ECEN 649 Pattern Recognition – Spring 2016

Computer Project 1

Due on: Mar 4

Consider the following Gaussian models, where the dimension is d, the prior class probabilities are $P(Y=0) = P(Y=1) = \frac{1}{2}$, the means are at the points $\boldsymbol{\mu}_0 = (0, \ldots, 0)$ and $\boldsymbol{\mu}_1 = (1, \ldots, 1)$.

Model M1: Independent variables, $\Sigma_0 = \Sigma_1 = \sigma^2 I_d$.

Model M2: Positively correlated variables in blocks of 2,

$$\Sigma_0 \;=\; \Sigma_1 \;=\; \sigma^2 \begin{pmatrix} 1 & \rho & 0 & 0 & \cdots & 0 & 0 \\ \rho & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \rho & \cdots & 0 & 0 \\ 0 & 0 & \rho & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & \rho \\ 0 & 0 & 0 & 0 & \cdots & \rho & 1 \end{pmatrix}_{d \times d}$$

where d is assumed to be a multiple of 2, and $0 \leq \rho \leq 1$.

Model M3: Positively correlated variables in blocks of 3,

$$\Sigma_0 \;=\; \Sigma_1 \;=\; \sigma^2 \begin{pmatrix} 1 & \rho & \rho & \cdots & 0 & 0 & 0 \\ \rho & 1 & \rho & \cdots & 0 & 0 & 0 \\ \rho & \rho & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & \rho & \rho \\ 0 & 0 & 0 & \cdots & \rho & 1 & \rho \\ 0 & 0 & 0 & \cdots & \rho & \rho & 1 \end{pmatrix}_{d \times d}$$

where d is assumed to be a multiple of 3, and $0 \leq \rho \leq 1$.

Note that these are homoskedastic Gaussian models (so that the optimal decision boundary is a hyperplane in each case), but with increasingly correlated features.

Problem 1: Show that the Bayes errors for each of these models are given by:

$$\varepsilon^*_{M1} \;=\; \Phi\left(-\frac{\sqrt{d}}{2\sigma}\right), \quad \varepsilon^*_{M2} \;=\; \Phi\left(-\frac{\sqrt{d}}{2\sigma}\frac{1}{\sqrt{1+\rho}}\right), \quad \text{and } \varepsilon^*_{M3} \;=\; \Phi\left(-\frac{\sqrt{d}}{2\sigma}\frac{1}{\sqrt{1+2\rho}}\right),$$

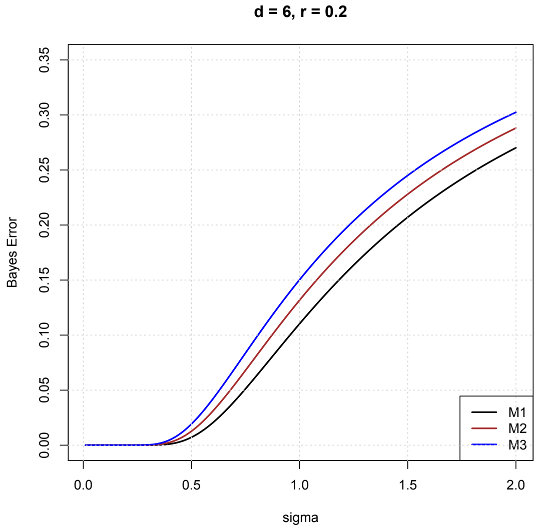
where $\Phi(x) = (2\pi)^{-1/2}\int_{-\infty}^{x} e^{-x^2/2}dx$ is the CDF of a standard normal random variable. Argue that $\varepsilon^*_{M1} < \varepsilon^*_{M2} < \varepsilon^*_{M3}$, and that the difference becomes larger as ρ increases. Give a justification of this. For $d = 6$, and $\rho = 0.2, 0.8$, plot on the same axes the optimal errors for the three models as a function of the standard deviation σ. How do you interpret the results in terms of the correlation between the features? Hint: Use the expression for the Bayes error in the homoskedastic Gaussian case, given in class.

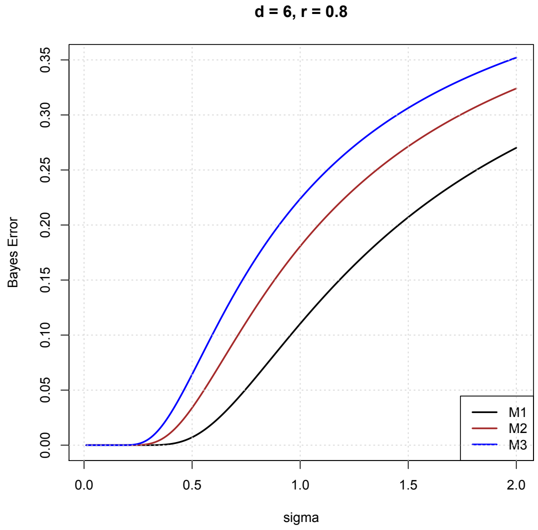
Solution: The Bayes error in each case is given by

$$\varepsilon^* = \Phi\left(-\frac{\delta}{2}\right) \tag{1}$$

where $\delta = \sqrt{(\mu_1 - \mu_0)^T \Sigma^{-1} (\mu_1 - \mu_0)}$. By using the fact that the inverse of a block matrix $M = \text{diag}(M_1, M_2, \ldots, M_N)$ is another block matrix $M^{-1} = \text{diag}(M_1^{-1}, M_2^{-1}, \ldots, M_N^{-1})$, it is easy to compute Σ^{-1}, and thus δ, for each model. The desired result follows by plugging δ in (1). It is clear that $\varepsilon^*_{M1} < \varepsilon^*_{M2} < \varepsilon^*_{M3}$, since the argument of $\Phi(\cdot)$ increases from one model to the next.

Below are plots of the Bayes error for $d = 6$ and $\rho = 0.2, 0.8$, for each of the models.





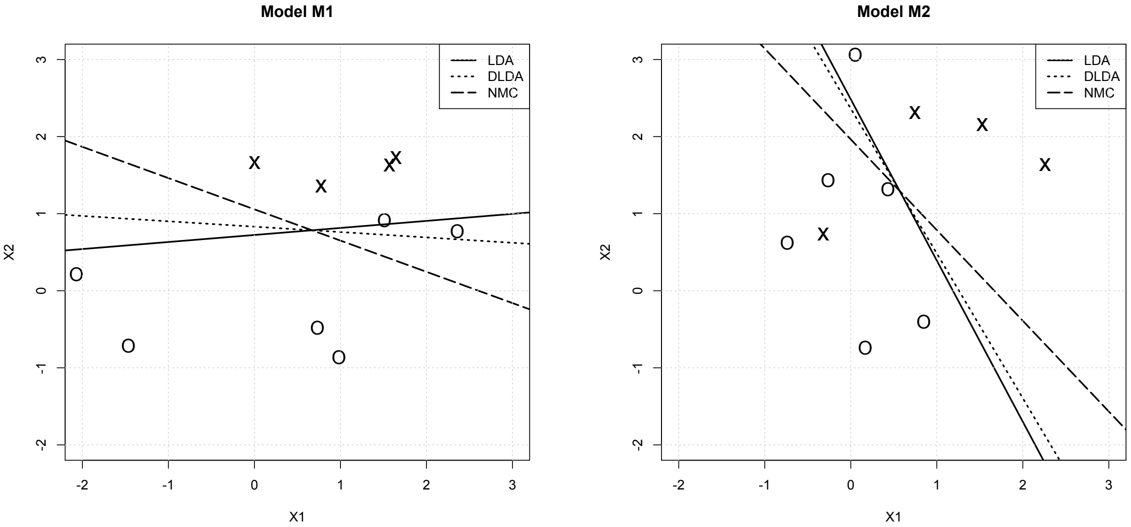
We can see that the Bayes error increases as σ increases, for all models, as expected, since larger variance means more overlapping between the classes. Also, we can see that the models with the smallest Bayes error are the ones with smallest correlations between features.

R code to generate the plots:

```
## Question 1
d <- 6
r <- 0.2
# r <- 0.8
sigma <- seq(0.01,2,0.01)
plot(sigma,pnorm(-sqrt(d)/(2*sigma)),type="l",lwd=2,xlab="sigma",
.... ylab="Bayes Error", ylim=c(0,0.35),main="d = 6, r = 0.2")
lines(sigma,pnorm(-sqrt(d)/(2*sigma*sqrt(1+r))),col="brown",lwd=2)
lines(sigma,pnorm(-sqrt(d)/(2*sigma*sqrt(1+2*r))),col="blue",lwd=2)
legend("bottomright",legend=c("M1","M2","M3"),lwd=2,col=c("black","brown","blue"))
grid()
```

Problem 2: For $d = 2$,$\rho = 0.2$, $\sigma = 1$, draw a sample of size $n = 10$ from models $M1$ and $M2$. Obtain the NMC, LDA, and DLDA decision boundaries corresponding to these data. Plot the data (using O's for class 0 and X's for class 1) in each case, with the superimposed decision boundaries for the NMC, LDA, DLDA classifiers. Compute the error of each classifier using the formula given in class.

Solution: Below are plots of the classifiers for $d = 2$ and $\rho = 0.2$, for each of the models.



The errors for the classifiers on the left are

```
Error LDA =  33.12%
Error DLDA =  30.47%
Error NMC =  26.85%
```

while the errors for those on the right are

```
Error LDA =  26.93%
Error DLDA =  26.89%
Error NMC =  27.35%
```

R code to generate the plots and compute the errors:

```
## Question 2
library(MASS)
mu0 <- c(0,0)
mu1 <- c(1,1)
r <- 0.2
Sig <- matrix(c(1,0,0,1),nrow=2)
Sigr <- matrix(c(1,r,r,1),nrow=2)
n <- 10
n0 <- rbinom(1,n,0.5)
n1 <- n - n0

# choose model here
#S0 <- mvrnorm(n0,mu0,Sig) # Model M1
#S1 <- mvrnorm(n1,mu1,Sig)
S0 <- mvrnorm(n0,mu0,Sigr) # Model M2
S1 <- mvrnorm(n1,mu1,Sigr)
```

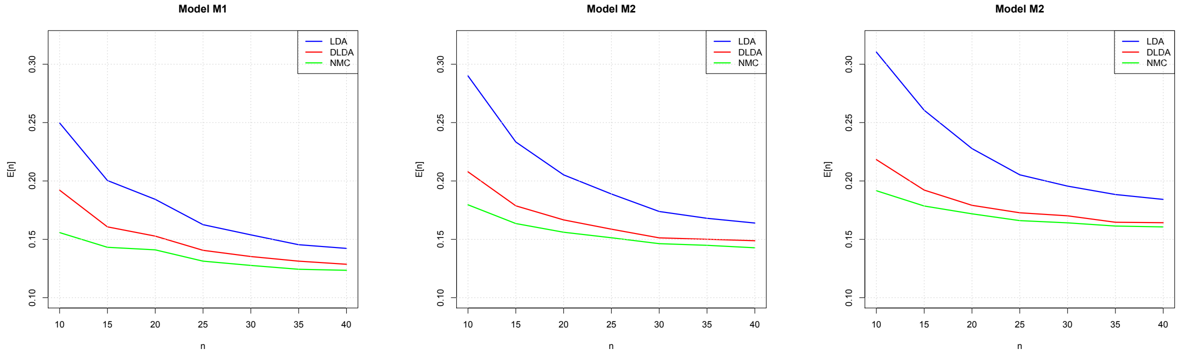
```
mh0 <- colMeans(S0)
mh1 <- colMeans(S1)
Sh0 <- cov(S0)
Sh1 <- cov(S1)
Sh <- 0.5*(Sh0+Sh1)
Si <- solve(Sh)
Sid <- diag(diag(Si))
a.LDA <- Si%*%(mh1-mh0)
b.LDA <- -0.5*t(a.LDA)%*%(mh0+mh1)
a.DLDA <- Sid%*%(mh1-mh0)
b.DLDA <- -0.5*t(a.DLDA)%*%(mh0+mh1)
a.NMC <- (mh1-mh0)
b.NMC <- -0.5*t(a.NMC)%*%(mh0+mh1)

plot(S0[,1],S0[,2],xlim=c(-2,3),ylim=c(-2,3),xlab="X1",ylab="X2",pch="O",
... cex=1.4,main="Model M2")
points(S1[,1],S1[,2],pch="x",cex=1.8)
#points(mh0[1],mh0[2],pch=18,cex=1.8)
#points(mh1[1],mh1[2],pch=22,cex=1.8)
abline(-b.LDA/a.LDA[2],-a.LDA[1]/a.LDA[2],lwd=2)
abline(-b.DLDA/a.DLDA[2],-a.DLDA[1]/a.DLDA[2],lwd=2,lty=3)
abline(-b.NMC/a.NMC[2],-a.NMC[1]/a.NMC[2],lwd=2,lty=5)
legend("topright",legend=c("LDA","DLDA","NMC"),lwd=c(2,2,2),lty=c(1,3,5))
grid()

cat(’Error LDA =
’,0.5*pnorm((t(a.LDA)%*%mu0+b.LDA)/sqrt(t(a.LDA)%*%Sig%*%a.LDA))
.... + 0.5*pnorm(-(t(a.LDA)%*%mu1+b.LDA)/sqrt(t(a.LDA)%*%Sig%*%a.LDA)),’\n’)
cat(’Error DLDA =
’,0.5*pnorm((t(a.DLDA)%*%mu0+b.DLDA)/sqrt(t(a.DLDA)%*%Sig%*%a.DLDA))
... + 0.5*pnorm(-(t(a.DLDA)%*%mu1+b.DLDA)/sqrt(t(a.DLDA)%*%Sig%*%a.DLDA)),’\n’)
cat(’Error NMC =
’,0.5*pnorm((t(a.NMC)%*%mu0+b.NMC)/sqrt(t(a.NMC)%*%Sig%*%a.NMC))
... + 0.5*pnorm(-(t(a.NMC)%*%mu1+b.NMC)/sqrt(t(a.NMC)%*%Sig%*%a.NMC)),’\n’)
```

Problem 3: Extend the previous problem by drawing 100 samples of sizes $n = 10, 15, 20, 25, 30, 35, 40$, $d = 6$, $\rho = 0.2$, $\sigma = 1$ for M1, M2, and M3. There is no need to plot the decision boundaries. Compute the error for the NMC, LDA, and DLDA classifiers in each case using the analytical formula. Average over the 100 samples to obtain an aproximation of $E[\varepsilon_n]$ and plot this as a function of n in each case. Interpret what you see.

Solution: Below are plots for each of the modelssifiers for $d = 6$, $\rho = 0.2$, $\sigma = 1$, for each of the models.



We can see that the expected classification error decreases as the sample size increase. We also notice that NMC gives the best error rates in the high-dimensional, small sample size setting, followed closely by DLDA, and then LDA. Once again, the error increase from Model M1 to M2 to M3, due to increasing correlation between the features.

R code to generate the plots:

```
## Question 3
library(limma)

mu0 <- c(0,0,0,0,0,0)
mu1 <- c(1,1,1,1,1,1)
r <- 0.2

# generate cov matrices
Sig1 <- diag(mu1)
Sigt <- matrix(c(1,r,r,1),nrow=2); Sig2 <- blockDiag(Sigt,Sigt,Sigt)
Sigt <- matrix(rep(r,9),nrow=3); diag(Sigt) <- 1; Sig3 <- blockDiag(Sigt,Sigt)

ee.LDA <- vector("numeric",length(n))
ee.DLDA <- vector("numeric",length(n))
ee.NMC <- vector("numeric",length(n))

n <- seq(10,40,5)

Sig <- Sig3 # choose model here

for (i in 1:length(n)) {
    for (j in 1:100) {

    n0 <- rbinom(1,n[i],0.5)
    n1 <- n[i] - n0
    if ((n0<=1)||(n0>=n[i]-1)) next

    # generate data
    S0 <- mvrnorm(n0,mu0,Sig)
    S1 <- mvrnorm(n1,mu1,Sig)

    mh0 <- colMeans(S0)
    mh1 <- colMeans(S1)
    Sh0 <- cov(S0)
    Sh1 <- cov(S1)
    Sh <- 0.5*(Sh0+Sh1)
    Si <- solve(Sh)
    Sid <- diag(diag(Si))

    a.LDA <- Si%*%(mh1-mh0)
    b.LDA <- -0.5*t(a.LDA)%*%(mh0+mh1)
    ee.LDA[i] <- ee.LDA[i]+0.5*pnorm((t(a.LDA)%*%mu0+b.LDA)/sqrt(t(a.LDA)%*%Sig%*%a.LDA))
    ... + 0.5*pnorm(-(t(a.LDA)%*%mu1+b.LDA)/sqrt(t(a.LDA)%*%Sig%*%a.LDA))

    a.DLDA <- Sid%*%(mh1-mh0)
    b.DLDA <- -0.5*t(a.DLDA)%*%(mh0+mh1)
    ee.DLDA[i] <- ee.DLDA[i]+0.5*pnorm((t(a.DLDA)%*%mu0+b.DLDA)/sqrt(t(a.DLDA)%*%Sig%*%a.DLDA))
    ... + 0.5*pnorm(-(t(a.DLDA)%*%mu1+b.DLDA)/sqrt(t(a.DLDA)%*%Sig%*%a.DLDA))

    a.NMC <- (mh1-mh0)
    b.NMC <- -0.5*t(a.NMC)%*%(mh0+mh1)
    ee.NMC[i] <- ee.NMC[i]+0.5*pnorm((t(a.NMC)%*%mu0+b.NMC)/sqrt(t(a.NMC)%*%Sig%*%a.NMC))
    ... + 0.5*pnorm(-(t(a.NMC)%*%mu1+b.NMC)/sqrt(t(a.NMC)%*%Sig%*%a.NMC))
}
ee.LDA[i] <- ee.LDA[i]/100
ee.DLDA[i] <- ee.DLDA[i]/100
ee.NMC[i] <- ee.NMC[i]/100
}

plot(n,ee.LDA,type="l",lwd=2,col="blue",xlab="n",ylab="E[n]",ylim=c(0.10,0.32),
... main="Model M1")
lines(n,ee.DLDA,lwd=2,col="red")
lines(n,ee.NMC,lwd=2,col="green")
legend("topright",legend=c("LDA","DLDA","NMC"),lwd=c(2,2,2),col=c("blue","red","green"))
grid()
```