Assignment #1
Juan Pablo Bernal
05/21/2020

Question #1

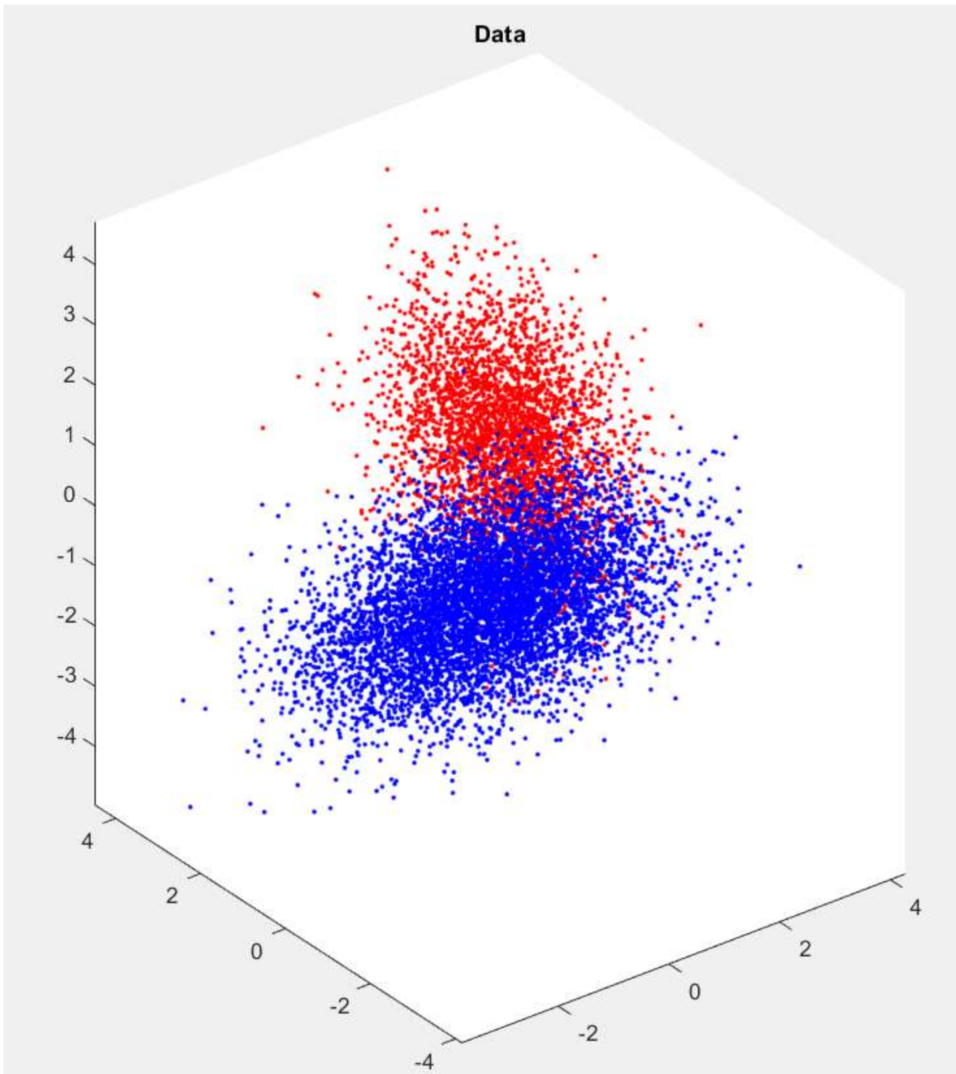

Figure 1. 10000 samples of data generated from 2 gaussian pdfs (Blue = class0, Red = class1)

```
>> Assigment1Q1
P Detect =0.94195
P error =0.058046
theo gamma =-0.61904
P Detection2 =0.94217
P error2 =0.057826
Gamma =-0.58166
Part B
P Detect =0.93009
P error =0.069914
theo gamma =-0.61904
P Detection2 =0.93321
P error2 =0.06679
Gamma =-0.26517
Part C
P detect =0.9356
P error =0.0644
gamma =0.35403
```

Figure 2. The output of queston1 code

Part A:

1.

Theoretical gamma: $ln(\frac{\lambda_{12}-\lambda_{22}}{\lambda_{21}-\lambda_{11}} * \frac{P(w2)}{P(w1)})$ with 0-1 risk the value of gamma is

$$ln(\frac{P(w2)}{P(w1)}) = ln(\frac{0.35}{0.65}) = \ -0.619$$

The classification rule: if the discriminant score $ln(\frac{p(x|L=0)}{p(x|L=1)})$ is less than or equal to the theoretical gamma threshold, it will decide that the sample came from class 1, if it is greater then it came from class 0.

2.

To generate an estimate of the ROC curve (figure 3), the function estimateROC(from the exam solution question 3 code of summer 1 2019) was used. For each gamma the $P(D = i|L = j;\gamma)$ for each of the 4 cases (FP, TP, FN, TN) was calculated by counting the total samples in each case and dividing by the total number of samples of the respective class $\frac{\#samples\ (D = i|L = j;\gamma)}{N_j}$ (the piece of code that calculated this was also extracted from the same code as the ROC function).

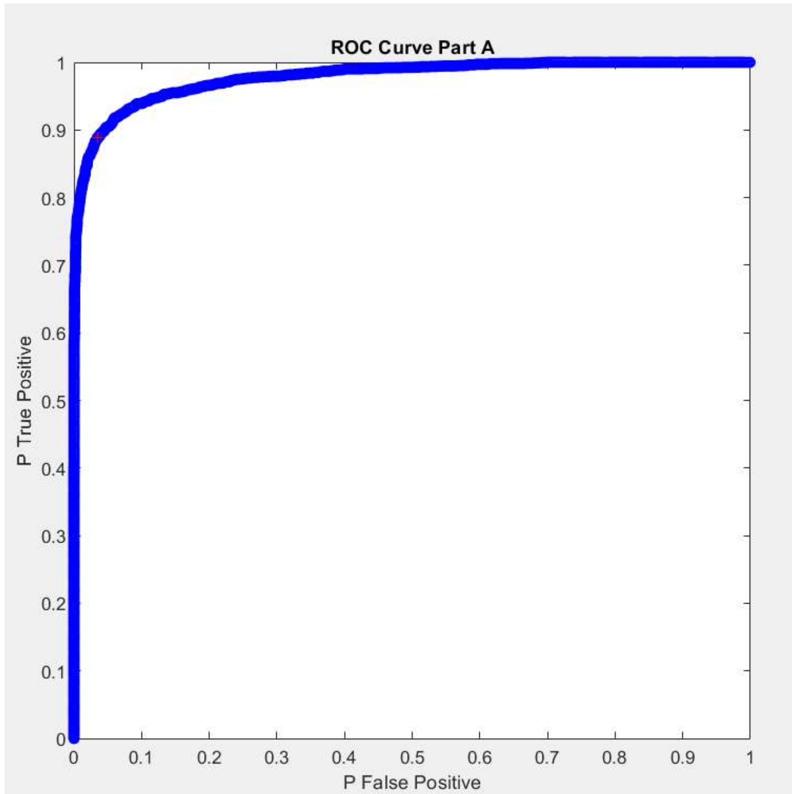

Figure.3 ROC curve (red + is the gamma that minimizes Perror)

3.

The P error for the theoretical gamma was calculated using the given formula

$$P(error;\ \gamma) = P(D = 1|L = 0;\ \gamma)P(L = 0) + P(D = 0|L = 1;\ \gamma)P(L = 1)$$

Using the pFP and pFN calculated in the same way as in part 2 and multiplying it by the respective class prior. For this model using the theoretical gamma the Perror = 0.058 or 5.8%

Then, The Perror for each gamma that was used to make the ROC curve in part 2 was calculated by using the same approach. Then, the gamma that produced the minimum probability of error was selected and its respective pFP and pTP were plotted as a red + in the ROC curve (figure 3). The minimum Perror found with a value of gamma -0.582 was 0.057 or 5.7%.

The values for the theoretical gamma and the empirically selected gamma are very close and almost the same the only differ by 0.037. This difference may come from the fact that the gamma values used were discrete and maybe the theoretical gamma value fell in between 2 consecutive values. Due to this difference, the probabilities of error were also very close, they differed by 0.1%. Moreover, the empirically selected gamma resulted in a lower Perror than the theoretical gamma. This might be because the empirically selected gamma came from the pdf information (discriminant scores) rather than only the priors and loss information.

Part B

1.

For part B everything was done in the same way but the values for the covariance matrices were different. This time a 3x3 Identity matrix was assumed for both sigmas. The theoretical threshold value and the classification rule didn't change since the loss and priors remained the same. What changed was the $P(x|w_i)$ since the sigmas were modified.

2.

The estimate of the ROC curve was calculated in the same way with the new sigma matrices for each class (figure 4.)

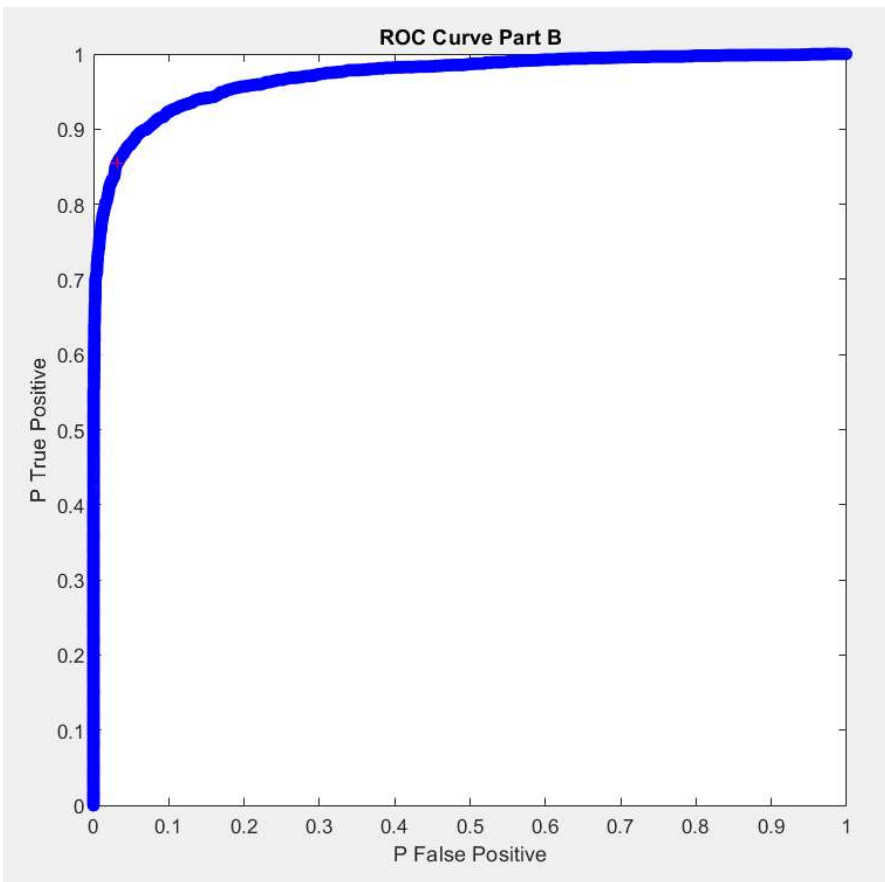

Figure.4 ROC curve part B (red + is the gamma that minimizes Perror)

3.

Both probabilities of errors were calculated in the same way as it was done in part A. With the theoretical gamma an error of 0.069 or 6.9% was found. And with a gamma value of -0.265, the error was minimized and found to be 0.067 or 6.7%. Again, its respective pFP and pTP were plotted as a red + in the ROC curve (figure 3)

This time the gamma values differed a lot because the second one was found by using the pdfs with the new incorrect sigmas. However, the error was very close with only a 0.3% difference. Both the ROC curve and the minimum achievable probability of error were negatively impacted. In the ROC curve, we can see that in the second one the AUC is smaller (area closer to 1 is better). Also, we can see an increment of 1.1% in the total probability of error from the theoretical gamma and 1% with the empirically selected gamma. Moreover, the empirically selected gamma resulted in a smaller Perror as it did in Part A.

Part C

For part c, the LDA classification rule was used instead of ERM. However, this time instead of using the true values for mean and covariance, the sample average and sample covariance were used as the gaussian parameters. Then with these values the Sb (scatter between) and Sw (scatter within) were calculated with: $Sb = (mean0 - mean1) * (mean0 - mean1)' \ and\ Sw = sigma0 + sigma1$ then the WLDA vector (projection vector) was found by doing the eigenvalue decomposition of $Sw^{-1} * Sb$ (code extracted from the exam solution question 3 code of summer 1 2019) and choosing the eigenvector with the largest eigenvalue.

The classification rule this time was if $w^T x >= \gamma$ then decide class 1 if it is less than gamma then decide class 0. This time there wasn't a theoretical gamma value and the threshold value was found by doing the ROC curve (same way as before) and finding the gamma that minimizes the total probability of error. This time the probability of error was found by using the equation $P(error;\ \gamma) = \frac{P(D=1|L=0;\gamma)*N0 + 1-P(D=1|L=1;\gamma)*N1}{N}$ (also taken from the exam solution question 3 code of summer 1 2019).

The final result was that the gamma value that minimizes the Perror was 0.35 and the total Perror was 0.064 or 6.4%. Compared to the previous models, it performed worse than the first model by a 0.7% increase in the total Perror. However, it performed better than the second model by a 0.3% reduction of the Perror. It performed worse than the first one because it wasn't using the real values of sigma and mean but an approximation. Moreover, it performed better than the second one because these approximations are better than the incorrect assumption of the identity matrix for covariances.

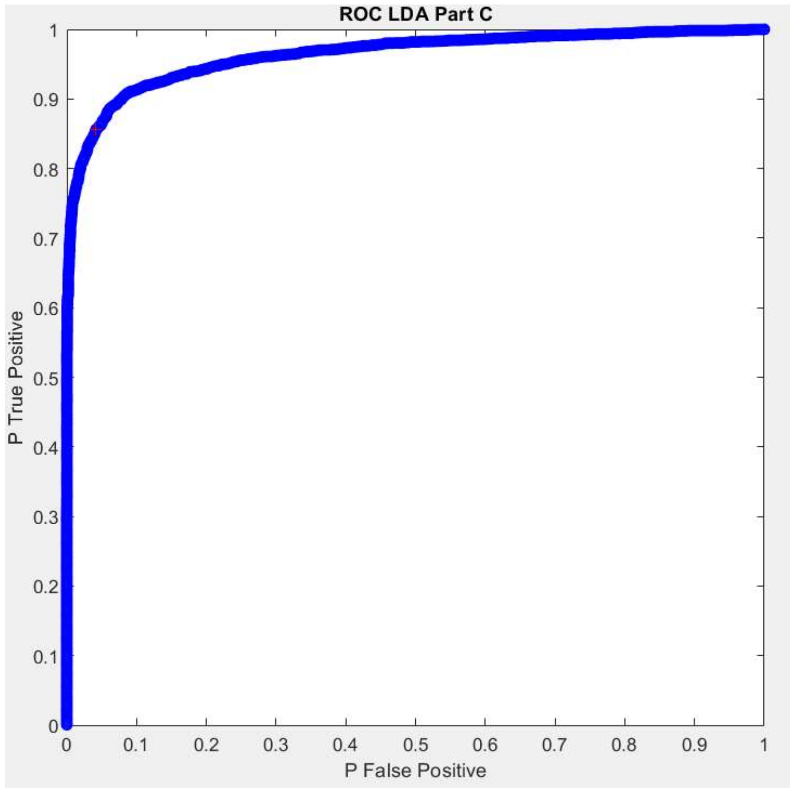

Figure.5 ROC LDA curve (red + is the gamma that minimizes Perror)

Question #2

The data this time was generated from 4 gaussian pdfs all equally likely (all class priors = 0.25) with mean and covariance matrices as follows:

Class 1: $\mu_1 = [4; 1]$ $\Sigma_1 = [4\ -3; -3\ 4]$ Blue in figure 6.
Class 2: $\mu_2 = [0; -2]$ $\Sigma_2 = [3\ 0; 0\ 1]$ Red in figure 6.
Class 3: $\mu_3 = [-4; 1]$ $\Sigma_3 = [4\ 3; 3\ 4]$ Black in figure 6.
Class 4: $\mu_4 = [0; 0]$ $\Sigma_4 = [1\ 0; 0\ 1]$ yellow in figure 6.

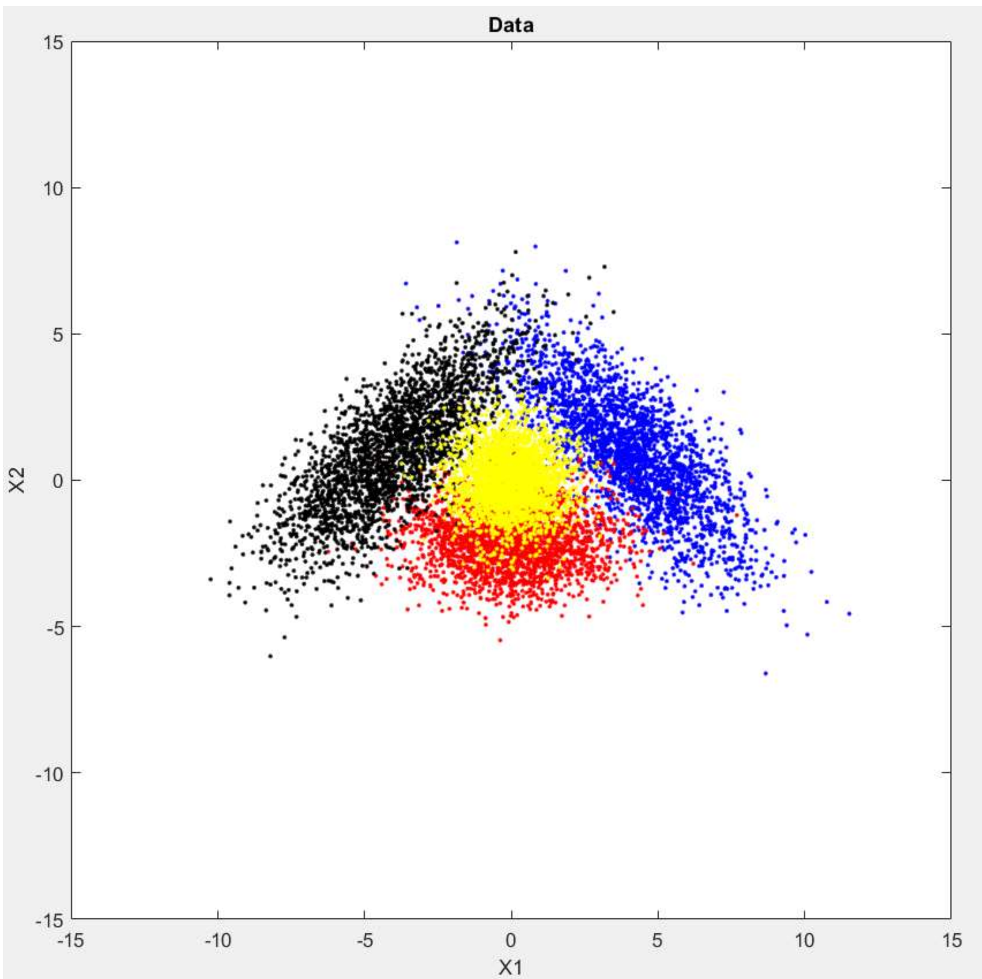

Figure 6.10000 Samples of Data from 4 gaussian pdfs

```
>> Assigment1Q2
Part A
P error =0.11601
Part B
P error =0.13206
```

Figure 7. Question #2 Code Output

Part A

Using 0-1 loss the classification rule used was MAP. Therefore, the decisions were made by first computing the likelihood of a sample with each the 4 different pdfs multiplied by the class prior. Then, it was decided that the sample came from the class that its pdf gave the maximum likelihood. After all the decisions were made, the $P(D = i|L = j)$ for $i, j \in \{1, 2, 3, 4\}$ were calculated. Then, the total probability of error was computed using the equation $1-(P(D = 1|L = 1) * P(L=1) + P(D = 2|L = 2) * P(L=2) + P(D = 3|L = 3) * P(L=3) + P(D = 4|L = 4) * P(L=4))$ and the result was 0.116 or 11.6%. Also, the confusion matrix was empirically estimated (figure 8). Where the rows are the decisions and the columns are the truth. Finally, the scatter plot of the decisions was made. In this plot (figure 9) the class markers are o class 1, + class 2, star class 3, triangle class 4. Read means that the sample was incorrectly classified and green that it was correctly classified.

	1	2	3	4
1	2344	39	62	53
2	25	2034	23	334
3	50	41	2360	62
4	50	371	54	2098

Figure 8. Confusion matrix

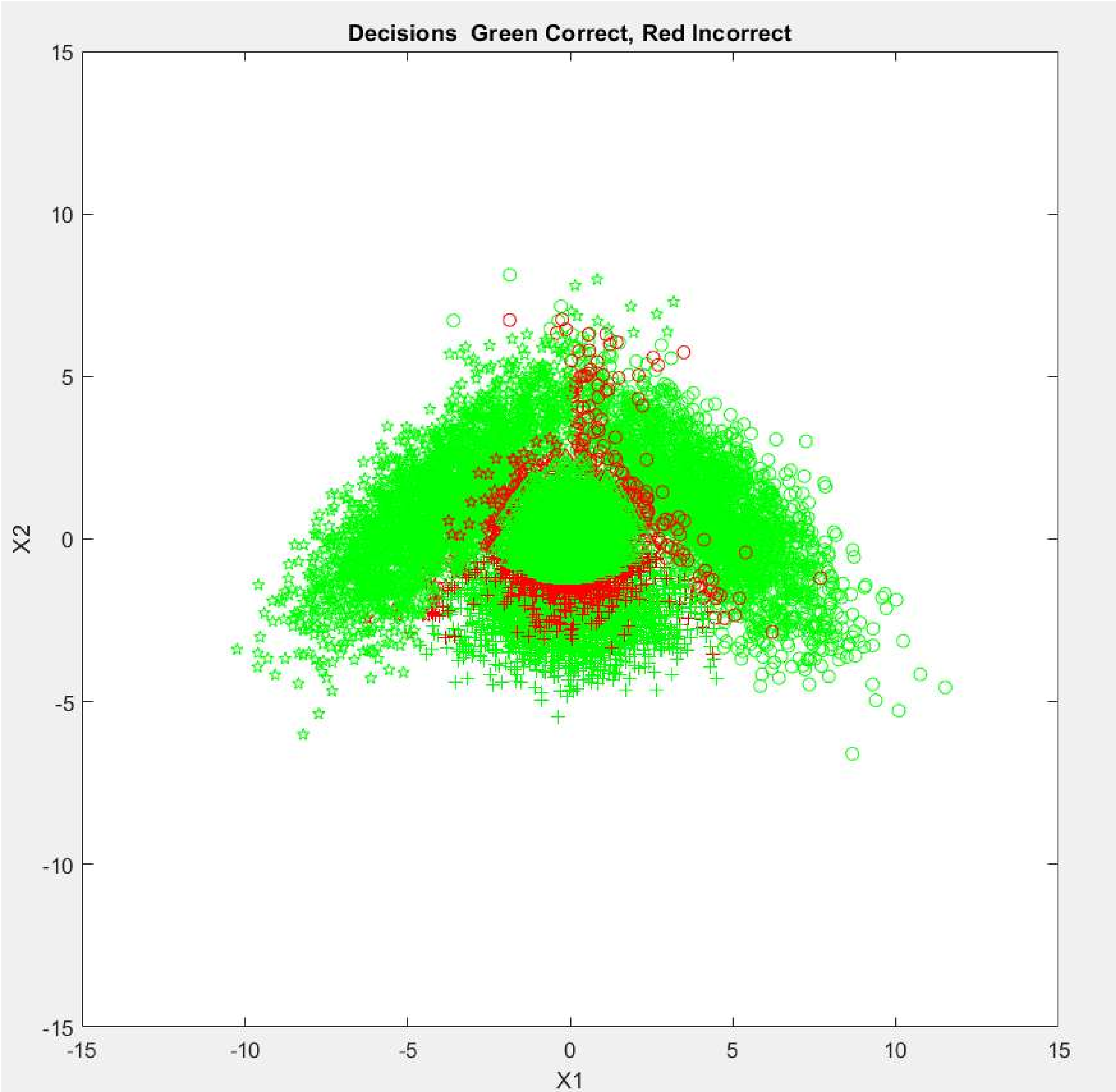

Figure 9. Scatter plot of the decisions made by the MAP model

Part B

With the same data, an ERM classification rule was designed. This time there was a risk of 3 if a sample was wrongly classified as class 4 (this was the class in the middle of the triangle) and 1 if it was wrongly classified as any of the other 3 classes. Therefore, the risk matrix then was:

$$\Lambda = \begin{bmatrix} 0 & 1 & 1 & 3 \\ 1 & 0 & 1 & 3 \\ 1 & 1 & 0 & 3 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

To implement the ERM classification, first, the risks of each action (deciding that a sample came from a specific class) were calculated with $R(L=j) = \sum_{i=1}^{c} P(x|w_i) * p(i) * loss(j,i)$ then a decision was made by taking the action that had the minimum risk. Again, a scatter plot of the decisions was made (figure 10). With the same markers and colors as before. As predicted we can see that the decision region for class 4 is bigger. Both by having more green triangles in the middle (correct decisions). Overall the classifier decided more often that a sample came from class 4 than the previous model.

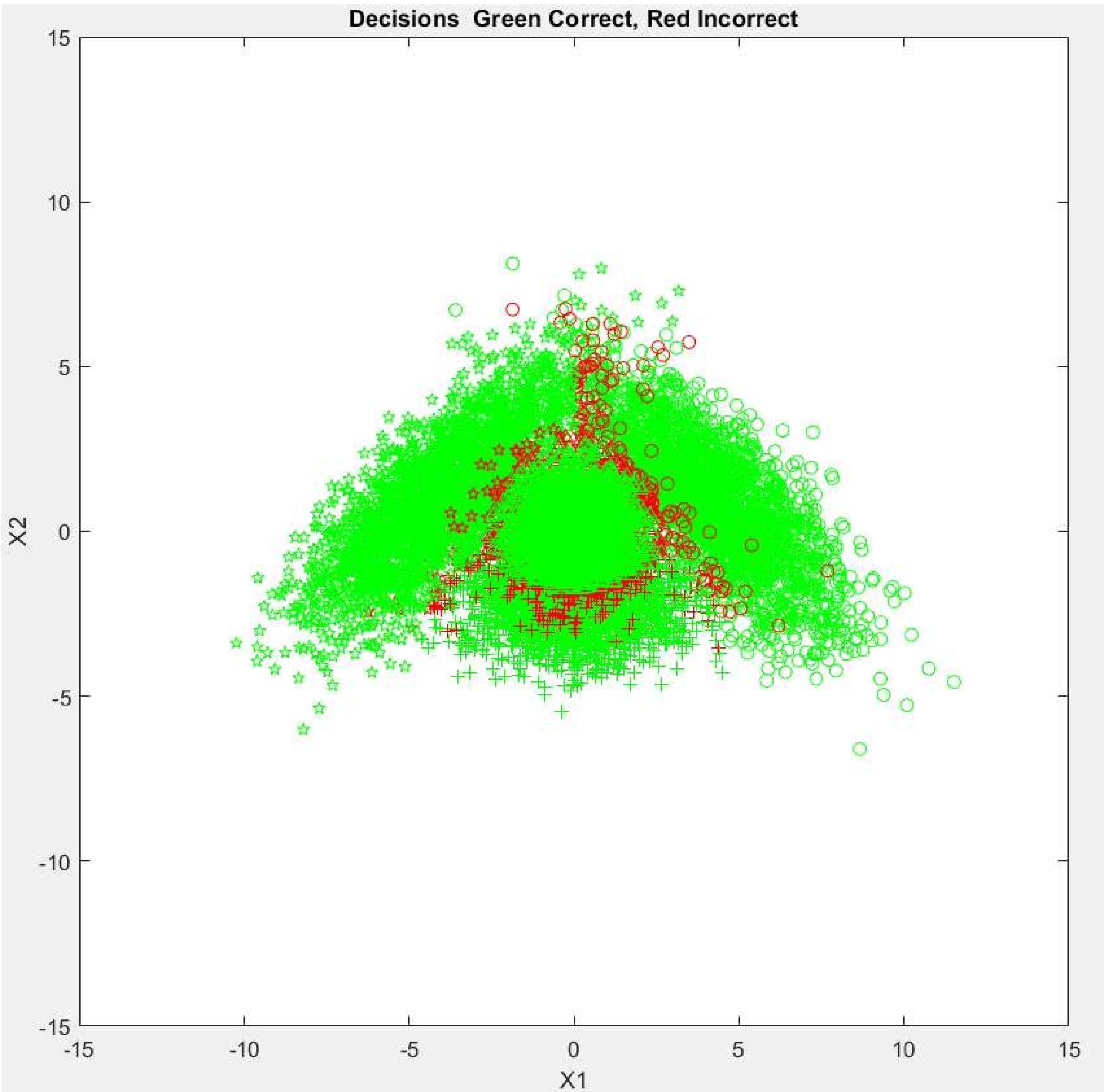

Figure 10. Scatter plot of the decisions made by the ERM model

Finally, the total Perror was calculated as it was in Part A and it was found to be 0.132 or 13.2%. We can see that the second model has a greater probability (a 1.6% increase) because it is forced to misclassify some border samples of the other classes and also a misclassification from class 4 counts as if it had made 3 errors. Therefore, this model makes more errors but reduces risk.

Citations

The format of the variables and the code was based on the solution for exam 1 question 3 from summer 1 2019. Moreover, the ROC estimation function and some pieces of generic code (like plotting, probabilities calculations, and basic equations) were copied from the same code.

Code can be found in g/drive
https://drive.google.com/drive/u/1/folders/1z8xTQoF4b07dwwHcF-m45HDsi3mrYIYq file Exam1_Q3.m

The gaussian data generation code was given in the assignment.

Appendix A

Code for question # 1

```
clear all, close all,
%generate 10000 samples from the gaussian pdfs
N = 10000; p0 = 0.65; p1 = 0.35;
u = rand(1,N)>=p0; N0 = length(find(u==0)); N1 = length(find(u==1)); Nc =
[N0 N1];
mu0 = [-1/2;-1/2;-1/2]; Sigma0 = [1,-0.5,0.3;-0.5,1,-0.5;0.3,-0.5,1];
x0 = mvnrnd(mu0, Sigma0, N0);
figure(1),subplot(1,2,1), plot3(x0(:,1),x0(:,2),x0(:,3),'.b'); axis equal,
hold on;
mu1 = [1;1;1]; Sigma1 = [1,0.3,-0.2;0.3,1,0.3;-0.2,0.3,1];
x1 = mvnrnd(mu1, Sigma1, N1);
figure(1),subplot(1,2,1), plot3(x1(:,1),x1(:,2),x1(:,3),'.r'); axis
equal,hold on;
title ('Data');

%put all samples together and make the labels vector
x = [x0 ; x1];
label = [zeros(1,N0) ones(1,N1)];
loss = [0 1; 1 0];
```

```
%Part A
%generate the discriminant scores for all samples
R0 = mvnpdf(x,mu0',Sigma0);
R1 = mvnpdf(x,mu1',Sigma1);
discriminantScore = (log((R0./R1)))';

%PART A.1
%threshold as a function of priors and loss value
Tgamma = log(((loss(1,2)-loss(2,2))*p1)/((loss(2,1)-loss(1,1))*p0));

%decision and calculation of probabilities of the 4 cases
decision = (discriminantScore <= Tgamma); % use smallest min-error
threshold
ind00 = find(decision==0 & label==0); p100 = length(ind00)/Nc(1); %
probability of true negative
ind10 = find(decision==1 & label==0); p110 = length(ind10)/Nc(1); %
probability of false positive
ind01 = find(decision==0 & label==1); p101 = length(ind01)/Nc(2); %
probability of false negative
ind11 = find(decision==1 & label==1); p111 = length(ind11)/Nc(2); %
probability of true positive

%A.2
%estimate the ROC curve and find the value of tao that minimizes the
probability of total error
[ROC,tau] = estimateROC(discriminantScore,Nc,label);
probError = [((ROC(1,:)'*p0)+(ROC(3,:)'*p1))]; % probability of total error
for different threshold values
[pEmin,ind] = min(probError);


% Display the estimated ROC curve with blue o
% and indicate the minimizer tau with a red +
figure(1),subplot(1,2,2), plot(ROC(1,:),ROC(2,:),'bo'); hold on,
plot(ROC(1,ind),ROC(2,ind),'r+');
axis equal, xlim([0,1]); ylim([0,1]),xlabel('P False Positive'),ylabel('P
True Positive');
title('ROC Curve Part A');

%A.3
```

```
TotalPError1 = p110*p0+p101*p1;
disp(strcat('P Detect = ',num2str(1-TotalPError1))),
disp(strcat('P error = ',num2str(TotalPError1))),
disp(strcat('theo gamma = ',num2str(Tgamma)));


disp(strcat('P Detection2 = ',num2str(1-pEmin))),
disp(strcat('P error2 = ',num2str(pEmin))),
disp(strcat('Gamma = ',num2str((tau(ind)))));


%part B

disp("Part B");
figure(2),subplot(1,2,1), plot3(x0(:,1),x0(:,2),x0(:,3),'.b'); axis equal,
hold on;
figure(2),subplot(1,2,1), plot3(x1(:,1),x1(:,2),x1(:,3),'.r'); axis
equal,hold on;
title ('Data');
newSigma0 = eye(3);
newSigma1 = eye(3);
%generate the discriminant scores for all samples
R0 = mvnpdf(x,mu0',newSigma0);
R1 = mvnpdf(x,mu1',newSigma1);
discriminantScore = (log((R0./R1)))';

%PART B.1
%threshold as a function of priors and loss value
Tgamma = log(((loss(1,2)-loss(2,2))*p1)/((loss(2,1)-loss(1,1))*p0));

%desicion and calculation of probabilities of the 4 cases
decision = (discriminantScore <= Tgamma); % use smallest min-error
threshold
ind00 = find(decision==0 & label==0); p100 = length(ind00)/Nc(1); %
probability of true negative
ind10 = find(decision==1 & label==0); p110 = length(ind10)/Nc(1); %
probability of false positive
ind01 = find(decision==0 & label==1); p101 = length(ind01)/Nc(2); %
probability of false negative
ind11 = find(decision==1 & label==1); p111 = length(ind11)/Nc(2); %
probability of true positive
```

```
%B.2
%estimate the ROC curve and find the value of tao that minimizes the
probability of total error
[ROC,tau] = estimateROC(discriminantScore,Nc,label);
probError = [((ROC(1,:)'*p0)+(ROC(3,:)'*p1))]; % probability of total error
for different threshold values
[pEmin,ind] = min(probError);


% Display the estimated ROC curve with blue o
% and indicate the minimizer tau with a red +
figure(2),subplot(1,2,2), plot(ROC(1,:),ROC(2,:),'bo'); hold on,
plot(ROC(1,ind),ROC(2,ind),'r+');
axis equal, xlim([0,1]); ylim([0,1]),xlabel('P False Positive'),ylabel('P
True Positive');
title('ROC Curve Part B');

%B.3

TotalPError1 = p110*p0+p101*p1;
disp(strcat('P Detect = ',num2str(1-TotalPError1))),
disp(strcat('P error = ',num2str(TotalPError1))),
disp(strcat('theo gamma = ',num2str(Tgamma)));


disp(strcat('P Detection2 = ',num2str(1-pEmin))),
disp(strcat('P error2 = ',num2str(pEmin))),
disp(strcat('Gamma = ',num2str((tau(ind)))));

%Part C
disp("Part C");
figure(3),subplot(1,3,1), plot3(x0(:,1),x0(:,2),x0(:,3),'.b'); axis equal,
hold on;
figure(3),subplot(1,3,1), plot3(x1(:,1),x1(:,2),x1(:,3),'.r'); axis
equal,hold on;
title ('Data');

Emu0 = mean(x0);
Esigma0 = cov(x0);
Emu1 = mean(x1);
Esigma1 = cov(x1);
```

```
Sb = (Emu0-Emu1)*(Emu0-Emu1)';
Sw = Esigma0+Esigma1;

[V,D] = eig(inv(Sw)*Sb);
[~,ind] = sort(diag(D),'descend');
w = V(:,ind(1));

y = w'*x';

wLDA = sign(mean(y(find(label==1)))-mean(y(find(label==0))))*w; % ensures
class1 falls on the + side of the axis
discriminantScore =
sign(mean(y(find(label==1)))-mean(y(find(label==0))))*y; % flip yLDA
accordingly
% Estimate the ROC curve for this LDA classifier
[ROC,tau] = estimateROCLDA(discriminantScore,label);
probError = [ROC(1,:)',1-ROC(2,:)']*Nc'/N; % probability of error for LDA
for different threshold values
[pEmin,ind] = min(probError);
% Display the estimated ROC curve for LDA and indicate the operating points
% with smallest empirical error probability estimates (could be multiple)
figure(3), subplot(1,2,2),plot(ROC(1,:),ROC(2,:),'bo'); hold on,
plot(ROC(1,ind),ROC(2,ind),'r+');
axis equal, xlim([0,1]); ylim([0,1]),xlabel('P False Positive'),ylabel('P
True Positive');
title('ROC LDA Part C');

%C.3

disp(strcat('P detect = ',num2str(1-pEmin))),
disp(strcat('P error = ',num2str(pEmin))),
disp(strcat('gamma = ',num2str(tau(ind))));


function [ROC,tau] = estimateROC(discriminantScore,Nc,label)
% Generate ROC curve samples
sortedScore = sort(discriminantScore,'ascend');
tau =
[sortedScore(1)-1,(sortedScore(2:end)+sortedScore(1:end-1))/2,sortedScore(e
nd)+1];
%thresholds at midpoints of consecutive scores in sorted list
for k = 1:length(tau)
```

```
    decision = (discriminantScore <= tau(k));
    ind00 = find(decision==0 & label==0); p00 = length(ind00)/Nc(1); %
probability of true negative
    ind10 = find(decision==1 & label==0); p10 = length(ind10)/Nc(1); %
probability of false positive
    ind01 = find(decision==0 & label==1); p01 = length(ind01)/Nc(2); %
probability of false negative
    ind11 = find(decision==1 & label==1); p11 = length(ind11)/Nc(2); %
probability of true positive
    ROC(:,k) = [p10;p11;p01;p00];
end
end

%ROC estimation function
function [ROC,tau] = estimateROCLDA(discriminantScore,label)
% Generate ROC curve samples
Nc = [length(find(label==0)),length(find(label==1))];
sortedScore = sort(discriminantScore,'ascend');
tau =
[sortedScore(1)-1,(sortedScore(2:end)+sortedScore(1:end-1))/2,sortedScore(e
nd)+1];
%thresholds at midpoints of consecutive scores in sorted list
for k = 1:length(tau)
    decision = (discriminantScore >= tau(k));
    ind10 = find(decision==1 & label==0); p10 = length(ind10)/Nc(1); %
probability of false positive
    ind11 = find(decision==1 & label==1); p11 = length(ind11)/Nc(2); %
probability of true positive
    ROC(:,k) = [p10;p11];
end
end
```

Appendix B

Code for question 2

```
clear all, close all,

%generate 10000 samples from 4 different gaussian pdfs
N = 10000; p = 0.25;
```

```
u = rand(1,N);
N1 = length(find(u <=0.25));
N2 = length(find((u > 0.25) & (u <=0.5)));
N3 = length(find((u > 0.5) & (u <=0.75)));
N4 = length(find(u>0.75));
Nc = [N1 N2 N3 N4];

mu1 = [4;1]; Sigma1 = [4 -3;-3 4];
x1 = mvnrnd(mu1, Sigma1, N1);
figure(1), plot(x1(:,1),x1(:,2),'.b'); axis equal, hold on;
mu2 = [0;-2]; Sigma2 = [3 0;0 1];
x2 = mvnrnd(mu2, Sigma2, N2);
figure(1),plot(x2(:,1),x2(:,2),'.r'); axis equal,hold on;
mu3 = [-4;1]; Sigma3 = [4 3;3 4];
x3 = mvnrnd(mu3, Sigma3, N3);
figure(1), plot(x3(:,1),x3(:,2),'.k'); axis equal,hold on;
mu4 = [0;0]; Sigma4 = [1 0;0 1];
x4 = mvnrnd(mu4, Sigma4, N4);
figure(1), plot(x4(:,1),x4(:,2),'.y'); axis equal,hold on;
title ('Data'),xlim([-15,15]), ylim([-15,15]),xlabel('X1'),ylabel('X2');

%put all samples together and make the labels vector
x = [x1; x2; x3; x4]';
label = [ones(1,N1) 2*ones(1,N2) 3*ones(1,N3) 4*ones(1,N4)];

score1 = mvnpdf(x',mu1',Sigma1)*0.25;
score2 = mvnpdf(x',mu2',Sigma2)*0.25;
score3 = mvnpdf(x',mu3',Sigma3)*0.25;
score4 = mvnpdf(x',mu4',Sigma4)*0.25;

%Part A

%put the scores together to search for the maximum
R = [score1 score2 score3 score4];

[~,decision] = max(R');
%find all possible combinations of decision vs truth
ind11 = find(decision==1 & label==1); p11 = length(ind11)/Nc(1);
ind12 = find(decision==1 & label==2); p12 = length(ind12)/Nc(2);
ind13 = find(decision==1 & label==3); p13 = length(ind13)/Nc(3);
ind14 = find(decision==1 & label==4); p14 = length(ind14)/Nc(4);
```

```
ind21 = find(decision==2 & label==1); p21 = length(ind21)/Nc(1);
ind22 = find(decision==2 & label==2); p22 = length(ind22)/Nc(2);
ind23 = find(decision==2 & label==3); p23 = length(ind23)/Nc(3);
ind24 = find(decision==2 & label==4); p24 = length(ind24)/Nc(4);

ind31 = find(decision==3 & label==1); p31 = length(ind31)/Nc(1);
ind32 = find(decision==3 & label==2); p32 = length(ind32)/Nc(2);
ind33 = find(decision==3 & label==3); p33 = length(ind33)/Nc(3);
ind34 = find(decision==3 & label==4); p34 = length(ind34)/Nc(4);

ind41 = find(decision==4 & label==1); p41 = length(ind41)/Nc(1);
ind42 = find(decision==4 & label==2); p42 = length(ind42)/Nc(2);
ind43 = find(decision==4 & label==3); p43 = length(ind43)/Nc(3);
ind44 = find(decision==4 & label==4); p44 = length(ind44)/Nc(4);
%calculate the Total probability of error
TotalPError = 1-(p11*0.25+p22*0.25+p33*0.25+p44*0.25);
%generate the confusion matrix
ConM = [length(ind11) length(ind12) length(ind13) length(ind14);
        length(ind21) length(ind22) length(ind23) length(ind24);
        length(ind31) length(ind32) length(ind33) length(ind34);
        length(ind41) length(ind42) length(ind43) length(ind44)];

disp('Part A');
disp(strcat('P error = ',num2str(TotalPError)));

%Plot all the decisions made, Correct are green and incorrect are red
%Markers are: o class 1, + class 2, star class 3, triangle class 4

figure(2),
plot(x(1,ind11),x(2,ind11),'og'); hold on,
plot(x(1,ind12),x(2,ind12),'or'); hold on,
plot(x(1,ind13),x(2,ind13),'or'); hold on,
plot(x(1,ind14),x(2,ind14),'or'); hold on,

plot(x(1,ind21),x(2,ind21),'+r'); hold on,
plot(x(1,ind22),x(2,ind22),'+g'); hold on,
plot(x(1,ind23),x(2,ind23),'+r'); hold on,
plot(x(1,ind24),x(2,ind24),'+r'); hold on,

plot(x(1,ind31),x(2,ind31),'pg'); hold on,
plot(x(1,ind32),x(2,ind32),'pr'); hold on,
plot(x(1,ind33),x(2,ind33),'pg'); hold on,
```

```
plot(x(1,ind34),x(2,ind34),'pr'); hold on,

plot(x(1,ind41),x(2,ind41),'^r'); hold on,
plot(x(1,ind42),x(2,ind42),'^r'); hold on,
plot(x(1,ind43),x(2,ind43),'^r'); hold on,
plot(x(1,ind44),x(2,ind44),'^g'); hold on,
axis equal, xlim([-15,15]), ylim([-15,15]),xlabel('X1'),ylabel('X2'),
title ('Decisions  Green Correct, Red Incorrect');

%Part B
%Loss/Cost/Risk matrix
loss = [0 1 1 3;1 0 1 3;1 1 0 3;1 1 1 0];
% new scores taking into account the loos values
A1 = score1*loss(1,1)+score2*loss(1,2)+score3*loss(1,3)+score4*loss(1,4);
A2 = score1*loss(2,1)+score2*loss(2,2)+score3*loss(2,3)+score4*loss(2,4);
A3 = score1*loss(3,1)+score2*loss(3,2)+score3*loss(3,3)+score4*loss(3,4);
A4 = score1*loss(4,1)+score2*loss(4,2)+score3*loss(4,3)+score4*loss(4,4);

%Put the action scores together to search for the action that minimizes the
%score
Action = [A1 A2 A3 A4];
[~,decision] = min(Action');

%find all possible combinations of decision vs truth
ind11 = find(decision==1 & label==1); p11 = length(ind11)/Nc(1);
ind12 = find(decision==1 & label==2); p12 = length(ind12)/Nc(2);
ind13 = find(decision==1 & label==3); p13 = length(ind13)/Nc(3);
ind14 = find(decision==1 & label==4); p14 = length(ind14)/Nc(4);

ind21 = find(decision==2 & label==1); p21 = length(ind21)/Nc(1);
ind22 = find(decision==2 & label==2); p22 = length(ind22)/Nc(2);
ind23 = find(decision==2 & label==3); p23 = length(ind23)/Nc(3);
ind24 = find(decision==2 & label==4); p24 = length(ind24)/Nc(4);

ind31 = find(decision==3 & label==1); p31 = length(ind31)/Nc(1);
ind32 = find(decision==3 & label==2); p32 = length(ind32)/Nc(2);
ind33 = find(decision==3 & label==3); p33 = length(ind33)/Nc(3);
ind34 = find(decision==3 & label==4); p34 = length(ind34)/Nc(4);

ind41 = find(decision==4 & label==1); p41 = 3*length(ind41)/Nc(1);
ind42 = find(decision==4 & label==2); p42 = 3*length(ind42)/Nc(2);
ind43 = find(decision==4 & label==3); p43 = 3*length(ind43)/Nc(3);
```

```
ind44 = find(decision==4 & label==4); p44 = length(ind44)/Nc(4);

%calculate the Total probability of error
TotalPError = 1-(p11*0.25+p22*0.25+p33*0.25+p44*0.25);
disp('Part B');
disp(strcat('P error = ',num2str(TotalPError)));

%Plot all the decisions made, Correct are green and incorrect are red
%Markers are: o class 1, + class 2, star class 3, triangle class 4
figure(3),
plot(x(1,ind11),x(2,ind11),'og'); hold on,
plot(x(1,ind12),x(2,ind12),'or'); hold on,
plot(x(1,ind13),x(2,ind13),'or'); hold on,
plot(x(1,ind14),x(2,ind14),'or'); hold on,

plot(x(1,ind21),x(2,ind21),'+r'); hold on,
plot(x(1,ind22),x(2,ind22),'+g'); hold on,
plot(x(1,ind23),x(2,ind23),'+r'); hold on,
plot(x(1,ind24),x(2,ind24),'+r'); hold on,

plot(x(1,ind31),x(2,ind31),'pg'); hold on,
plot(x(1,ind32),x(2,ind32),'pr'); hold on,
plot(x(1,ind33),x(2,ind33),'pg'); hold on,
plot(x(1,ind34),x(2,ind34),'pr'); hold on,

plot(x(1,ind41),x(2,ind41),'^r'); hold on,
plot(x(1,ind42),x(2,ind42),'^r'); hold on,
plot(x(1,ind43),x(2,ind43),'^r'); hold on,
plot(x(1,ind44),x(2,ind44),'^g'); hold on,
axis equal, xlim([-15,15]), ylim([-15,15]),xlabel('X1'),ylabel('X2'),
title ('Decisions  Green Correct, Red Incorrect');
```