Machine learning

Classification models

September 2019

Aim of the practical session

- being able to use the R langage to handle proof of concepts with classication methods.

Remarks

- The practical work must be carried on with a group of two students. 'R studio' is used to program with the R langage.
- You should provide a small report using the Markdown format on exercice D (`R markdown` file). The work is due before the next session. Your names have to be written in the first lines of program file with comments.

A. Data set and Simulated data

- Use the library `mvtnorm` to simulate a two dimensional sample composed of a mixture of two gaussians as illustrated in the following figure.

The first group (blue points) contains $n = 100$ observations distributed as $\mathcal{N}(\begin{bmatrix} 2 & 2 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix})$.

The second group (red points) contains $n = 100$ observations distributed as $\mathcal{N}(\begin{bmatrix} 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0.75 \\ 0.75 & 1 \end{bmatrix})$.

```
## Warning: package 'mvtnorm' was built under R version 3.4.3
```

- Helping yourself with following instructions, generate a grid of points regularly spaced.

```
#Points of the Grid generation.
K=40;
seqx1=seq(min(X[,1]),max(X[,1]), length=K);
seqx2=seq(min(X[,2]),max(X[,2]), length=K);
mygrid=expand.grid(z1=seqx1,z2=seqx2);
names(mygrid)=names(Z)[c(1,2)];
```

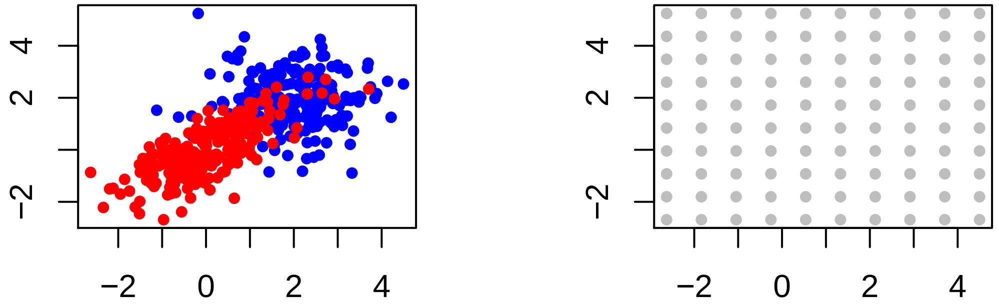

Figure 1: Gaussian Mixture and Grid of points for prediction (here K=10)

B. Machine learning classifiers

This section details the instructions in order to implement different supervized classifiers in R.

- The Bayes Classfiers
- The Linear Discriminant Analysis (LDA)
- The Quadratic Discriminant Analysis (QDA)
- The Logistic Regression classifier (LogReg)
- The K Nearest Neighbours classifieurs (KNN)

1. Bayes classifier

Using the previous simulated data, program the Bayes classifieur in order to construct a predictive model able to automatically separate the 2 dimentionnal space in 2 groups.

a) First estimate the average and the standard deviation for each group, and for each of the two dimensions.

b) Use the `dmvnorm()`function to compute the likelihood of a new point to belong to both distributions

c) Generate $K * K$ points regularly spaced ($K = 40$) and compute for each point the class provided by the bayes classifier with the help of the following instructions where `Ypredgrid` contains the predictive values $1, 2$ for the points af the generated grid.

d) With the help of the following instructions, visualize the 2 domains automatically classified.

```
red2=rgb(red=254/255,green=231/255,blue=240/255,alpha=0.2); # red color with transparency
blue2=rgb(red=51/255,green=161/255,blue=201/255,alpha=0.2); # blue color with transparency
plot.new();
image(seqx1,seqx2,matrix(Ypredgrid,K),col=c(blue2, red2),xlab="",ylab="",xaxt="n",yaxt="n");
contour(seqx1,seqx2,matrix(Ypredgrid,K),col="black",lty=1, add=TRUE,drawlabels = FALSE,nlevels=1);

points(X[,1],X[,2], col=c("blue","red")[Y],pch=16,lwd=2,cex=0.8);
contour(seqx1,seqx2,matrix(Ypredgrid,K),col="black",lty=1,add=TRUE, drawlabels = FALSE);
```

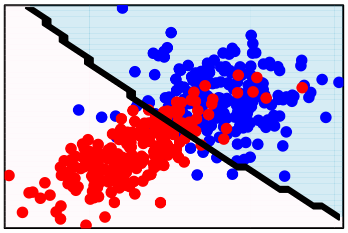

Figure 2: Bayes classifier

2. Linear Discriminant Analysis

The `lda()` function of the `MASS` library is the function used to perform a Linear Discriminant Analysis in R. With the help of the following instructions and the previous work for the Bayes classifieur, use the simulated data set to construct a classifier using the LDA method.

```
library(MASS);
model.lda=lda(Y~.,data=Z);
Ypred=predict(model.lda,newdata=mygrid); Ypredgrid=Ypred$class;
```

Here, Z denotes the data frame containing the labelled simulated data.

```
##         X1        X2 Y
## 1 1.147948 2.3544203 1
## 2 2.858082 2.4905117 1
## 3 2.427799 0.8673624 1
```

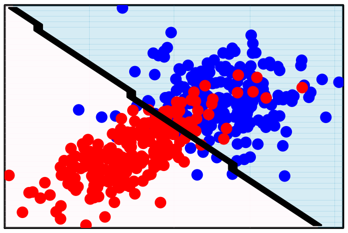

Figure 3: LDA classifier

3. Quadratic Discriminant Analysis

The `qda()` function of the `MASS` library is the function used to perform a Linear Discriminant Analysis in R. With the help of the following instructions and the previous work for the LDA classifier, use the simulated data set to construct a classifier using the QDA method.

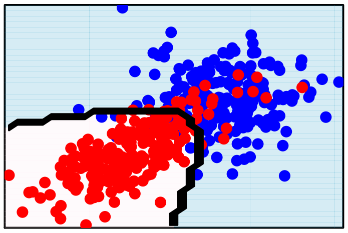

Figure 4: QDA classifier

4. Logistic regression

The `glm()` function of the `MASS` library is the function used to perform a logistic regression in R. With the help of the following instructions and the previous work for the classifiers, use the simulated data set to construct a classifier using the logistic regression.

```
library(MASS);
Z0=Z; Z0$Y=as.numeric(Z0$Y)-1;
modelglm=glm(Y~.,data=Z0,family="binomial");
Ypred=predict(modelglm,newdata=mygrid); Ypredgrid=1*(Ypred>0)+1;
```

5. K Nearest Neighbours (KNN)

The function `knn()` of the library class is the function used to perform the K nearest Neigbours methods on data. With the help of the following instructions and the previous work for the classifiers, use the simulated data set to construct two classifiers using the KNN method with $k = 3$ and $k = 41$.

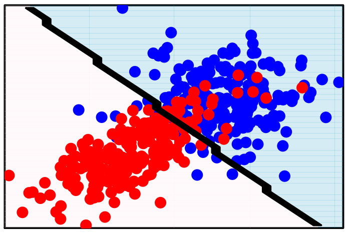

Figure 5: Logistic regression

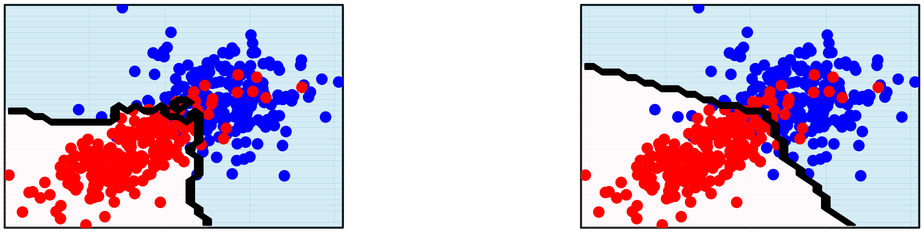

Figure 6: KNN classifier

C. Evaluation of the Predictive Power of the classifiers. Generalization.

Our goal is now to compare the different classifiers, regarding their capacity of generalization, i.e. the gobal performance (or the error), the False Positive (FP) and the True Negative proportions (TN).

- Using the K fold methodology, compute for each of the classifiers the different criteria of performance on the train set and for the test sets, for K repetitions.
- Visualize the performances using the `boxplot` R function as presented on the following figure, for the global performance.

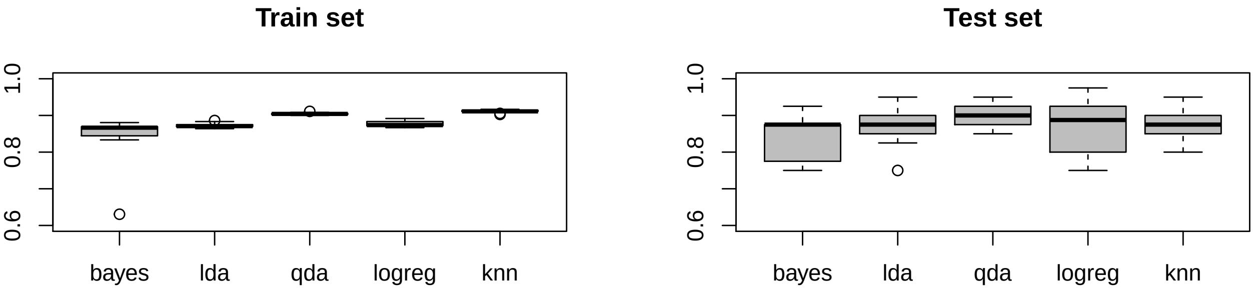

Figure 7: Classifier Comparison

- Conclusions.

D. On your own...

- Simulate 2 groups of 2 dimensional points with your favorite distributions and visualize it!
- Find the best classifier ! by implementing and evaluating the performances on train and test sets of the studied classifiers.

Here, some ideas of distributions.

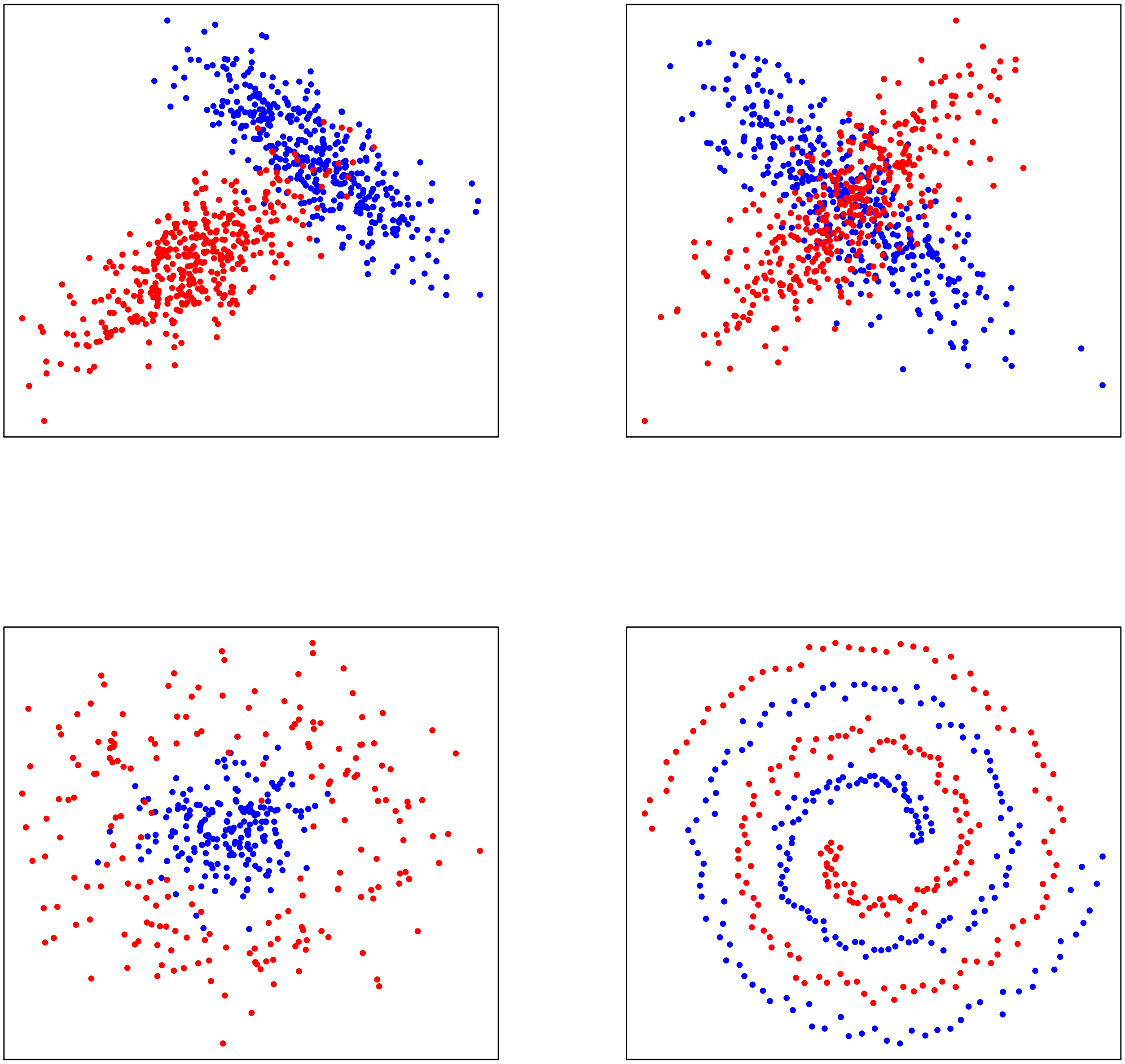

Figure 8: 2D distributions