M3S7-Statistical Pattern Recognition
Project 1

Alexandre Elkrief
CID: 00732974

February 16, 2015

1 Equal priors classification problem

Consider a two-class classification problem, with priors $P(C_1) = 0.5$ and $P(C_2) = 0.5$ and class conditional densities given by

$$p(x|C_i) = f_i(x; \sigma_i) = \frac{x}{\sigma_i^2} exp(-\frac{x^2}{2\sigma_i^2}), \qquad 0 \leq \ x < \infty$$

for $i = 1, 2$ where $\sigma_1 > \sigma_2 > 0$.

(a) Express the decision threshold $T_{min} > 0$ for minimum error as a function of $\sigma_1$and σ_2.

$$\frac{p(x|C_1)}{p(x|C_2)} = \frac{p(C_2)}{p(C_1)}$$

$$\Leftrightarrow \frac{\frac{x}{\sigma_1^2} exp(\frac{-x^2}{2\sigma_1^2})}{\frac{x}{\sigma_2^2} exp(\frac{-x^2}{2\sigma_2^2})} = 1$$

$$\Leftrightarrow \frac{\sigma_2^2}{\sigma_1^2} exp(\frac{-x^2}{2\sigma_1^2} + \frac{x^2}{2\sigma_2^2}) = 1$$

$$\Leftrightarrow exp(\frac{-x^2}{2\sigma_1^2} + \frac{x^2}{2\sigma_2^2}) = \frac{\sigma_1^2}{\sigma_2^2}$$

$$\Leftrightarrow \frac{-x^2}{2\sigma_1^2} + \frac{x^2}{2\sigma_2^2} = 2\log(\frac{\sigma_1}{\sigma_2})$$

$$\Leftrightarrow x^2 \frac{\sigma_1^2 - \sigma_2^2}{2\sigma_1^2\sigma_2^2} = 2\log(\frac{\sigma_1}{\sigma_2})$$

$$\Leftrightarrow x^2 = \frac{4\sigma_1^2\sigma_2^2}{\sigma_1^2 - \sigma_2^2} \log(\frac{\sigma_1}{\sigma_2})$$

$$\Rightarrow x = \pm 2\sigma_1\sigma_2 \sqrt{\frac{\log(\frac{\sigma_1}{\sigma_2})}{\sigma_1^2 - \sigma_2^2}}$$

$$\Rightarrow T_{min} = 2\sigma_1\sigma_2 \sqrt{\frac{\log(\frac{\sigma_1}{\sigma_2})}{\sigma_1^2 - \sigma_2^2}}$$

(b) In terms of F_i, the cumulative distribution functions of the class conditional densities, obtain an expression for the error rate associated with some decision threshold, $T \geq 0$. For σ_1, and three separate cases of (i) $\sigma_2 = 0.5$, (ii) $\sigma_2 = 1$, and (iii) $\sigma_2 = 1.5$, plot the error rate as a function of T for $0 \leq T \leq 4$ using the same axes for all three plots.

$$e = p(C_1)p(x|C_1 \leq T)p(C_2)p(x|C_2 \geq T)$$
$$\Rightarrow e = 0.5[F_1(T) + (1 - F_2(T))]$$

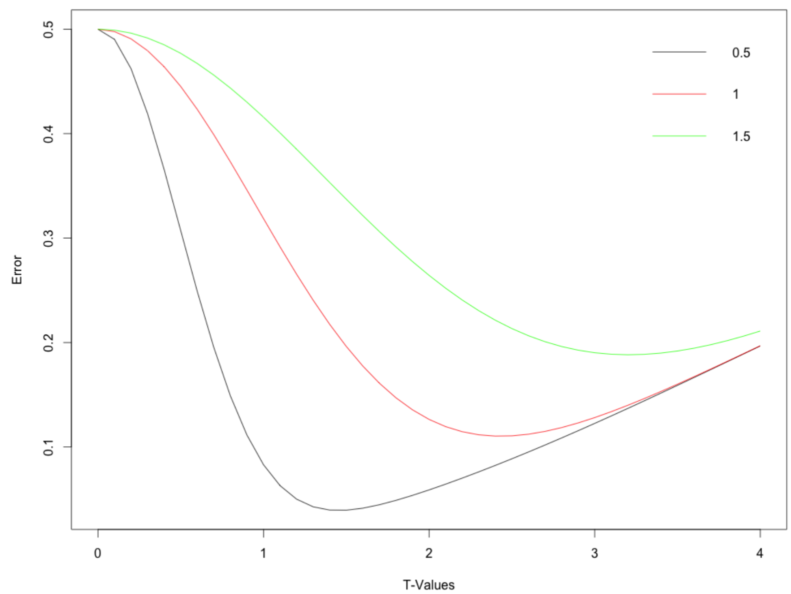


Figure 1: Error Rate

(c) Give an expression for the Bayes error rate for general σ_1 and σ_2. Your answer should be given in terms of r $= \sigma_1/\sigma_2$.

$$F(u|C_i) = \int_0^u \frac{x}{\sigma_i^2} exp(-\frac{x^2}{2\sigma_i^2})dx$$
$$\Leftrightarrow F(u|C_i) = -[\exp(-\frac{x^2}{2\sigma_i^2})]_0^u$$
$$\Rightarrow F(u|C_i) = 1 - \exp(-\frac{u^2}{2\sigma_i^2})$$

Therefore using the expression for the error rate from (b) we get the following:

$$e_B = 0.5\Bigg[1 - \exp\Bigg(-\frac{t_{min}^2}{2\sigma_1^2}\Bigg) + \exp\Bigg(-\frac{t_{min}^2}{2\sigma_2^2}\Bigg)\Bigg]$$

Plugging in $T_{min} = 2\sigma_1\sigma_2\sqrt{\frac{log(\frac{\sigma_1}{\sigma_2})}{\sigma_1^2 - \sigma_2^2}}$ yields:

$$e_B = 0.5\Bigg[1 - \exp\Bigg(-\frac{2\sigma_2^2 log(\frac{\sigma_1}{\sigma_2})}{\sigma_1^2 - \sigma_2^2}\Bigg) + \exp\Bigg(-\frac{2\sigma_1^2 log(\frac{\sigma_1}{\sigma_2})}{\sigma_1^2 - \sigma_2^2}\Bigg)\Bigg]$$

$$\Leftrightarrow e_B = 0.5\Bigg[1 - \exp\Bigg(-\frac{2log(r)}{r^2 - 1}\Bigg) + \exp\Bigg(-\frac{2log(r)}{1 - r^{-2}}\Bigg)\Bigg]$$

(d) For $\sigma_1 = 4$ and $\sigma_2 = 1$, compute the value of the minimum error threshold and the Bayes error rate to 2 decimal places. Construct a plot of the prior weighted class conditional densities and construct a plot of the posterior densities. Include a marker for the minimum error decision threshold in both plots.

Using the formula for T_{min} developed in (a) and the error rate from (c) we get the following:

$$T_{min} = 2.43$$

$$e_B = 0.11$$

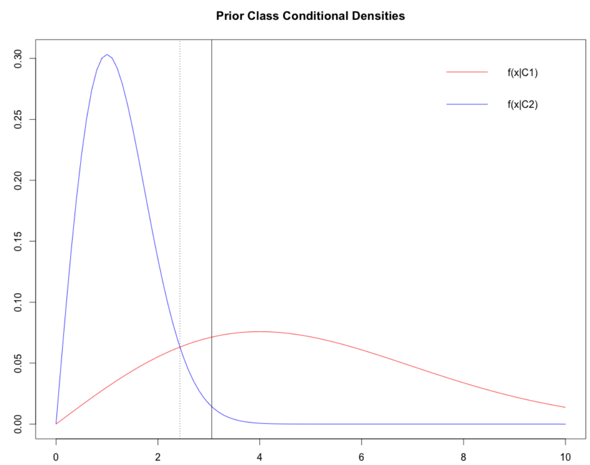


Figure 2: Prior Class Conditional Densities

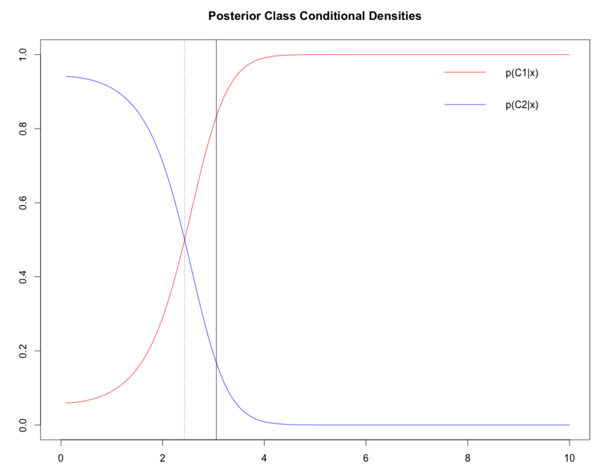


Figure 3: Posterior Densities

(e) Suppose we now associate the following costs with class allocation:

	C_1	C_2
C_1	0	5
C_2	1	0

For general $\sigma_1 > \sigma_2 > 0$ express in terms of σ_1 and σ_2 the Bayes decision threshold for minimum risk. Using the values of $\sigma_1 = 4$ and $\sigma_2 = 1$, compute this threshold and place a marker for it in both the plots you made in part (d).

$$\lambda_{21} p(C_1|x) \leq \lambda_{12} p(C_2|x)$$
$$\Leftrightarrow \frac{p(C_1|x)}{p(C_2|x)} \geq \frac{\lambda_{21}}{\lambda_{12}}$$

The decision threshold corresponds to the equality:

$$\Rightarrow \frac{p(C_1|x)}{p(C_2|x)} = 5$$
$$\Leftrightarrow \frac{p(C_1)p(x|C_1)}{p(C_2)p(x|C_2)} = 5$$
$$\Leftrightarrow \frac{p(x|C_1)}{p(x|C_2)} = 5$$
$$\Leftrightarrow \frac{\frac{x}{\sigma_1^2} exp(\frac{-x^2}{2\sigma_1^2})}{\frac{x}{\sigma_2^2} exp(\frac{-x^2}{2\sigma_2^2})} = 5$$
$$\Leftrightarrow exp(\frac{-x^2}{2\sigma_1^2} + \frac{x^2}{2\sigma_2^2}) = 5\frac{\sigma_1^2}{\sigma_2^2}$$
$$\Leftrightarrow x^2 \frac{\sigma_1^2 - \sigma_2^2}{2\sigma_1^2\sigma_2^2} = log(5\frac{\sigma_1^2}{\sigma_2^2})$$
$$\Leftrightarrow x^2 = \frac{2\sigma_1^2\sigma_2^2}{\sigma_1^2 - \sigma_2^2} log(5\frac{\sigma_1^2}{\sigma_2^2})$$
$$\Rightarrow x = \pm\sigma_1\sigma_2\sqrt{\frac{2log(5\frac{\sigma_1^2}{\sigma_2^2})}{\sigma_1^2 - \sigma_2^2}}$$
$$\Rightarrow T_{min} = \sigma_1\sigma_2\sqrt{\frac{2log(5\frac{\sigma_1^2}{\sigma_2^2})}{\sigma_1^2 - \sigma_2^2}}$$

For $\sigma_1 = 4,\ \sigma_2 = 1,\ \Rightarrow T_{min} = 3.06$

2 Normal Classification Problem

Consider a two class classification problem with equal priors and bivariate normal class conditional densities

$$p(x|C_1) \sim N\left(\begin{pmatrix}-1\\-1\end{pmatrix}, \mathbf{\Sigma}_1\right) \quad \text{and} \quad p(x|C_2) \sim N\left(\begin{pmatrix}1\\1\end{pmatrix}, \mathbf{\Sigma}_2\right)$$

where

$$\mathbf{\Sigma}_1 = \begin{pmatrix}1 & 0.2\\0.2 & 1\end{pmatrix} \quad \text{and} \quad \mathbf{\Sigma}_2 = \begin{pmatrix}1 & -0.7\\-0.7 & 1\end{pmatrix}$$

(a) Sample a random vector, **x***, uniformly from the rectangular region (-6, 6) x (-6, 6). Compute an appropriate discriminant score, $g_i(x)$, for **x*** for C_i, i = 1,2. On the basis of these discriminant scores, to which class should **x*** be assigned?

As the two covariance matrices $\mathbf{\Sigma}_1$,$\mathbf{\Sigma}_1$ are arbitrary, we compute the discriminant functions in the following way:

$$g_i(\mathbf{x}) = \mathbf{x}^t \mathbf{W}_i \mathbf{x} + \mathbf{w}_i^t \mathbf{x} + w_{i0}$$

where

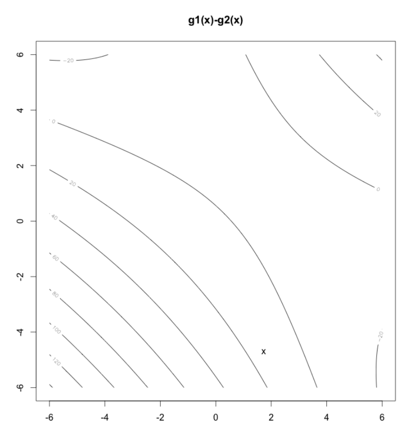
$$\mathbf{W}_i = -\frac{1}{2}\mathbf{\Sigma}_i^{-1} \quad \text{and} \quad \mathbf{w}_i = \mathbf{\Sigma}_i^{-1}\boldsymbol{\mu}_i$$

and

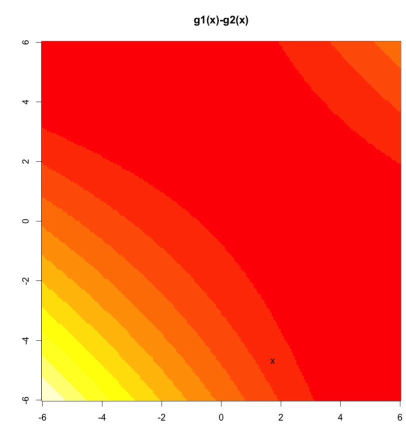
$$w_{i0} = -\frac{1}{2}\boldsymbol{\mu}_i^t \mathbf{\Sigma}_i^{-1}\boldsymbol{\mu}_i - \frac{1}{2}\ln|\mathbf{\Sigma}_i^{-1}| + \ln p(c_i)$$

Sampling from the uniform distribution, we get **x*** =(1.73, -4.70).
Plugging this point in the discriminant equation yields: $g_1(\mathbf{x*}) = -13.77$ and $g_2(\mathbf{x*}) = -26.99$
On the basis of these discriminant scores, we assign **x*** to the class yielding the highest score, namely C_1.

(b) Construct a plot that displays $g_1(\mathbf{x}) - g_2(\mathbf{x})$ evaluated over a regular grid on $(-6, 6) \times (6, 6)$. Add a marker for **x*** to the plot.



(a) Contour Plot



(b) Heat Plot

3 2-Class p-dimensional probelm

Consider a 2-class p-dimensional problem (p will be ≥ 10 in this question) with $P(C_1) = 0.25$ and $P(C_2) = 0.75$, with multivariate normal class conditional densities. Mean vector $\boldsymbol{\mu}_1$, is a p-dimensional vector of zeros, and $\boldsymbol{\mu}_2$ is yet unspecified. Covariance matrix $\mathbf{\Sigma}_1 = \mathbf{I}_p$, where $\mathbf{I}_p$ is the p dimensional identity matrix. The diagonal elements of $\mathbf{\Sigma}_2$ are all equal to 0.5, and all off-diagonal elements are zero, except for $\sigma_{13} = \sigma_{31} = 0.3$, $\sigma_{23} = \sigma_{32} = 0.2$ and $\sigma_{48} = \sigma_{84} = 0.1$.

(a) Select an integer uniformly from the set $\{10, 11, 12, 13, 14, 15\}$ for p.

(b) Select a random vector of length p uniformly from the region defined by $(0,1)^p$, for $\boldsymbol{\mu}_2$.

(c) For $n = 100$, 200, 300, 400, 500, 600, 700, 800, 900 and 1000 do the following;

i. Obtain a training sample of n observations, and a test sample of 4000 observations, in the same ratio as the prior probabilities.

ii. Compute the out-of-sample empirical error rate, using the test set, for linear and quadratic discriminant functions estimated from the training data.

(d) On the same set of axes plot the training set size on the x-axis and the out-of-sample empirical error rate on the y-axis for both your linear discriminant analysis and your quadratic discriminant analysis.

(e) Comment on your results.

(f) Compute an estimate of the Bayes error rate.

Answer:
For questions (a),(b),(c), see R-code attached.

(d) The following plot corresponds to $p = 12$ and $\boldsymbol{\mu}_2 = (0.73, 0.15, 0.24, 0.17, 0.06, 0.67, 0.61, 0.77, 0.91, 0.23, 0.69, 0.85)$

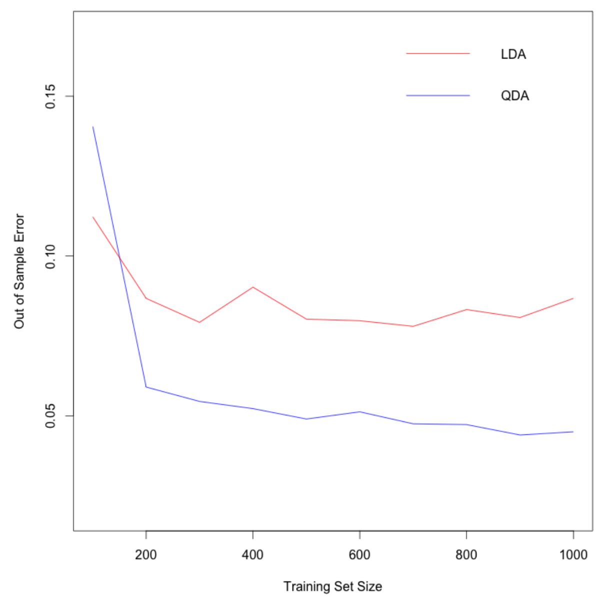


Figure 5: Out-of-sample empirical error rate

(e) For a small set of training data the LDA appears to perform better than the QDA. This poor performance of the QDA can be explained by the problem of overfitting. Indeed, for a p-dimensional covariance matrix the QDA has to estimate $\frac{p(p+1)}{2}$ elements. In our case with $p \in \{10, 11, 12, 13, 14, 15\}$ that can lead to the approximation of up to 120 elements (for $p = 15$) with a training data set of 100 elements, therefore causing the model to overfit. On the contrary, as the size of the training set increases the performance of the QDA improves significantly and clearly performs better than the LDA.

(f) As the error rate seems to converge to some irreducible error rate, I have tried computing the error rate for a training size of 100'000 elements with a test set of 400'000. The R-code attached uses a test set of 15'000 to limit the computation time while the error rate shows no significant difference. Using different values for p, I got a Bayes error rate ranging from 0.27 to 0.65.

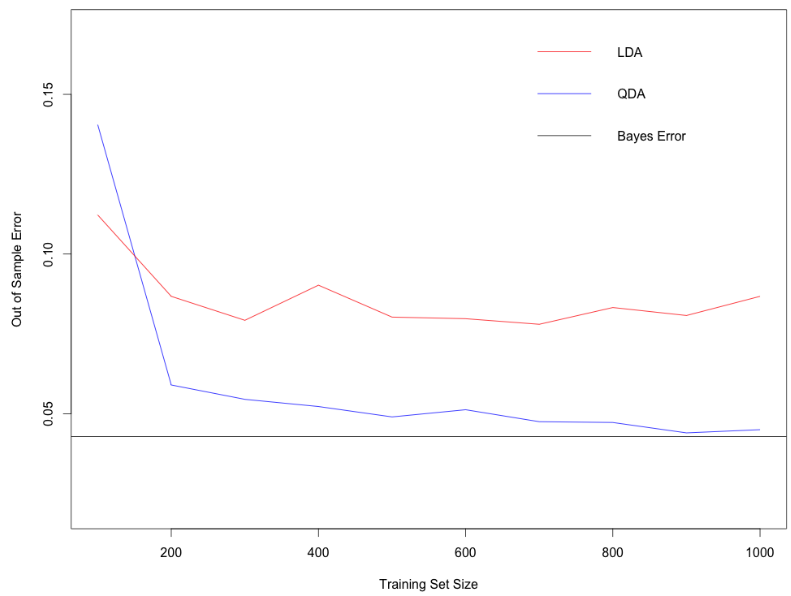


Figure 6: Out-of-sample empirical error rate

4 Golden ratio bracketing algorithm

Suppose we are interested in finding a minimum of a function $f : \mathbf{R} \mapsto \mathbf{R}$ on an interval (a,b). Write an R function that uses the *Golden ratio bracketing algorithm* to determine the location of the minimum. The bracketing function should be written to take a target function f as an argument.
Test the R function using

$$f(x) = \exp(x)\sin(x) + \sinh(x)$$

to find the minimum in the interval $(4, 0)$.

Answer:
See R-code in Appendix

5 Appendix

```
#Question 1
#Q1b
tvals<-seq(0,4,by=0.1)

s1<-rep(0,length(tvals))
s2<-rep(0,length(tvals))
s3<-rep(0,length(tvals))
for (i in 1:length(tvals)){
  s1[i]<-0.5*(1-exp(-(tvals[i]^2)/32)+exp(-(tvals[i]^2)/(2*0.5^2)))
  s2[i]<-0.5*(1-exp(-(tvals[i]^2)/32)+exp(-(tvals[i]^2)/(2*1^2)))
  s3[i]<-0.5*(1-exp(-(tvals[i]^2)/32)+exp(-(tvals[i]^2)/(2*1.5^2)))
}
sigma<-c(0.5,1,1.5)

#plot Q1b error rate
plot(tvals,s1,type="l",xlab = "T-Values",ylab = "Error")
lines(tvals,s2,type="l",col="red")
lines(tvals,s3,type="l",col="green")
legend("topright",legend=sigma,col=c("black","red","green"),lty=1,bty='n')


#Q1d
#Tmin for sigma=4& 1
Q1d<-2*4*1*(log(4/1)/(4^2-1^2))^0.5
Q1d
#Bayes error rate
eb<-0.5*(1-exp(-(Q1d^2)/32)+exp(-(Q1d^2)/(2*1^2)))
eb
#Q1e cost allocation new threshold
Q1e<-4*1*(2*log(5*(16/1))/(4^2-1^2))^0.5
Q1e

#Q1d-prior distribution
tvals<-seq(0,10,by=0.1)
f1<-rep(0,length(tvals))
f2<-rep(0,length(tvals))
for (i in 1:length(tvals)){
  f1[i]<-0.5*(tvals[i]/4^2)*exp(-tvals[i]^2/(2*4^2))
  f2[i]<-0.5*(tvals[i]/1^2)*exp(-tvals[i]^2/(2*1^2))
}
plot(tvals,f2,type="l",xlab="",ylab="",main="Prior Class Conditional Densities", col="blue")
lines(tvals,f1,type="l",col="red")
abline(v=Q1d,lty="dotted")
abline(v=Q1e)
legend("topright",legend=c("f(x|C1)","f(x|C2)"),col=c("red","blue"),lty=1,bty='n')


#Posterior distribution
tvals<-seq(0,10,by=0.1)
post1<-rep(0,length(tvals))
post2<-rep(0,length(tvals))
for (i in 1:length(tvals)){
  post1[i]<-(tvals[i]/4^2)*exp(-tvals[i]^2/(2*4^2))/((tvals[i]/4^2)*exp(-tvals[i]^2/(2*4^2))+(tvals
```

```
      [i]/1^2)*exp(-tvals[i]^2/(2*1^2)))
  post2[i]<-(tvals[i]/1^2)*exp(-tvals[i]^2/(2*1^2))/((tvals[i]/4^2)*exp(-tvals[i]^2/(2*4^2))+(tvals
      [i]/1^2)*exp(-tvals[i]^2/(2*1^2)))

}

plot(tvals,post1,type="l",xlab="",ylab="",ylim=c(0,1),col="red",main="Posterior Class Conditional
    Densities")
lines(tvals,post2,type="l",col="blue")
abline(v=Q1d,lty="dotted")
abline(v=Q1e,)
legend("topright",legend=c("p(C1|x)","p(C2|x)"),col=c("red","blue"),lty=1,bty='n')

#Question 2
#Q2a
#Random choice of x*
xstar <- runif(2,-6,6)
mu1 <- c(-1,-1)
mu2 <- c(1,1)
cov1<-matrix(c(1,0.2,0.2,1),nrow=2,ncol=2)
covinv1<-solve(cov1)
cov2<-matrix(c(1,-0.7,-0.7,1),nrow=2,ncol=2)
covinv2<-solve(cov2)

#Discriminant Function
discr<- function(x, mu1 = c(-1, -1), mu2 = c(1, 1))
{
  W1<-(-0.5)*covinv1
  W2<-(-0.5)*covinv2

  w1<-covinv1%*%mu1
  w2<-covinv2%*%mu2

  w10<-(-0.5)*t(mu1)%*%covinv1%*%mu1-0.5*log(abs(det(cov1)))+log(0.5)
  w20<-(-0.5)*t(mu2)%*%covinv2%*%mu2-0.5*log(abs(det(cov2)))+log(0.5)

  g1<-t(x)%*%W1%*%x+t(w1)%*%x+w10
  g2<-t(x)%*%W2%*%x+t(w2)%*%x+w20
  g1-g2
}


l <- 200
x <- seq(-6,6,length=l)
y <- x
gr <- as.matrix(expand.grid(x,y))
score.diff <-apply(gr,1,discr)
par(pty="s")
#Contour Plot
contour(x,y,matrix(score.diff,l),main="g1(x)-g2(x)")
text(xstar[1],xstar[2],"x")
#Heat Plot
image(x,y,matrix(score.diff,l),xlab = "",ylab = "",main="g1(x)-g2(x)")
text(xstar[1],xstar[2],"x")
```

```
#Question 3
#Q3a-b
p<-floor(runif(1,0,6))+10

mu1<-rep(0,p)
mu2<-runif(p)

cov1<-diag(p)
cov2<-0.5*diag(p)
cov2[1,3]<--0.3
cov2[3,1]<--0.3
cov2[2,3]<-0.2
cov2[3,2]<-0.2
cov2[4,8]<-0.1
cov2[8,4]<-0.1


#Q3c

emperror.lda<-rep(0,10)
emperror.qda<-rep(0,10)


for (i in 1:10){
  trainC1<-mvrnorm(0.25*i*100,mu1,cov1)
  trainC2<-mvrnorm(0.75*i*100,mu2,cov2)
  traindata<-rbind(trainC1,trainC2)

  trainclassC1<-rep(1,0.25*i*100)
  trainclassC2<-rep(2,0.75*i*100)
  trainclass<-c(trainclassC1,trainclassC2)


  testC1<-mvrnorm(1000,mu1,cov1)
  testC2<-mvrnorm(3000,mu2,cov2)
  testdata<-rbind(testC1,testC2)

  testclassC1<-rep(1,1000)
  testclassC2<-rep(2,3000)
  testclass<-c(testclassC1,testclassC2)


  #qda
  train.qda<-qda(traindata,trainclass)
  prediction.qda<-predict(train.qda,testdata)
  table.qda<-table(prediction.qda$class,testclass)
  emperror.qda[i]<-(table.qda[1,2]+table.qda[2,1])/4000

  #lda
  train.lda<-lda(traindata,trainclass)
  prediction.lda<-predict(train.lda,testdata)
  table.lda<-table(prediction.lda$class,testclass)
  emperror.lda[i]<-(table.lda[1,2]+table.lda[2,1])/4000
}
emperror.lda
emperror.qda
```

```
#Q3d
plot(seq(100,1000,100),emperror.qda,type="l",xlab = "Training Set Size",ylab = "Out of Sample Error
    ",col="blue",ylim=c(0.02,max(emperror.qda)+0.03))
lines(seq(100,1000,100),emperror.lda,col="red")
legend("topright", legend=c("LDA","QDA","Bayes Error"),col=c("red","blue","black"),lty=1,bty='n')

#Q3f

bayes.emperror.qda<-rep(0,10)

for (i in 1:10){
  trainC1<-mvrnorm(0.25*15000,mu1,cov1)
  trainC2<-mvrnorm(0.75*15000,mu2,cov2)
  traindata<-rbind(trainC1,trainC2)

  trainclassC1<-rep(1,0.25*15000)
  trainclassC2<-rep(2,0.75*15000)
  trainclass<-c(trainclassC1,trainclassC2)


  testC1<-mvrnorm(15000,mu1,cov1)
  testC2<-mvrnorm(45000,mu2,cov2)
  testdata<-rbind(testC1,testC2)

  testclassC1<-rep(1,15000)
  testclassC2<-rep(2,45000)
  testclass<-c(testclassC1,testclassC2)

  length(testclass)

  #qda
  train.qda<-qda(traindata,trainclass)
  prediction.qda<-predict(train.qda,testdata)
  table.qda<-table(prediction.qda$class,testclass)
  bayes.emperror.qda[i]<-(table.qda[1,2]+table.qda[2,1])/60000
}
bayes.emperror.qda
#Taking the average error of the 10 trials
average.bayes.error<-mean(bayes.emperror.qda)
average.bayes.error
#Adding the Bayes error on the plot
abline(h=average.bayes.error)

#Question 4
#defining the function for which we find the minimum
f<-function(x){
  f.x=exp(-x)*sin(x) + sinh(x)
  return(f.x)

}
#setting the limiting values
x1<--4
x3<-0
eps<-0.000000001
```

```
grbrack<-function(f,x1,x3,eps){
  ratio<-(1+sqrt(5))/2
  x2<-x1+(x3-x1)/ratio
  x2
  i=0

  while ((x3-x1)>eps){
    i=i+1

    f.x1=f(x1)
    f.x2=f(x2)
    f.x3=f(x3)
    if (f.x2<f.x1 && f.x2<f.x3){
      x3<-x2
      x2<-x1+(x3-x1)/ratio
    }
    else{
      x1<-x2
      x2<-x1+(x3-x1)/ratio
    }
  }
  return(f.x2)
}
grbrack(f,x1,x3,eps)
```