Tutorial: Bagging, Boosting and Random Forests

We outline several of the bagging and random forest methods outlined.

Bagging and Random Forests We'll consider the `CarSeats` data set. This is a simulated data set containing sales of child car seats at 400 different stores. Sales can be predicted by 10 other variables.

Follow the script to load the required packages and locate the data set, examine the descriptives suggested in the script. The outcome of interest will be a binary version of the `Sales` variable, which gives the number of sales (in thousands) at each location.

Plot the data (according to the script), split it into training and test sets, of equal sizes.

Single Classification Tree We have already discussed classification trees. We'll start by running a single classification tree and see how it performs. We'll then use the bagging and random forest techniques and see if they give improvement.

Follow the script, find the classification tree, check its performance for prediction (follow the script).

One important measure is the *Receiver Operating Characteristic (ROC)* curve. This plots the *true positve rate* against the *false positive rate* at various threshold setings. The area under the receiver operating characteristic curve (AUROC) is a good measure of the performance (larger the better).

Bagging of Classification Trees

We now try *bagging*, which means that we take bootstrap samples and construct the tree for each bootstrap sample, as described in the lectures. The training and test sets are defined quite naturally; the 'out of bag' observations are used for testing.

Follow the script. Two R routines for bagging are suggested, the `bag(.)`function and the `treebag` option in `train`. Try both of these. Do they give the same result (as claimed)? Follow the script. Compute the confusion matrices for the test data based on the classifier computed using the training data. Does bagging represent an improvement over a single classification tree?

Random Forest for Classification Trees

We now see how the random forest approach works out in practise and whether or not it gives better classification than bagging or taking a single tree.

Under `train` a random forest is computed using `method = ''rf''`; in all other respects it is similar to `treebag`. You need the package **randomForest**.

Classification Forest for Conditional Inference Tree

Now try using conditional inference trees (found in the **party** package) for learning (follow the script). Does this represent an improvement over the methods considered so far?

Random Forest with Boosting

We considered `adaboost` (available in the **ada** package) in the lecture. There are other techniques such as *gradient boosting* (available in **gbm**). This simply minimises the loss using gradient descent. Test this and see if it offers any improvement.

Model Comparison

Finally, plot all the ROC curves obtained from the various methods on the same plot. Which is the most powerful classifier?