ME 466 Introduction to AI Fall
2021
Programming Assignment 2
Submitted on: 7 November 2021

Name: Korkut Emre Arslantürk

Student ID: 250206039

Grade:

1	
2	
3	
Σ	

I hereby declare that the paper I am submitting under this cover is product of my own efforts only. Even if I worked on some of the problems together with my classmates, I prepared this paper on my own, without looking at any other classmate's paper. I am knowledgeable about everything that is written under this cover, and I am prepared to explain any scientific/technical content written here if a short oral examination about this paper is conducted by the instructor. I am aware of the serious consequences of cheating.

Signature:

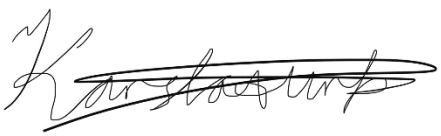

TABLE OF CONTENTS

1. INTRODUCTION

In this assigment, there are 3 different problems.

In the first part of the assignment, we act as an astronout and we try to decide sun will be rise on everyday on unknown planet or not in order to build a solar power plant. Firstly, I had prior probabilities and I tried to find probability of the sun always rises on that planet using total probability principle, conditional probability and Bayes' Rule. Then, I calculated probability of error for my decision by applying Bayes Decision Theory. After that, I found how many days are needed to be sure to build a solar power plant take into consideration cost of solar power plant and daily expenditures.

In the second problem, I faced with two category classification problem in 2 dimensions. Mean and standart standard deviation matrices were given. First of all, equation of the Bayes decision boundary is found and plotted as well as decisions regions were defined according to the equal prior probabilities. Then, same processes are done according to $P(w1) = 1/4$ and $P(w2) = 3/4$. After that previous 2 steps are done with new mean and standart standard deviation matrices. Then possible set of $\mu 1$, $\Sigma 1$, $\mu 2$, $\Sigma 2$ values are found for an ellipse.

In the third problem, I studied on the Wisconsin Breast Cancer data set. The data includes 569 instances and they include features which are extracted from both of benign and malign images' cells. When I read the data on Matlab, I had 569 samples with 30 features. In that problem, my goal is to train a classifier to find a cell is benign or malign. First of all, I divided data sets for training and test using a 10-fold cross validation scheme. Then, the means and covariances are estimated and MAP decision rule is applied to determine the cell as a malign or benign. Confusion matrix was determined. After that, the probability distributions were determined by using kerner density estimation and MAP rule was applied to classify. Determine an appropriate window size h by trial and error. Determine the confusion matrix. Then, amount of Type I and Type 2 errors are calculated. Type I error means that a helathy cell marked as a cancer while Type II means that diseased cell marked as a healthy. After that, because of Type II error has a lot risk, a method to reduce Type II error, has suggested. Finaly, this method was applied for previous parts.

2. BODY OF THE REPORT

Question 1

Pierre-Simon Laplace is a person who establish the sunrise problem in 18th century[1]. In this assignment, my aim is to check whether the sun rises every day on an unknown distant planet or not. According to limited ability of observation on Earth, prior probability of sun rises on that planet is 0.6 while prior probability of it rises 50% of the time is 0.4. When I stayed on the planet for three days, the sun has risen every day.

Part A)

(a) What is the probability that the sun always rises on that planet?

We would like to learn probability of sunrise every day according to first 3 days observation.

First of all, let's say:

P(F)= Probability of sunrise every day = 0.6

P(H)= Probability of sun rises half of the time = 0.4

If we calculate probability of sun has risen 3 days according to P(F) and P(H), using the conditional probability principle:

P(S)= Sunny 3 days situation

Firstly, let's take P(S|H):

In this situation universal set is that sun rises 50% of the time because of that, probability of sun has risen 3 days equal to multiplication of 1/2 for each day.

$$P(S|H) = \frac{1}{2} * \frac{1}{2} * \frac{1}{2} = \frac{1}{8}$$

Then, let's get P(S|F):

In this situation universal set is that sun rises every day, so probability of sun has risen 3 days equal to multiplication of 1 for each day)

$$P(S|F) = 1 * 1 * 1 = 1$$

To find probability of that the sun always rises on that planet, my aim is to find P(F|S). Because of we found P(S|F), we should use Bayes Theorem to find P(F|S). Using Bayes Theorem:

$P(F|S) = \frac{P(S|F).P(F)}{P(S)}$

In this equation we know P(S|F) and P(F), so we need to calculate P(S) and we can do that using total probability principle:

$$P(S) = P(S|F).P(F) + P(H|S).P(H) = 1 * 0.6 * \frac{1}{8} * 0.4 = 0.65$$

Then we can say that:

$$P(F|S) = \frac{P(S|F).P(F)}{P(S)} = \frac{P(S|F).P(F)}{P(S|F).P(F)+P(H|S).P(H)} = \frac{(1)*(0,6)}{1*(0.6)+\left(\frac{1}{8}\right)*(0.4)} = \frac{12}{13} = 0.923$$

(b) What is your decision? What is the probability of error for your decision?

According to the Bayes Decision Theory,

$$P(error|x) = \begin{cases} p(w2|x), if\ we\ decide\ w1 \\ p(w1|x), if\ we\ decide\ w2 \end{cases}$$

In our problem P(W1) = 0.923 and P(W2) = 0.077

So we can say that:

Error for the possibility that the sun rises every day = $P_E = 1/13 = 0.077$

Part B)

Each day you stay on the planet costs $10000. If you decide that the sun will rise every day, your team will build a solar power plant that costs $10 million. But if your decision is wrong, if the sun does not rise even for one day, the power plant will be destroyed and $10 million will be wasted. How many days would you stay and see the sun rise before deciding to build the plant?

To be sure as possible as, I think $P(F|S) = 0.9999$ is enough ratio to build the power plant.

When I applied formula to calculate how many days I need to be sure:

$$P(F|S) = 0.9999 = \frac{P(S|F).P(F)}{P(S)} = \frac{P(S|F).P(F)}{P(S|F).P(F) + P(H|S).P(H)} = \frac{(1) * (0,6)}{1 * (0.6) + \left(\frac{1}{2^n}\right) * (0.4)}$$

$\frac{0.6}{0.6+\frac{0.4}{2^n}} = 0.9999$ .Then, $\frac{(0.6*2^n)}{(0.6)2^n+0.4} = 0.9999$

If we employ Y= $(0.6)2^n$. We may get these:

$$\frac{Y}{Y+0.4} = 0.9999$$

Thus, $Y = 0.9999Y + 0.39996$

$0.0001Y = 0.39996$

$Y = 3999{,}6$. That means this: $(0.6 * 2^n) = 3999{,}6$

Finally, $2^n = 6666$. Hence, 13>n>12. If we seil n to upper limit, we may claim that 13 days will be enough for us to make a decision.

Question 2

Calculate the equation for the decision boundary that divides these two class.

$$p(\omega_i|x) \propto p(x|\omega_i)p(\omega_i)$$

Then in order to get the fewest error ratio for classification, we may write the discriminant function for each class:

$$g_i(x) = p(x|\omega_i)p(\omega_i)$$

Because of we have Gaussian distribution,we can determine function as:

$$g_i(x) = \ln p(x|\omega_i) + \ln p(\omega_i)$$

Then, general formula of g(x):

$$g_i(x) = -\frac{1}{2}(x-\mu_i)^T\Sigma_i^{-1}(x-\mu_i) - \ln 2\pi - \frac{1}{2}\ln|\Sigma_i| + \ln p(\omega_i)$$

The decision boundary would be the solution of $g_1(x) = g_2(x)$.

According to given values: $\mu_1=\begin{pmatrix}0\\0\end{pmatrix}$, $\mu_2=\begin{pmatrix}0\\0\end{pmatrix}$, $\Sigma_1=\begin{pmatrix}1&0\\0&1\end{pmatrix}$, $\Sigma_2=\begin{pmatrix}1&0\\0&1\end{pmatrix}$.

For $G_1(x)$ ➔ $\mu_1=\begin{pmatrix}0\\0\end{pmatrix}$, $\Sigma_1=\begin{pmatrix}1&0\\0&1\end{pmatrix}$ and $x=\begin{bmatrix}x1\\x2\end{bmatrix}$

$Gi(X)= -\frac{1}{2} * (x-\mu i)^T * \sum_i^{-1}(x-\mu i) + lnP(wi)$

$G_1(x) = -\frac{1}{2} * \begin{bmatrix} x1 \\ x2 \end{bmatrix}^T * \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} x1 \\ x2 \end{bmatrix} + lnP(w1)$=(-1/2)* $(x_1)^2 + (x_2)^2$ + - ln2π - (1/2)*ln(1) + lnP{W_1}

For $G_2(x)$ ➔ $\mu_1=\begin{pmatrix} 0 \\ 0 \end{pmatrix}$, ,$\Sigma_1=\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ and $x=\begin{bmatrix} x1 \\ x2 \end{bmatrix}$

$G_2(x) = -\frac{1}{2} * \begin{bmatrix} x1-1 \\ x2-1 \end{bmatrix}^T * \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} x1-1 \\ x2-1 \end{bmatrix} - ln2\pi - \frac{1}{2}\ln |\Sigma 2| + lnP(w2) =$

-(1/2)* $(x_1-1)^2 + (x_2-1)^2$ - ln2π - (1/2)*ln(1) + lnP{W_2}

Part A

For the equal prior probabilities:

$G_1(X)=G_2(X)=x1 + x2 - 1 = 0$

```
% Part A
nu_1c=[0 0]; nu_2c=[1 1];
sigma_1c_a=[1 0;0 1]; sigma_2c_a=[1 0;0 1];
R = chol(sigma_1c_a);
z1 = repmat(nu_1c,100,1) + randn(100,2)*R;
R = chol(sigma_2c_a);
z2 = repmat(nu_2c,100,1) + randn(100,2)*R;
figure
plot(z1(:,1),z1(:,2),'b.')
hold on;
plot(z2(:,1),z2(:,2),'r.')
syms x1 x2;
g1_e = -0.5*([x1;x2]-nu_1c')'*inv(sigma_1c_a)*([x1;x2]-nu_1c')-0.5*log(det(sigma_1c_a))+log(0.5);
g2_e = -0.5*([x1;x2]-nu_2c')'*inv(sigma_2c_a)*([x1;x2]-nu_2c')-0.5*log(det(sigma_1c_a))+log(0.5);

g_e=g1_e-g2_e; % boundary region equation
fimplicit(g_e);
title('Part A: Equation of Bayes  D. boundary for the equal prior probabilities');
legend('Gaussian Ran. Var. (g1)','Gaussian Ran. Var. (g2)','Decision B.');
hold off
```

Figure 1: The Equation of the Bayes Decision Algorithm for equal prior probabilities

First of all, μ_1, μ_2(mean) , Σ_1 and Σ_2(gaussian distribution) matrices were defined. So, deviation between mean and distributed gaussian was specified by Sigma and variance is determined by nu_1c and nu2_c. Then, a matrix defined positively is transformed into uppertriangular or lowertriangular by the Cholesky factorization method. After that, gaussion random variables were gotten by getting repeat copies arrays with the repmat function and adding random values with the randn function to it related to μ values. After that step, according to the general formula and the values given in the question, the values of g1_e and g2_e are calculated and subtracted from each other to get equation

of boundary region. Z1 and z2 random dots printed used plot function and graph of equation printed by using fimplicit function.

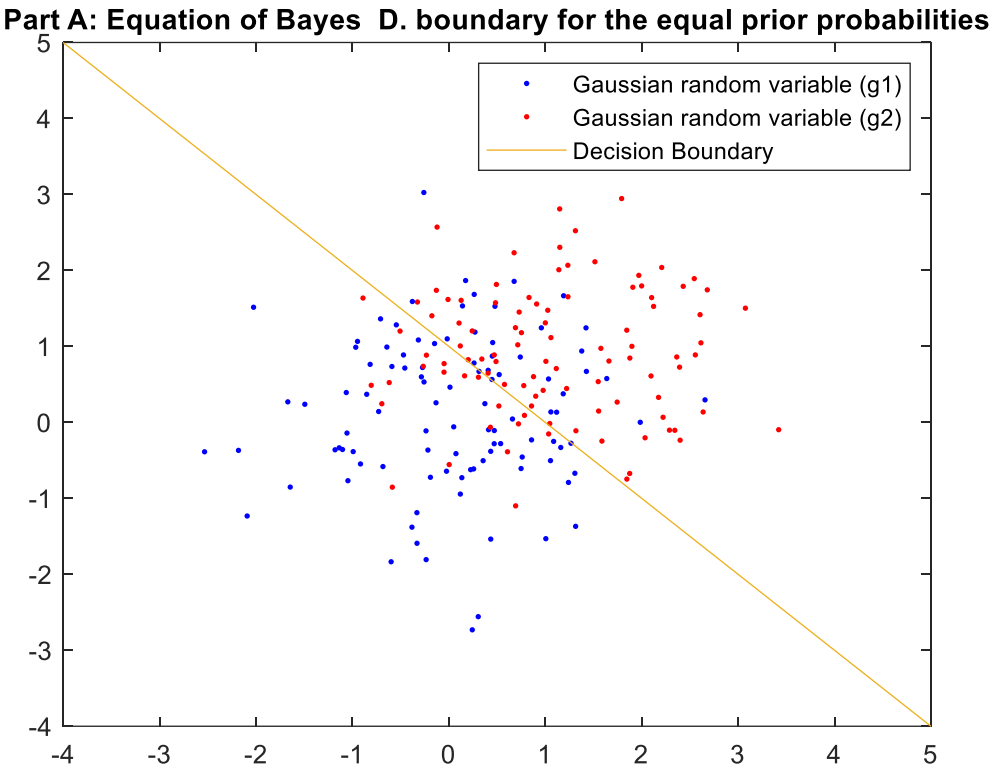

Figure 2: The Equation of the Bayes Decision Plot for equal prior probabilities

As a result of this plot, we can say we get similar result to calculated values.

Part B

$P(w_1)$ =1/4 and $P(w_2)$=3/4

For $G_1(x)$ ➔ $\mu_1=\begin{pmatrix}0\\0\end{pmatrix}$, $\Sigma_1=\begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}$ and $x=\begin{bmatrix}x1\\x2\end{bmatrix}$

$$G_i(X)= -\frac{1}{2} * (x-\mu i)^T * \sum_i^{-1}(x-\mu i) + lnP(wi) = -\frac{1}{2} * \begin{bmatrix}x1\\x2\end{bmatrix}^T * \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix} * \begin{bmatrix}x1\\x2\end{bmatrix} + lnP(w1)$$

$=(-1/2)* (x_1)^2 + (x_2)^2 + \ - \ln 2\pi - (1/2)*\ln(1) + \ln P\{W_1\}$

For $G_2(x)$ ➔ $\mu_1=\begin{pmatrix}0\\0\end{pmatrix}$, $\Sigma_1=\begin{pmatrix}1 & 0\\0 & 1\end{pmatrix}$ and $x=\begin{bmatrix}x1\\x2\end{bmatrix}$

$$G_2(x) = -\frac{1}{2} * \begin{bmatrix}x1-1\\x2-1\end{bmatrix}^T * \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix} * \begin{bmatrix}x1-1\\x2-1\end{bmatrix} - ln2\pi - \frac{1}{2}\ln\ |\Sigma 2| + lnP(w2) =$$

$-(1/2)* (x_1-1)^2 + (x_2-1)^2 - \ln 2\pi - (1/2)*\ln(1) + \ln P\{W_2\}$

Fınally, we can say that:

$$g1(x) = g2(x) \rightarrow x1 + x2 + 0.1 = 0$$

```
Pw1_1=1/4; Pw2_1=3/4;
nu1_b=[0 0]; nu2_b=[1 1];
sigma1_a=[1 0;0 1]; sigma2_b=[1 0;0 1];
figure
R_b = chol(sigma1_a);
z1_b = repmat(nu1_b,100,1) + randn(100,2)*R_b;
R_b = chol(sigma2_b);
z2_b = repmat(nu2_b,100,1) + randn(100,2)*R_b;
plot(z1_b(:,1),z1_b(:,2),'b.')
hold on;
plot(z2_b(:,1),z2_b(:,2),'r.')
syms x1 x2;
g1b_e = -0.5*([x1;x2]-nu1_b')'*inv(sigma1_a)*([x1;x2]-nu1_b')-0.5*log(det(sigma1_a))+log(Pw1_1);
g2b_e = -0.5*([x1;x2]-nu2_b')'*inv(sigma2_b)*([x1;x2]-nu2_b')-0.5*log(det(sigma1_a))+log(Pw2_1);
g_e1=g1b_e-g2b_e;
fimplicit(g_e1);
title('Part B: Equation of the Bayes d. boundary for P(w1) = 1/4 and P(w2) = 3/4');
legend('Gaussian random variable (g1)','Gaussian random variable (g2)','Decision Boundary');
hold off;
```

Figure 3: Equation of Bayes D. Algorithm for prior probabilities are P(w1) = 1/4 and P(w2) = ¾

Variable definitions were made as in Part A. In the same way, upper or lower triangular matrices were obtained using the chol function. By using the repmat function, repeat copies of array were obtained and random values were assigned with the randn function to the result and gaussian random values were obtained. Then, the function values g1b_e and g2b_e were calculated according to general expression related to sigma and P(w) values.

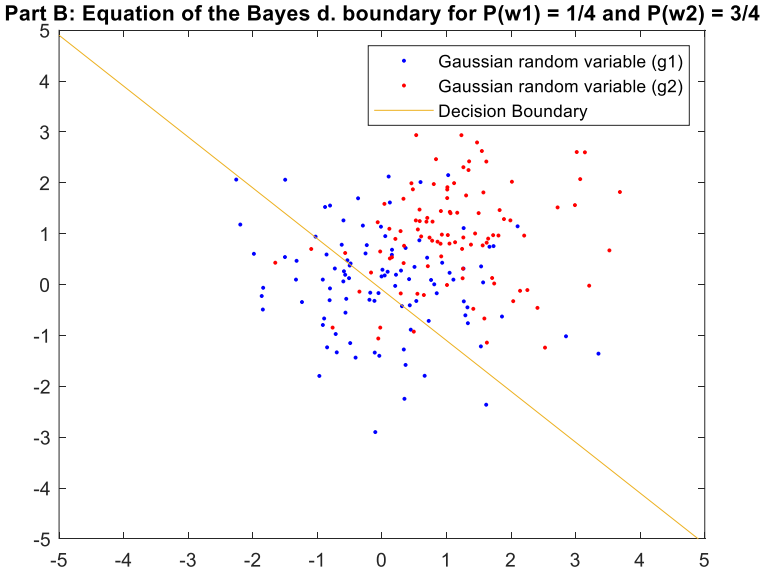

Figure 4: Equation of Bayes D. Plot for prior probabilities are P(w1) = 1/4 and P(w2) = ¾

As a result of this plot, we can say we get similar result to calculated values but our boundary is fewer different because of prior probabilities.

Part C

According to given new values: $\mu_1=\begin{pmatrix}0\\0\end{pmatrix}$, $\mu_2=\begin{pmatrix}1\\1\end{pmatrix}$, $\Sigma_1=\begin{pmatrix}2 & 0.5\\0.5 & 2\end{pmatrix}$, $\Sigma_2=\begin{pmatrix}5 & 4\\4 & 5\end{pmatrix}$.

$$g1(x) = -\frac{1}{2} * \begin{bmatrix}x1\\x2\end{bmatrix}^T * \begin{bmatrix}0.53 & -0.13\\-0.13 & 0.53\end{bmatrix} * \begin{bmatrix}x1\\x2\end{bmatrix} + lnP(w1) = -\frac{1}{2} *(0.53*(x_1^2+x_2^2)-0.26*x_1*x_2)+\ln P(w_1)$$

$$g2(x) = -\frac{1}{2} * \begin{bmatrix}x1-1\\x2-1\end{bmatrix}^T * \begin{bmatrix}0.55 & -0.44\\-0.44 & 0.55\end{bmatrix} * \begin{bmatrix}x1-1\\x2-1\end{bmatrix} +\ln P(W_2)$$

$$= -\frac{1}{2} * (0.55 * (x1^2 + x2^2) - 0.88x1x2 - 0.22(x1 + x2) + 0.42) + lnP(w2) - 1/2\ln(\Sigma 1)$$

For the equal prior probabilities;

$$g1(x) = g2(x) \rightarrow 0.02 * (x1^2 + x2^2) - 0.62x1x2 - 0.22(x1 + x2) + 1.27$$

```
nu_1c=[0 0]; nu_2c=[1 1];
sigma_1c_a=[2 0.5;0.5 2]; sigma_2c_a=[5 4;4 5];
figure
R = chol(sigma_1c_a);
z1 = repmat(nu_1c,100,1) + randn(100,2)*R;
R = chol(sigma_2c_a);
z2 = repmat(nu_2c,100,1) + randn(100,2)*R;
plot(z1(:,1),z1(:,2),'b.')
hold on;
plot(z2(:,1),z2(:,2),'r.')
syms x1 x2;
g1b_e = -0.5*([x1;x2]-nu_1c')'*inv(sigma_1c_a)*([x1;x2]-nu_1c')-0.5*log(det(sigma_1c_a))+log(0.5);
g2b_e = -0.5*([x1;x2]-nu_2c')'*inv(sigma_2c_a)*([x1;x2]-nu_2c')--0.5*log(det(sigma_1c_a))+log(0.5);
g_e=g1b_e-g2b_e;
fimplicit(g_e);
title(' Part C: Equation of the Bayes d.b for P(w1) = P(w2) for new values');
legend('Gaussian random variable (g1)','Gaussian random variable (g2)','Decision Boundary');
hold off;
```

Figure 5: Equation of Bayes D. Algorithm for equal prior probabilities and for new μ,Σ values

Variable definitions were made as in Part A and part B but we had new μ,Σ values . The chol function was used to create upper and lower triangular matrices in the same way. Repeat copies of the array were obtained using the repmat function, and random values were applied to the result using the randn function, yielding gaussian random values. Then, using a generic expression related to sigma and P(w) values, the function values g1be and g2be were calculated.

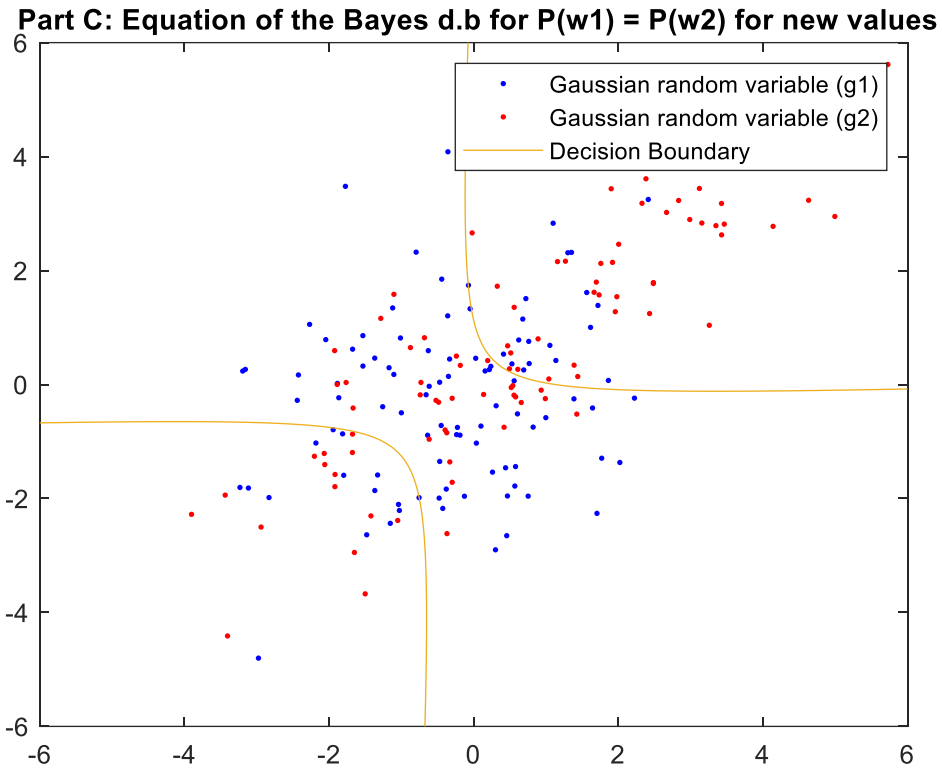

Figure 6: Equation of Bayes D. Plot for equal prior probabilities and for new μ,Σ values

According to graph, we can say our decision boundry is changed according to new μ,Σ values as we expected.

According to different prior probabilities:

P(w1) =1/4 P(w2)=3/4;

According to given new values: $\mu_1=\begin{pmatrix}0\\0\end{pmatrix}$, $\mu_2=\begin{pmatrix}1\\1\end{pmatrix}$, $\Sigma_1=\begin{pmatrix}2 & 0.5\\0.5 & 2\end{pmatrix}$, $\Sigma_2=\begin{pmatrix}5 & 4\\4 & 5\end{pmatrix}$.

$$g1(x) = -\frac{1}{2} * \begin{bmatrix}x1\\x2\end{bmatrix}^T * \begin{bmatrix}0.53 & -0.13\\-0.13 & 0.53\end{bmatrix} * \begin{bmatrix}x1\\x2\end{bmatrix} + lnP(w1) = -\frac{1}{2} * (0.53*(x_1^2+x_2^2)-0.26*x_1*x_2)+\ln P(w_1)$$

$$g2(x) = -\frac{1}{2} * \begin{bmatrix}x1-1\\x2-1\end{bmatrix}^T * \begin{bmatrix}0.55 & -0.44\\-0.44 & 0.55\end{bmatrix} * \begin{bmatrix}x1-1\\x2-1\end{bmatrix} + \ln P(W_2)$$

$$= -\frac{1}{2} * (0.55 * (x1^2 + x2^2) - 0.88x1x2 - 0.22(x1 + x2) + 0.42) + lnP(w2) - 1/2\ln(\Sigma 1)$$

$$g1(x) = g2(x) \rightarrow 0.02 * (x1^2 + x2^2) - 0.62x1x2 - 0.22(x1 + x2) - 3.2$$

```
Pw_3c_b=1/4; Pw_4c_b=3/4;
mu1_d=[0 0]; mu2_d=[1 1];
sigma1_3=[2 0.5;0.5 2]; sigma2_3=[5 4;4 5];
R_3 = chol(sigma1_3); R_2_3 = chol(sigma2_3);
z1_3 = repmat(mu1_d,100,1) + randn(100,2)*R_3;
z2_3 = repmat(mu2_d,100,1) + randn(100,2)*R_2_3;
figure
plot(z1_3(:,1),z1_3(:,2),'b.')
hold on;
plot(z2_3(:,1),z2_3(:,2),'r.')
syms x1 x2;
g13_e = -0.5*([x1;x2]-mu1_d')'*inv(sigma1_3)*([x1;x2]-mu1_d')-0.5*log(det(sigma1_3))-log(Pw_3c_b);
g23_e = -0.5*([x1;x2]-mu2_d')'*inv(sigma2_3)*([x1;x2]-mu2_d')-0.5*log(det(sigma1_3))-log(Pw_4c_b);
g_e_3=g13_e-g23_e; % equation of boundary region
fimplicit(g_e_3);
title(' Part C: Equation of the Bayes d.b for P(w1) = 1/4 anSd SP(w2) = 3/4 for new values')
legend('Gaussian random variable g1','Gaussian random variable g2','Decision Boundary');
hold off;
```

Figure 7: Equation of Bayes D. Algorithm for different prior probabilities and for new μ,Σ values

Variable definitions were made as in previous part with new μ,Σ values but in this part we had different prior probabilities . Similarly, the chol function was used to generate upper and lower triangular matrices. The repmat function was used to generate repeat copies of the array, and the randn function was used to add random values to the result, resulting in gaussian random values. The function values g13_e and g23_e were then determined using a general expression for sigma and P(w) values.

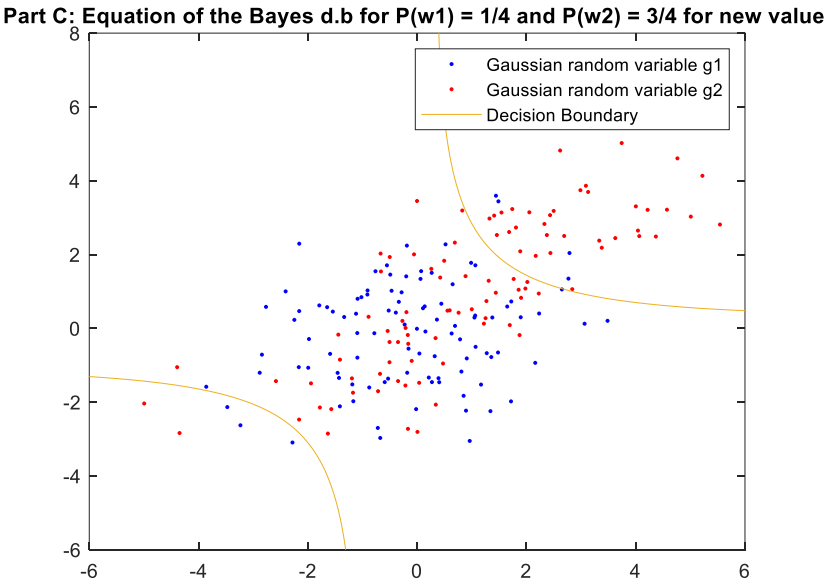

Figure 8: Equation of Bayes D. Algorithm for different prior probabilities and for new μ,Σ values

According to plotted graph, we may say our decision boundry is similar to previous one but it is different owing to different prior probabilities.

Part D

An elliptical shape can be gotten only if sigma values have a fold relationship.Therefore I ensured that create a set of possible values where the decision boundary is an ellipse.

```
Pw1=2/4; Pw2=2/4;
nu_1c=[0 0]; nu_2c=[1 1];
sigma_1c_a=[1 0;0 1]; sigma_2c_a=[3 0;0 3];
R = chol(sigma_1c_a); R = chol(sigma_2c_a);
z1 = repmat(nu_1c,100,1) + randn(100,2)*R;
z2 = repmat(nu_2c,100,1) + randn(100,2)*R;
figure
plot(z1(:,1),z1(:,2),'b.')
hold on;
plot(z2(:,1),z2(:,2),'r.')
syms x1 x2;
g1b_e = -0.5*([x1;x2]-nu_1c')'*inv(sigma_1c_a)*([x1;x2]-nu_1c')-0.5*log(det(sigma_1c_a))+log(Pw1);
g2b_e = -0.5*([x1;x2]-nu_2c')'*inv(sigma_2c_a)*([x1;x2]-nu_2c')-0.5*log(det(sigma_1c_a))+log(Pw2);
g_e=g1b_e-g2b_e; % equation of boundary region
fimplicit(g_e);
title('Part D Figure of possible set to get an ellipse');
legend('Gaussian random variable (g1)','Gaussian random variable (g2)','Decision Boundary');
hold off;
```

Figure 9: Algorithm of possible set to get an ellipse

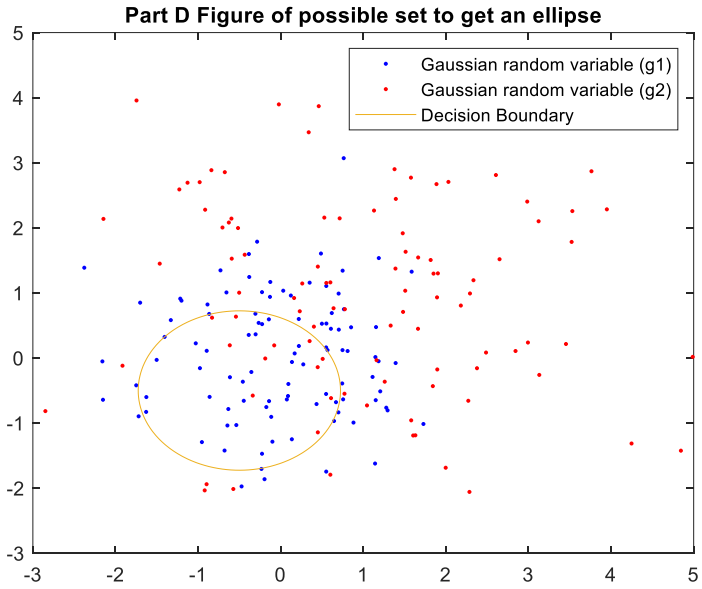

Figure 10: Plot of possible set to get an ellipse

According to plotted figure, we may claim that we get a elliptic decision boundary as a result of given sigma values.

Question 3

In this question, I studied on the Wisconsin Breast Cancer data set. Data set consist of 569 instances and it includes features which are extracted from both of benign and malign images cells.

```
wdbc.M = grp2idx(wdbc.M)-1;
info_matrix = table2array(wdbc);
classes = info_matrix(:,2);
features = info_matrix(:,3:end);
c = cvpartition(length(features),'KFold',10);
```

Figure 11: Doing preprocessing of data set.

First of all, data file imported using given steps on the procedure. Then, a data which belong to ID data is extracted from the dataset. After that, to show data fittingly, table is manipulated using table2array function. Then, data is divided into training and test sets using 10-fold cross validation scheme using cvpartition function.

```
index = zeros(1,length(info_matrix));
predicted = zeros(1,length(info_matrix));
data0 = features(classes == 0,:);
mu0 = mean(data0,1);
sigma0 = cov(data0);
data1 = features(classes == 1,:);
mu1 = mean(data1,1);
sigma1 = cov(data1);
```

Figure 12: Estimation of sigma and mean values

Because of data is divided into training and test sets using 10-fold cross validation scheme using cvpartition function, the test and training datasets are shown in then different folds. Mean and covariance values matrices may be find because we will process in folds with Gaussian Probability Density Function.

```
for i = 1:length(info_matrix)
    newsample = info_matrix(i,3:end);
    y0 = mvnpdf(newsample,mu0,0.2*sigma0);
    y1 = mvnpdf(newsample,mu1,sigma1);
    [~,idx] = max([y0 y1]);
    index(i) = info_matrix(i,2);
    predicted(i) = idx-1;
end
figure;
plotconfusion(index,predicted);

CM = zeros(2);
```

Figure 13: Algorithm for Gaussian Probability Density Function

After that process, assignation of each cell as a sick or healthy is done. Also, the covariance and mean values are estimated. The dataset used for a test. To do this, each element of the test samples is assigned to the variable x. The mvnpdf function is then used to have two different separate multivariate Gaussian probability density functions. A confusion matrix is constructed based on the information about which value is larger for that sample.

```
for i = 1:c.NumTestSets
    training = features(c.training(i),:);
    test = features(c.test(i),:);
    trueclass_labels = classes(c.test(i));
    train0 = training(classes(c.training(i))==0,:);
    train1 = training(classes(c.training(i))==1,:);
    var = 5;
    for j = 1:length(test)
        data = test(j,:);
        y0 = KDE(data,train0,var); y1 = KDE(data,train1,var);
        y_arr = [y0 y1];
        [~,idx] = max(y_arr);
        row = idx;
        column = trueclass_labels(j) + 1;
        CM(row,column) = CM(row,column) + 1;
    end
end
CM;
```

Figure 14: **Confusion Matrix for Kernel**

Probability distributions were estimated using Gaussian Kernel. It is determined using the MAP Rule.

```
function y=KDE(x,training,h)

n=size(training,1);

sum=0;

for k=1:n
    F=exp(-(x-training(k,:))*(x-training(k,:))'/(2*h^2));
    sum=sum+F;
end

sum=sum/(n*h);

y=sum;
end
```

Figure 15: KDE function

KDE function is written according to using that formula: $f(x)=\frac{1}{nh^d}\sum_{i=1}^{n}K(\frac{x-x_i}{h})$

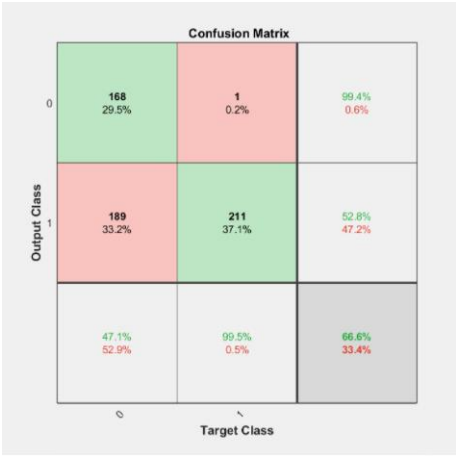

Figure 16: Confusion Matrix for Gaussian Distribution

I observed that confusion matrix and it shows Type errors and the output probabilities. When variance increases, Type II error decreases, while variance decreases Type I error increases.

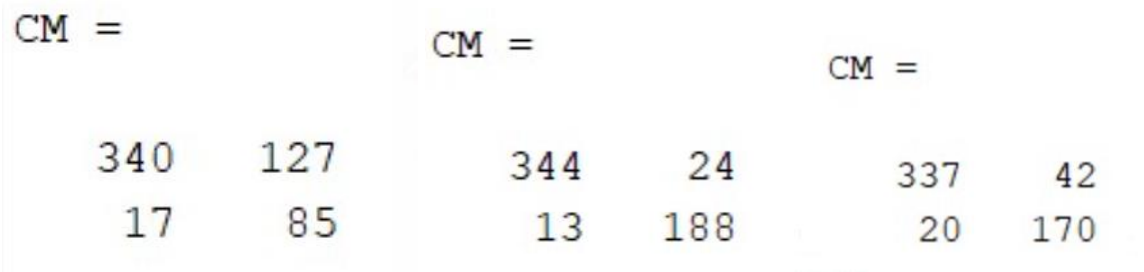

Figure 17: Comparing values

Variance value is set to 5 to get an optimal value for the confusion matrix. It is done using kernel function. The Type2 error proportion, (it is shown in the upper right corner of each part) drops if the variance value which belong to the kernel function is increased. The Type1 error rate in the lower left corner reduces as the variance value decreases. Therefore we can say, Type1 error and TypeII error inversely proportional

3. CONCLUSION

We had an 3 different problems in this assignment.

In first part, we made a decision about building solar power plant on the unknown planet. First of all, we try to find probability of sun will rise in each day using prior probabilities and total probability principle, Bayes' Theorem and conditional probability. Then, probability of error of decision is estimated. Finally, I decided to 13 days will be enough to decide for making investment.

In second problem, I observed two category classification problem in 2 dimensions. I did classification process according to different mean and standard deviation values and prior probabilities. I observed how decision region may change due to given values. In the final step, I get an elliptic boundary setting sigma values with a fold relationship.

In third problem, I studied with the Wisconsin Breast Cancer data set. First of all, I imported data set to Matlab and I divide it into test and training set using cvpartition(). Then, MAP decision rule is appiled to classify the cell as a sick or not. Then, probabilitiy distributions were determined by using Kernel density estimation with Gaussian Kernel. Also, confusion matrix is determined. Then, Type I and Type II errors are defined. Type I error is accepted as a falsely diagnoses a healthy cell while Type II error is accepted as a diseased cell is falsely classified as healthy. Then, a method is offered to reduce Type II since it is more dangerous.

REFERENCES

[1] Howie, David. (2002). Interpreting probability: controversies and developments in the early twentieth century. Cambridge University Press. pp. 24.