TIS02 Automne 2015
TP 3
Bagging, Random Forests, Boosting

1 Bagging, Random Forests

We want to deploy bagging and random forests on several multiclass datasets, in order to compare their performances. In a first step, these techniques will be compared to a single decision tree. You can use the library tree in order to train, prune, and use a decision tree. Bagging and random forests can be applied via the functions in the library randomForest. Remember that you can load a library (say, e.g., libname) into memory using the following command :

```
library(libname)
```

The code for repeating the procedure of splitting a dataset and training/testing so as to estimate the misclassification rate, given during the first practical session, should be recycled. First deploy the various algorithms on synthetic data generated according to a mixture of two bivariate Gaussian distributions. Then, use the datasets iris, crabs (library MASS), Pima, bcw, spambase, and pendigits. The last four datasets should be loaded by hand (see the command read.csv in the code in TP1). The crabs dataset should be processed as follows :

```
Donn <- crabs[,4:7]/matrix(rep(crabs[,8],4),ncol=4,byrow=FALSE)
Donn <- cbind(Donn,c(rep('BM',50),rep('BF',50),rep('OM',50),rep('OF',50)))
```

For all datasets, the class information is contained in the last column. This should be formalized using the following command :

```
Donn[,dim(Donn)[2]] <- as.factor(Donn[,dim(Donn)[2]])
names(Donn)[dim(Donn)[2]] <- "class"
```

– For each of the datasets, train a single decision tree to predict the class according to the other data (without repeating the splitting/training/testing procedure). Compare the performances of the unpruned and pruned decision trees. You can use the command table to construct the confusion matrix of the classes. What can you observe ?
– Use bagging and a random forest so as to predict the class according to the other variables. Process the results with the same analyses as above. Compare the results obtained.

2 Boosting and overall comparison

Now, we want to use boosting to predict the class information. You can download the gbm library. As before, for each dataset, first test the procedure on a single dataset, in order to get a grasp on the data. Perform the same analyses as for bagging and random forests. (Don't forget to make a distinction between two-class and multiclass datasets.)
Then, repeat the splitting/training/testing procedure in order to compute average error rates. Compare the single decision tree (unpruned and pruned), bagging, random forests, boosting, linear discriminant analysis, quadratic discriminant analysis, and naive Bayes. (For the discriminant analysis models, don't forget to scale the data first.)