates for mean and covariance matrix for each class,

determine the Fisher LDA projection vector wLDA. For the classification rule wT LDAx compared to a threshold τ, which takes values from −∞ to ∞, trace the ROC curve, identify the threshold at which the probability of error (based on sample count estimates) is minimized, and clearly mark that operating point on the ROC curve.
Note: In order for us to have a uniform solution across all submissions, When finding the Fisher LDA projection matrix, do not be concerned about the difference in the class priors. When determining the between-class and within-class scatter matrices, use equal weights for the class means and covariances, like we did in class.

Source Code : https://github.com/vikas753/eece5644/blob/master/hw1/question1part3.m

Please find a short derivate of the Fischer LDA decision rule and projections matrix equations as shown in below snip .
(**Assumptions : both classes have equal weights for accounting in-between and within class scatter matrices as given in question)**



Code reference :

```
% Calculate the between/within-class scatter matrices
% assuming equal weights for in and between class scatter matrices
Sb = (meanClass1-meanClass2)*(meanClass1-meanClass2)';
Sw = varianceClass1 + varianceClass2;

% Solve for the Fisher LDA projection vector (in w)
[V,D] = eig(inv(Sw)*Sb);
[~,ind] = sort(diag(D),'descend');
w = V(:,ind(1)); % Fisher LDA projection vector
```

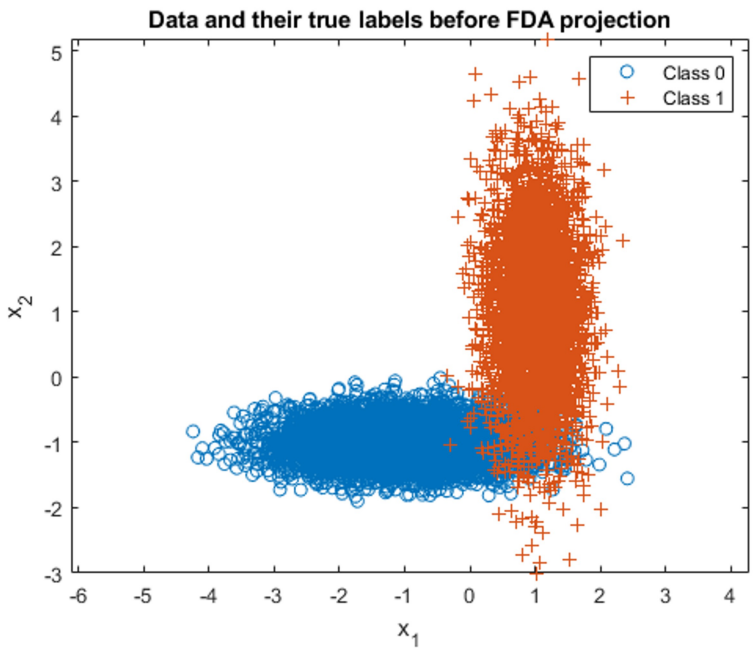
For this exercise , algorithm or pseudocode goes as follows :
1 . Generate two normal \ gaussian PDFs one of 6000 samples and other of 4000 samples
2 . Compute their means , variances , within and between class scatter matrix .
3 . Compute the projection matrix : ratio of within and between class scatter matrix
4 . Transform the dataset using projection matrix onto projection line which would be an unit vector and flip the sign based on delta of means between class 0 and 1 .
5 . Vary the threshold from -100 to 100 and compare the projected matrix against the threshold to classify them to be data in label 1 or 0 .
Based on the same , compute :
P(D=1|L=1) : True Positive Probability and P(D=1|L=0) : False Positive Probability for each threshold and make an array .
Minimum Probability of the classifier and threshold , true positive probability and false positive probability where the probability of error is lowest .

Please find the dataset as below :

Code reference :

```
figure(2), clf,
plot(class1Data(1,:),class1Data(2,:),'o'), hold on,
plot(class2Data(1,:),class2Data(2,:),'+'), axis equal,
```

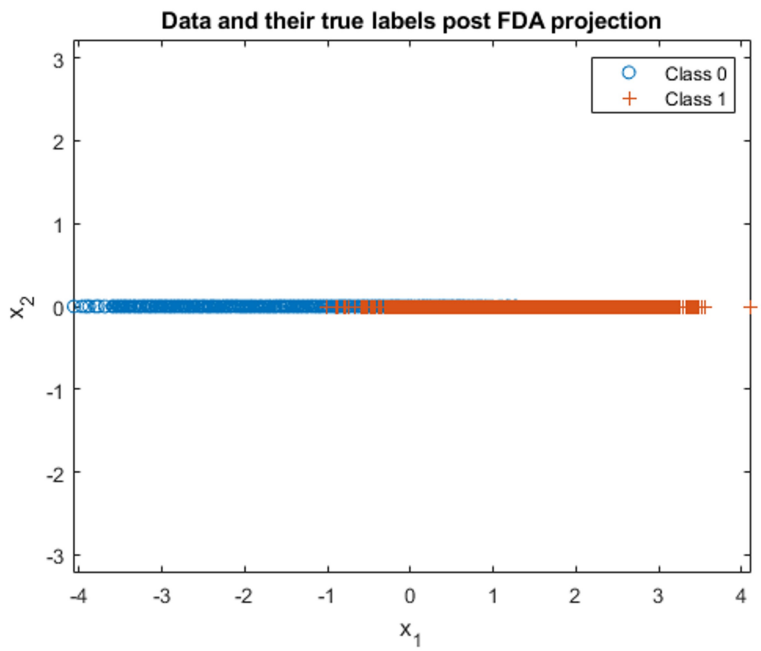
```
legend('Class 0','Class 1'),
title('Data and their true labels before FDA projection'),
xlabel('x_1'), ylabel('x_2'),
```



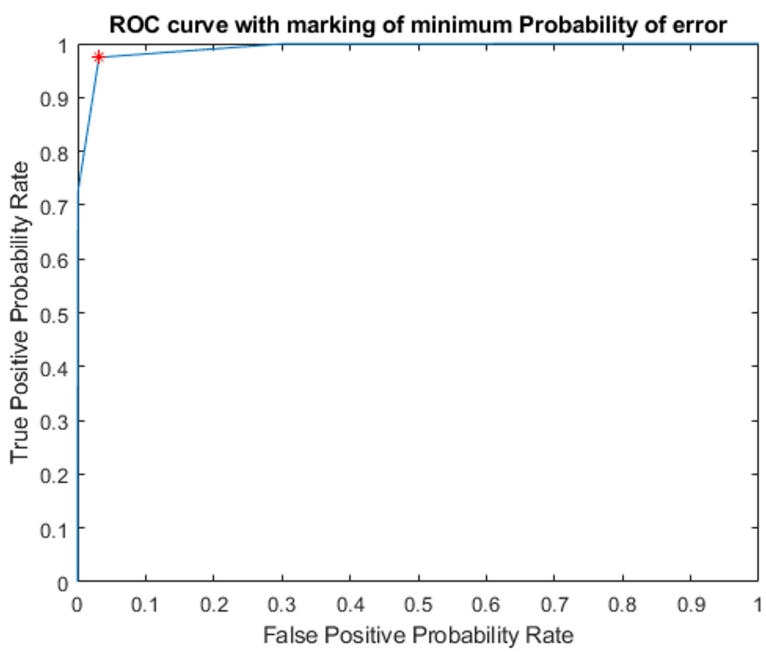
Projection of data set onto an unit vector , with the line rotated to be X-axis for convenience :

Code reference :

```
% Linearly project the data from both categories on to w
ProjectionClass1 = (w'*class1Data);
ProjectionClass2 = (w'*class2Data);

% Compute the difference of mean of class 2 and 1 .
% Based on the delta compute the sign to flip projected matrix
% to ease the job of decision making .
signMatrix = sign(mean(ProjectionClass2) - mean(ProjectionClass1));

% Linearly project the data from both categories on to w
ProjectionClass1 = signMatrix * ProjectionClass1;
ProjectionClass2 = signMatrix * ProjectionClass2;

figure(3), clf,
plot(ProjectionClass1(1,:),zeros(1,N1),'o'), hold on,
plot(ProjectionClass2(1,:),zeros(1,N2),'+'), axis equal,
legend('Class 0','Class 1'),
title('Data and their true labels post FDA projection'),
xlabel('x_1'), ylabel('x_2'),
```



ROC curve : (with the required operating point marker shown in a red star or * as below)



Performance characteristics of the Fischer LDA classifier as below table :
It has outstanding results for this data distribution with min P-error at a very low value .
Interpretation of above ROC curve shows us that the results of this classifier are very inclined to being an ideal discriminator with only a lesser area left out that contributes to errors in probability .

Characteristics	Value
Min P-error	**0.0570**
Threshold	**0**
ROC Marker - True Positive	**0.9745**
ROC Marker - False Positive	**0.0315**

Comments : Compared to the Bayes Classifier , Fischer LDA performs better but there is a catch that in case of datasets having a very high intersection region or say datasets with same mean , then the fischer LDA fails to perform a correct projection thereby failing the classifier

Question 2) For class labels 0 and 1, pick two class priors and two class-conditional pdfs (both in the form of mixtures of two Gaussians). Do not set the class priors to be equal. Use four different Gaussian components when constructing the class-conditional pdfs. Within each class-conditional pdf, do not select the mixture coefficients to be equal. Select your Gaussian mixtures to create an interesting/challenging example.

1 . Provide scatter plots of 1000 total samples from this data distribution. Note that both the true class label and within that class, which Gaussian component generates each sample should be selected randomly, in accordance with class prior probabilities, and Gaussian component probabilities (weights), respectively. Do NOT specify number of samples for any of the labels or components. Indicate true class label fo each sample with a different marker shape in the scatter plot.

Source Code : https://github.com/vikas753/eece5644/blob/master/hw1/question2.m

Please find the equation of the this combined gaussian mixture as below :
(suffixes and their meanings as below :
c11 : Class 1 \ label 0 , component 1 , c12 : Class 1 \ label 0 , component 2
c21 : Class 1 \ label 1 , component 1 , c22 : Class 2 \ label 1 , component 2)

Means : $\mu_{c11}, \mu_{c12}, \mu_{c21}, \mu_{c22}$

Variances : $\sigma_{c11}, \sigma_{c12}, \sigma_{c21}, \sigma_{c22}$

Weights of individual components : $\alpha_{c11}, \alpha_{c12}, \alpha_{c21}, \alpha_{c22}$

Class priors : P_{c1}, P_{c2}

Thus, final dataset or mixture would

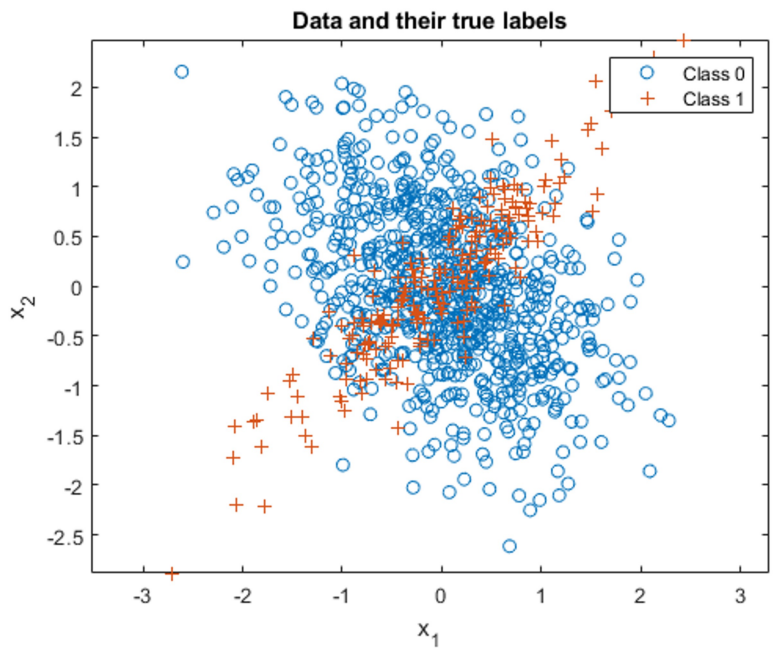
$$\Rightarrow P = P_{c1}\left[\mathcal{N}(\mu_{c11}, \sigma_{c11}) \times \alpha_{c11} + \alpha_{c12} \times \mathcal{N}(\mu_{c12}, \sigma_{c12})\right] + P_{c2}\left[\alpha_{c21} \times \mathcal{N}(\mu_{c21}, \sigma_{21}) + \alpha_{c22} \times \mathcal{N}(\mu_{c22}, \sigma_{c22})\right]$$

Inorder for the dataset to be challenging , it is necessary for large number of samples to be colliding amongst themselves thereby making it difficult to plot a correct decision boundary .
We have chosen a very similar combination of PDFs to realize the same .
Please find the gaussian components as below from the code directly :

```
% means of gaussian mixtures of class label 0
mu(:,1,1) = [-0.1;0];
mu(:,1,2) = [0.3;0];

% means of gaussian mixtures of class label 1
mu(:,2,1) = [0.1;0];
mu(:,2,2) = [-0.3;0];

% Variances of gaussian mixtures of class label 0
Sigma(:,:,1,1) = [1 -0.9;-0.9 1];
```

```
Sigma(:,:,1,2) = [2 1.8;1.8 2];

% Variances of gaussian mixtures of class label 1
Sigma(:,:,2,1) = [1 0.9;0.9 1];
Sigma(:,:,2,2) = [2 1.8;1.8 2];
```

Dataset visuals of the above components is as below :
(Class 1 samples colliding with Class 0 ones with only few of them in distinct co-ordinates)



Decision rule is worked out as shown in below page : (from the final equation decision boundary , would be four parabolae's (each having a concentric group of parabolaes) with different axis and vertex)

As we know from Likelihood - ratio test that

$$\frac{P(L=0|x)}{P(L=1|x)} \underset{L=1}{\overset{L=0}{\gtrless}} \frac{(\lambda_{12}-\lambda_{22})\,P(\omega_2)}{(\lambda_{21}-\lambda_{11})\,P(\omega_1)}$$

threshold

As we have seen previously, that "threshold" is '1' for minimum-probability of error case as it gives an equal chance of classification if the Risk associated with misclassification is not a concern

Thus, we have

$$\frac{P(L=0|x)}{P(L=1|x)} \underset{L=1}{\overset{L=0}{\gtrless}} 1$$

Taking - "log" on both side and substituting pdf equation's we would get

$$\ln\left[\frac{1}{|\Sigma_{00}|^2} e^{-(x-\mu_{00})\Sigma_{00}^{-1}(x-\mu_{00})^T} + \frac{1}{|\Sigma_{01}|^2} e^{-(x-\mu_{01})\Sigma_{01}^{-1}(x-\mu_{01})^T}\right]$$

$$\Rightarrow \underset{L=1}{\overset{L=0}{\gtrless}} \ln\left[\frac{1}{|\Sigma_{10}|^2} e^{-(x-\mu_{10})\Sigma_{10}^{-1}(x-\mu_{10})^T} + \frac{1}{|\Sigma_{11}|^2} e^{-(x-\mu_{11})\Sigma_{11}^{-1}(x-\mu_{11})^T}\right]$$

2 .) Determine the minimum-P(error) classification rule, specify it, draw its decision boundary superimposed on the scatter plot of data, and classify each sample with this classifier and with color cues indicate if the samples are correctly or incorrectly classifier. Using your data samples, calculate an estimate of the smallest probability of error achievable for this dataset and report it.

Performance of the classifier on above dataset is as below (based on above equation) : (as expected it doesn't do a good job) . It says that smallest probability of error that can be achieved that too by varying threshold from 1 to 10000 is around 0.5 with a threshold of 1 .

Code reference :

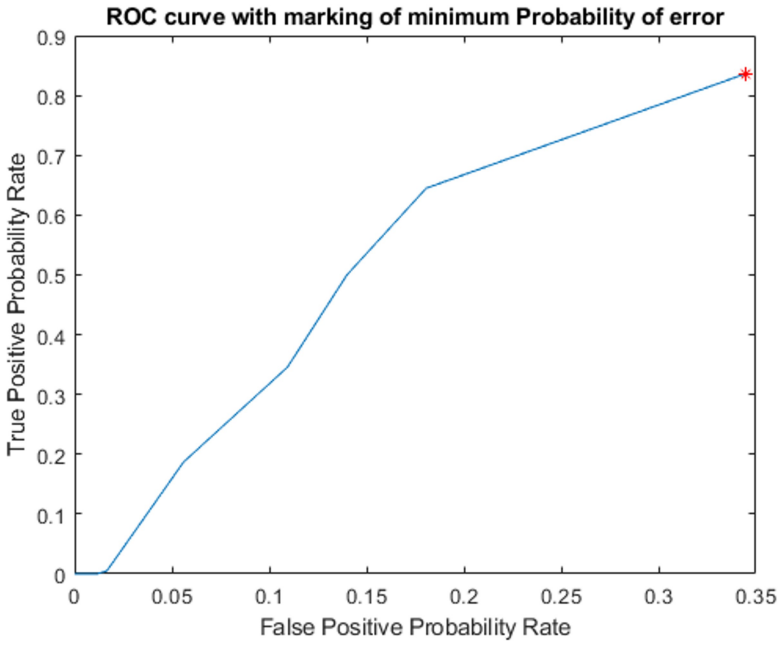
```
discriminantScore = log(weightPdf(1,2) * evalGaussian(x,mu(:,2,1),Sigma(:,:,2,1)) +          weightPdf(2,2) * evalGaussian(x,mu(:,2,2),Sigma(:,:,2,2)))-log(weightPdf(1,1) * evalGaussian(x,mu(:,1,1),Sigma(:,:,1,1)) + weightPdf(2,1) * evalGaussian(x,mu(:,1,2),Sigma(:,:,1,2)));

decision = (discriminantScore >= log(gamma));
```

Charachteristics table (estimates of Min Perror) :

Characteristics	Value
Min P-error	**0.5083**

Threshold	**1**
ROC Marker - True Positive	**0.8364**
ROC Marker - False Positive	**0.3448**

ROC curve : (It is very close to being a linear curve , which gives a notion that the classifier doesn't do any job of prediction . In my opinion it is difficult to design a perfect classifier for any dataset like above having high number of collisions)

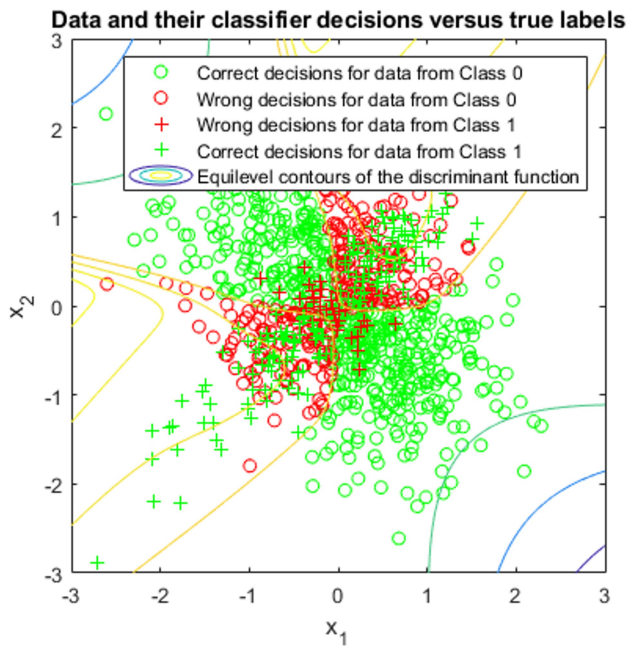


Please find the classifier decision boundary (as expected those are 4 concentric parabolae's with different axis and vertices) . Legend indicates the color and symbol for correctly classified as well as misclassified data .

Code reference :

```
discriminantScore = log(weightPdf(1,2) * evalGaussian(x,mu(:,2,1),Sigma(:,:,2,1)) + weightPdf(2,2) *
evalGaussian(x,mu(:,2,2),Sigma(:,:,2,2)))-log(weightPdf(1,1) * evalGaussian(x,mu(:,1,1),Sigma(:,:,1,1)) + weightPdf(2,1)
* evalGaussian(x,mu(:,1,2),Sigma(:,:,1,2)));
decision = (discriminantScore >= 0);

ind00 = find(decision==0 & label==0);
ind10 = find(decision==1 & label==0);
ind01 = find(decision==0 & label==1);
ind11 = find(decision==1 & label==1);

figure(1), % class 0 circle, class 1 +, correct green, incorrect red
plot(x(1,ind00),x(2,ind00),'og'); hold on,
plot(x(1,ind10),x(2,ind10),'or'); hold on,
plot(x(1,ind01),x(2,ind01),'+r'); hold on,
plot(x(1,ind11),x(2,ind11),'+g'); hold on,
axis equal,
```

Classified Data with contours :



Comments : Although the above classifier fared very well when performed against the dataset as in Question 1 , but it fails rather very badly when we have knowledge about the PDF having a lot of intersection of data in between , we could probably work it out and improve it's performance by chosing a threshold and risk based on prior probability knowledge .

Question 3)

For a scalar random variable that may be generated by one of two classes, where class priors are equal and class-conditional pdfs are both unit-variance Gaussians with mean values -2 and 2, determine the classification rule that achieves minimum probability of error. Also express the

smallest error probability achievable by this classifier in terms of definite integrals of Gaussian class conditional pdfs involved.

Please find the solution to Question 3 worked as shown in below pages

Let the scalar - random variable be X

Thus the classes are [label 0, label 1]

$\mu_0 = -2, \ \sigma_0 = 1$

$\mu_1 = 2, \ \sigma_1 = 1$

and as per the question [since class priors are equal]

$P(L=0) = 0.5$

$P(L=1) = 0.5$

$$P(x \mid L=0) = \frac{1}{\sqrt{2\pi}} e^{-(x+2)^2} \quad \ldots (1)$$

$$\left[\text{since, } P(x \mid \omega_i) = \frac{1}{\sqrt{2\pi}\,\sigma^2} e^{-(x-\mu)^2}\right]$$

gaussian pdf for random scalar vector

$$P(x \mid L=1) = \frac{1}{\sqrt{2\pi}} e^{-(x-2)^2}$$

As per the Likelihood - ratio test equation

$$\frac{P(x \mid L=0)}{P(x \mid L=1)} \underset{L=1}{\overset{L=0}{\gtrless}} \underbrace{y}_{\text{threshold}} \frac{P(L=1)}{P(L=0)}$$

$$\therefore \frac{e^{-(x+2)^2}}{e^{-(x-2)^2}} \underset{L=1}{\overset{L=0}{\gtrless}} y$$

$\therefore$ Taking 'log' on both sides

$\therefore (x-2)^2 - (x+2)^2 \gtrless \ln y$

$\therefore -8x \gtrless \ln y$

For above equation, min-risk would be when $y = 1$

$$\therefore x \underset{L=0}{\overset{L=1}{\gtrless}} 0 \quad \ldots (2)$$

Average probability of error would be

Label = 1, for feature $x = 0$ ... $p10$

Label = 0, for feature $x = 1$ ... $p01$

$$\therefore \quad p10 = \int_{-\infty}^{0} p(x \mid L=1)\, dx$$

$$\therefore \quad p01 = \int_{0}^{\infty} p(x \mid L=0)\, dx$$

$$p_{Error} = p10 + p01$$

$$\therefore \quad p_{Error} = \int_{-\infty}^{0} \left(1/\sqrt{2\pi}\right) e^{-(x-2)^2} dx + \int_{0}^{\infty} \left(\frac{1}{\sqrt{2\pi}}\right) e^{-(x+2)^2} dx$$

$$\Rightarrow \quad \boxed{\therefore \quad p_{Error} = \frac{1}{\sqrt{2\pi}} \int_{0}^{\infty} \left[e^{-(x+2)^2} - e^{-(x-2)^2} \right] dx}$$