INFOF422: bonus 6

Classification and ROC CURVE

G. Bontempi

Question

Consider a binary classification task where the input $\mathbf{x} \in \mathbb{R}^2$ is bivariate and the categorical output variable $\mathbf{y}$ may take two values: 0 (associated to red) and 1 (associated to green).

Suppose that the a-priori probability is $p(\mathbf{y} = 1) = 0.2$ and that the inverse (or class-conditional) distributions are

- green/cross class : mixture of three Gaussians

$$p(x|\mathbf{y} = 1) = \sum_{i=1}^{3} w_i \mathcal{N}(\mu_{1i}, \Sigma)$$

  where in $\mu_{11} = [1, 1]^T, \mu_{12} = [-1, -1], \mu_{13} = [3, -3]^T$, and $w_1 = 0.2$, $w_2 = 0.3$.
- red/circel class: bivariate Gaussian $p(x|\mathbf{y} = 0) = \mathcal{N}(\mu_0, \Sigma)$ where $\mu_0 = [0, 0]^T$

The matrix Σ is a diagonal identity matrix.

The student should

- by using the R function `rmvnorm`, sample a dataset of $N = 1000$ input/output observations according to the conditional distribution described above,
- visualise in a 2D graph the dataset by using the appropriate colors,
- plot the ROC curves of the following classifiers

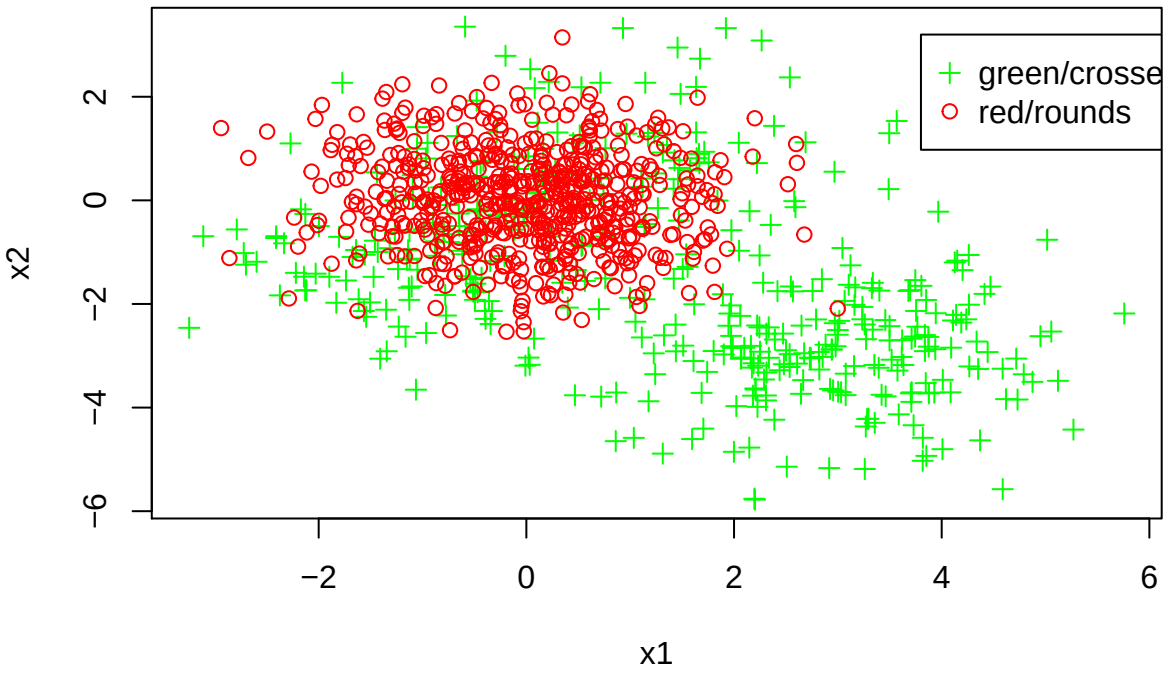
  1. linear regression coding the two classes by 0 and 1,
  2. Linear Discriminant Analysis where $\sigma^2 = 1$,
  3. Naive Bayes where the univariate conditional distributions are Gaussian,
  4. k Nearest Neighbour with $k = 3, 5, 10$.

  The classifiers should be trained and tested on the same training set.
- Choose the best classifier on the basis of the ROC curves above.

Data generation

The sampling of data from a mixture of gaussians requires two steps: first the sampling of the component (using the distribution characterized by the weights) and then the sampling of the associated gaussian distribution. Equivalently you may for each ith component draw a number of samples equal to $w_i N$.

```
library(mvtnorm)
N=1000

p1=0.4

w1=0.2
w2=0.3

mu11=c(1,1)
mu12=c(-1,-1)
mu13=c(3,-3)
Sigma1=diag(2)
N1=N*p1
X=rbind(rmvnorm(round(w1*N1),mu11,Sigma1),
        rmvnorm(round(w2*N1),mu12,Sigma1),
        rmvnorm(N1-round(w1*N1)-round(w2*N1),mu13,Sigma1))

N0=N-N1
mu0=c(0,0)
Sigma0=diag(2)
X=rbind(X,rmvnorm(N0,mu0,Sigma0))
Y=numeric(N)
Y[1:N1]=1


plot(X[1:N1,1],X[1:N1,2],col="green",pch=3,xlab="x1",ylab="x2")
points(X[(N1+1):N,1],X[(N1+1):N,2],col="red",pch=1)

legend(3.8,3.2,c("green/crosses","red/rounds"),
       col=c("green","red"),pch=c(3,1))
```



Implementation classifiers

```
Lin<-function(X,Y){
  N=length(Y)
  X=cbind(numeric(N)+1,X)
  Yhat<-X%*%solve(t(X)%*%X)%*%t(X)%*%Y
  return(Yhat)

}

LDA<-function(X,Y){
  sigma=1 ## value fixed in the problem assignement
  I1=which(Y==1)
  P1=length(I1)/length(Y)
  muhat1=apply(X[I1,],2,mean)
  w1=array(muhat1/sigma^2,c(NCOL(X),1))
  w10=-1/(2*sigma^2)*t(muhat1)%*%muhat1+log(P1)
  return(X%*%w1+c(w10))
}

KNN<-function(X,Y,K=3){
  N=length(Y)
  Yhat=numeric(N)
  for (i in 1:N){
    d=apply((sweep(X,2,X[i,]))^2,1,sum)
    nn=sort(d,decr=FALSE,index=TRUE)$ix[1:K]
    Yhat[i]=mean(Y[nn])
  }

  return(Yhat)
```

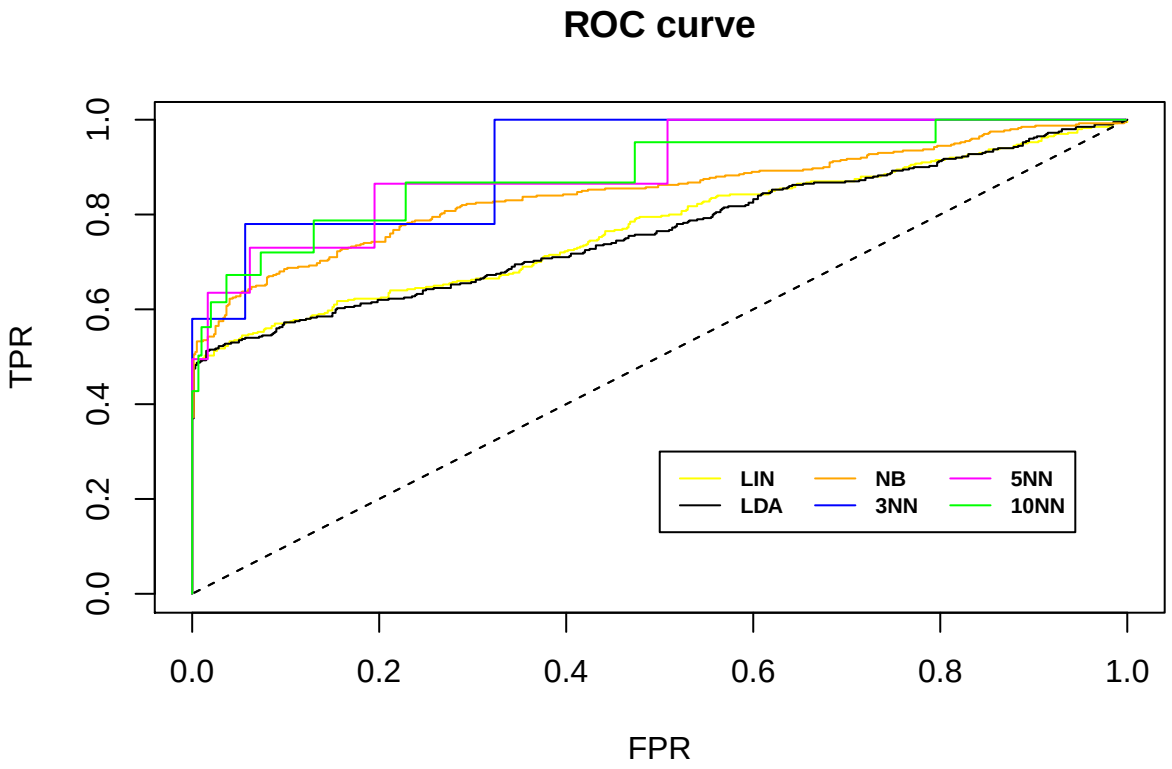
```
}

NB<-function(X,Y,K=3){
  N=length(Y)
  Yhat=numeric(N)
  I1=which(Y==1)
  I0=which(Y==0)
  p1=length(I1)/N
  p0=1-p1
  for (i in 1:N){
    p1x1=dnorm(X[i,1],mean(X[I1,1]),sd(X[I1,1]))
    p1x2=dnorm(X[i,2],mean(X[I1,2]),sd(X[I1,2]))

    p0x1=dnorm(X[i,1],mean(X[I0,1]),sd(X[I0,1]))
    p0x2=dnorm(X[i,2],mean(X[I0,2]),sd(X[I0,2]))

    Yhat[i]=p1x1*p1x2*p1/(p1x1*p1x2*p1+p0x1*p0x2*p0)
  }

  return(Yhat)

}
```

Computation and plot of ROC curves

```
Yhat1=Lin(X,Y)
Yhat2=LDA(X,Y)
Yhat3=NB(X,Y)
Yhat4=KNN(X,Y,3)
Yhat5=KNN(X,Y,5)
Yhat6=KNN(X,Y,10)
s1<-sort(Yhat1,decreasing=FALSE,index=TRUE)
s2<-sort(Yhat2,decreasing=FALSE,index=TRUE)
s3<-sort(Yhat3,decreasing=FALSE,index=TRUE)
s4<-sort(Yhat4,decreasing=FALSE,index=TRUE)
s5<-sort(Yhat5,decreasing=FALSE,index=TRUE)
s6<-sort(Yhat6,decreasing=FALSE,index=TRUE)
TPR1=NULL
FPR1=NULL
TPR2=NULL
FPR2=NULL
TPR3=NULL
FPR3=NULL
TPR4=NULL
FPR4=NULL
TPR5=NULL
FPR5=NULL
TPR6=NULL
FPR6=NULL

FP1=NULL
FN1=NULL
```

```
FP2=NULL
FN2=NULL
FP3=NULL
FN3=NULL
FP4=NULL
FN4=NULL
FP5=NULL
FN5=NULL
FP6=NULL
FN6=NULL

for (i in 1:N){
  I1=s1$ix[1:i]
  TPR1=c(TPR1,length(which(Y[setdiff(1:N,I1)]==1))/N1)
  FPR1=c(FPR1,length(which(Y[setdiff(1:N,I1)]==0))/N0)
  FP1=c(FP1,length(which(Y[setdiff(1:N,I1)]==0)))
  FN1=c(FN1,length(which(Y[I1]==1)))


  I2=s2$ix[1:i]
  TPR2=c(TPR2,length(which(Y[setdiff(1:N,I2)]==1))/N1)
  FPR2=c(FPR2,length(which(Y[setdiff(1:N,I2)]==0))/N0)
  FP2=c(FP2,length(which(Y[setdiff(1:N,I2)]==0)))
  FN2=c(FN2,length(which(Y[I2]==1)))

  I3=s3$ix[1:i]
  TPR3=c(TPR3,length(which(Y[setdiff(1:N,I3)]==1))/N1)
  FPR3=c(FPR3,length(which(Y[setdiff(1:N,I3)]==0))/N0)
  FP3=c(FP3,length(which(Y[setdiff(1:N,I3)]==0)))
  FN3=c(FN3,length(which(Y[I3]==1)))

  I4=s4$ix[1:i]
  TPR4=c(TPR4,length(which(Y[setdiff(1:N,I4)]==1))/N1)
  FPR4=c(FPR4,length(which(Y[setdiff(1:N,I4)]==0))/N0)
  FP4=c(FP4,length(which(Y[setdiff(1:N,I4)]==0)))
  FN4=c(FN4,length(which(Y[I4]==1)))

  I5=s5$ix[1:i]
  TPR5=c(TPR5,length(which(Y[setdiff(1:N,I5)]==1))/N1)
  FPR5=c(FPR5,length(which(Y[setdiff(1:N,I5)]==0))/N0)
  FP5=c(FP5,length(which(Y[setdiff(1:N,I5)]==0)))
  FN5=c(FN5,length(which(Y[I5]==1)))

  I6=s6$ix[1:i]
  TPR6=c(TPR6,length(which(Y[setdiff(1:N,I6)]==1))/N1)
  FPR6=c(FPR6,length(which(Y[setdiff(1:N,I6)]==0))/N0)
  FP6=c(FP6,length(which(Y[setdiff(1:N,I6)]==0)))
  FN6=c(FN6,length(which(Y[I6]==1)))

}


plot(FPR1,TPR1,type="l",col="yellow",
```

```
     main="ROC curve",xlab="FPR",
     ylab="TPR")
lines(FPR1,FPR1,lty=2)
lines(FPR2,TPR2,col="black")
lines(FPR3,TPR3,col="orange")

lines(FPR4,TPR4,col="blue",type="l")
lines(FPR1,FPR1,lty=2)
lines(FPR5,TPR5,col="magenta")
lines(FPR6,TPR6,col="green")
legend(0.5,0.3,c("LIN","LDA","NB","3NN","5NN","10NN"),
       col=c("yellow","black","orange","blue","magenta","green"),lty=1,
       text.font = 2,pt.cex = 1, cex = 0.7,ncol=3)
```



Looking at the ROC curves the best two classifiers are the 3NN and 5NN. For a better assessment of their accuracy the area under the ROC curve should be computed.