Machine Learning

Classification. Decision Trees

November 2017

Aim of the Practical session

- Classification and Regression Trees. Bagging. Random Forest

Remarks

- The practical work must be carried on with a group of two students. 'R studio' is used to program the work with the help of R langage.
- You should provide a small report using the Markdown format on exercice C (`R markdown` file). The work is due for 'monday 17^{th} december2017 and will be uploaded on the web exercice site (MALTP1). The report should not exceed 10 pages.

A. Simulated data

Simulate $n = 400$ observations randomly distributed as in the following figure.

```
## Warning: package 'mvtnorm' was built under R version 3.3.2
```

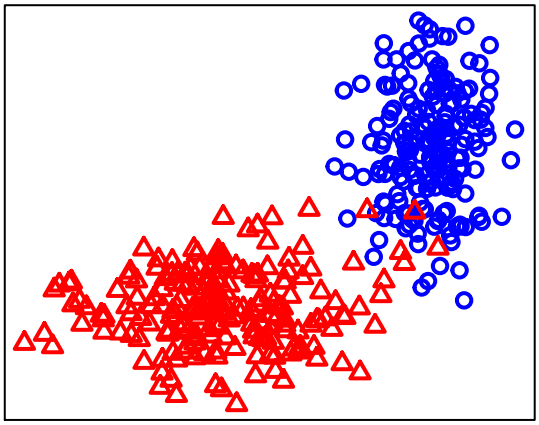

Figure 1: Data Set

We suppose that the simulated data are stored in a data frame called Z with 3 columns corresponding to X_1 (horizontal axis), X_2 (vertical axis), and Y coding the number of the class.

Y is a factor ($Z\$Y = as.factor(Z\$Y)$). We consider that the data frame Z contains the training data.

```
##         X1       X2 Y
## 1 3.051654 4.055483 1
## 2 3.069501 2.818031 1
## 3 2.840009 2.474248 1
```

a) The Classification And Regression Tree (CART) algorithm can be appied on data using the following instructions. Note the name of the library (`tree`) and the function to compute the tree.

```
library(tree);
Z$Y=as.factor(Z$Y);
modtree=tree(Y~.,data=Z);
plot(modtree); text(modtree,cex=0.8);
```

Note that for classification, the target variable Y has to be defined as a 'factor' otherwise the `tree()` function performs regression tree.

b) We consider now a grid of points regularly spaced on the domain as already programmed in the previous practical session. With the help of the grid and the predict.tree() function, compute and visualize the decision function of the classifier on all the domain.

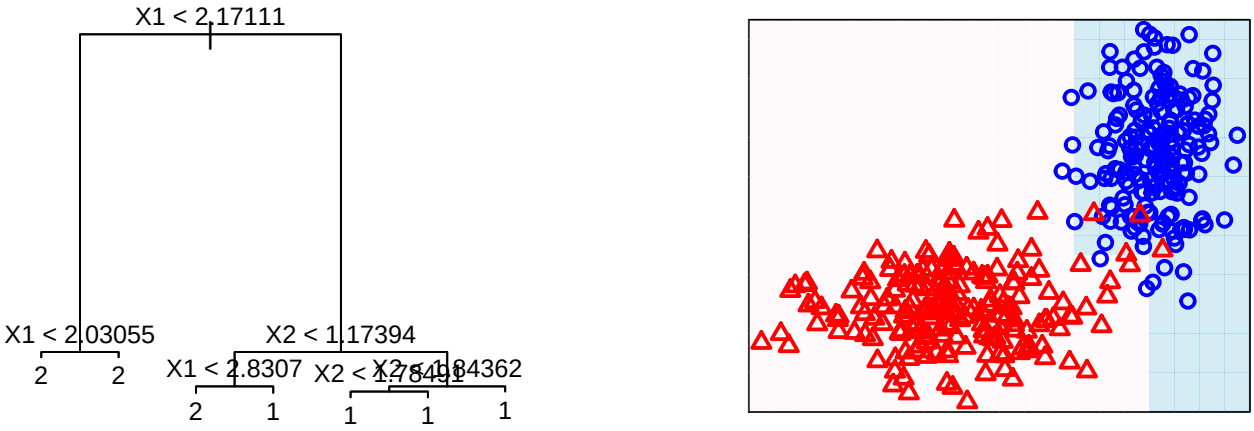

Figure 2: Classification Tree. Representation of decision domains

- Conclusion.

Bagging with trees

- The 'ipred' library contains a `bagging()` function to implement a CART classifier with bagging
- With the help of the following instructions, implement a bagging classifier on the simulated data.

```
library(ipred)
modbag = bagging(Y~.,data = Z,coob = TRUE);
```

- Excecute the following instruction and comment the results.

```
summary(modbag)
```

- How many bootstrap samples are by default generated with the bagging function?
- What means OOB ? What does the field *modbag$err* contain ?
- Conclusion
- With the help of the grid and the predict.tree() function, computed and visualize the decision function of the classifier.

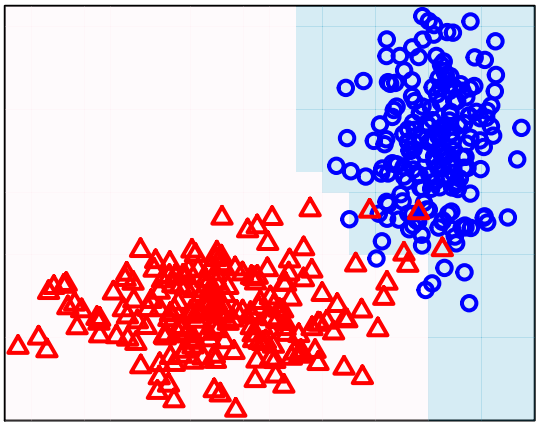

Figure 3: Classification bagging Tree: decision domains

Random Forest

- Use the library and the function 'randomForest' to implement a random forest classifier on the simulated data.
- How many trees are by default generated ?
- Visualize the decision domains

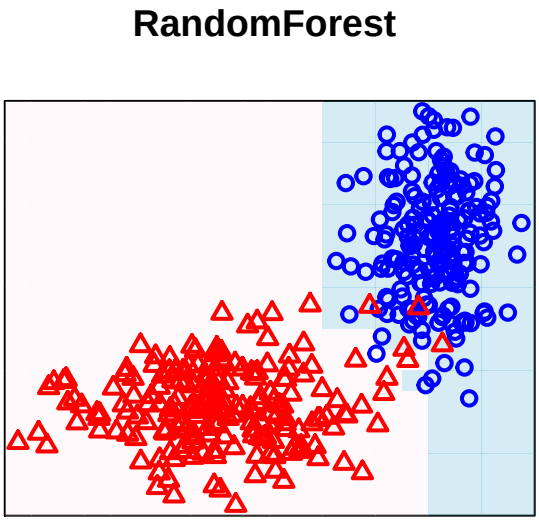

Figure 4: Classification Random Forest Tree

B. Application on health. Heart Attack data.

Random Forest

- Use the library and the function 'randomForest' to implemente a random forest classifier.
- How many trees are by default generated with the randomForest function?
- Compute and visualize the decision domains using the grid of points.

C. Spam detection engines.

The aim of this section is to study different learning machines to detect automatically a regular email from a spam. The package `ElemStatLearn` contains the dataset named `spam`. Use the instruction `help(spam)` to get information on this data set. To load the package:

```
library(ElemStatLearn)
```

- Compare the performances of the different machines already studied (Bayes, ADL, QDA, Logistic Regression, CART, Bagging and Random Forest).
- What machine would you advised ? justify your choice.