bootfs - Bootstrapped feature selection

Christian Bender *

November 25, 2012

This document describes the package 'bootfs' for robust feature selection in classification problems from high-throughput data, such as genomic or proteomic screening data. Several methods for classification are combined, in order to derive a robust estimation of the importance of each feature used for classification.

1 Using bootfs

This section contains the basic steps for analysis of high-throughput data, consisting of a number of samples to classify (such as patients) and a number of features, such as genes or proteins. Samples originate in two sample groups, for instance healthy versus sick patients. The aim is to find the most important features discriminating the two classes of patients.

The usage of the package is illustrated for three classification algorithms: PAMR (Prediction analysis for Microarrays, [3]), RF-Boruta (Random forests with the Boruta algorithm for feature selection, [2]) and SCAD-SVM (Support Vector Machines with Smoothly Clipped Absolute Deviation feature selection, [4], implementation from [1]). First of all load the package:

For illustration purpose, some data can be simulated:

1.1 Simulating data

In this section we show how to generate artificial networks and data. A reference signalling network is simulated and used to sample measurements that incorporate the network structure.

First, simulate a network with 6 nodes and 2 distinct input stimuli.

*Translational Oncology (TRON) Mainz, Germany. eMail: christian.bender@tron-mainz.de

```
> set.seed(1234)
> data <- simDataSet(nsam=30, ngen=100, sigma=1.5, plot=TRUE)
> logX <- data$logX
> groupings <- data$groupings
```

1.2 Assessing performance of the classifiers

The first step is to verify, if the feature selection algorithms perform sufficienctly on the given data set. For this, a crossvalidation of the different classification algorithms is run:

```
> ## run the crossvalidation
> ## note the number of repeats should be set to 10 or so,
> ## it is set to 2 here to have a low running time of this illustration
> retCV <- doCV(logX, groupings, fs.methods = c("pamr", "scad", "rf_boruta"),
+         DIR = "cv", seed = 123, ncv = 5, repeats = 2,
+         jitter=FALSE, maxiter = 100, maxevals = 50,
+         max_allowed_feat = 50, n.threshold = 50, maxRuns = 30)
```

The above command uses the classification methods PAMR, SCAD-SVM and RF-Boruta and performs a 5-fold (ncv=5) crossvalidation on the training data, repeating the crossvalidation 2 times with different training/test set assignments.

The results are summarised as ROC curves, shown in figures 1, 2 and 3.

1.3 Do the feature selection and importance ranking

If the performance is of sufficient quality, as seen in the ROC curves generated during the cross-validation, the bootstrapping approach for deriving the most important features for this classification task can be done. Again, we select the three algorithms from above and perform bootstrapping on the input data. For each bootstrapping data set the feature selections are done using each algorithm.

```
> ## run the bootstrapping
> retBS <- doBS(logX, groupings,
+         fs.methods=c("pamr","scad","rf_boruta"),
+         DIR="bs",
+         seed=123, bstr=15, saveres=FALSE, jitter=FALSE,
+         maxiter=100, maxevals=50, bounds=NULL,
+         max_allowed_feat=NULL, n.threshold=50,
+         maxRuns=30)
```

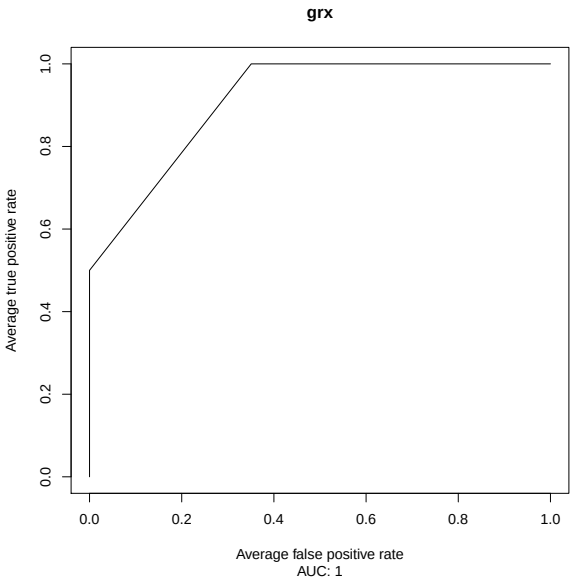

Figure 1: Receiver Operator Characteristic (ROC) curve for the PAM algorithm cross-validation.

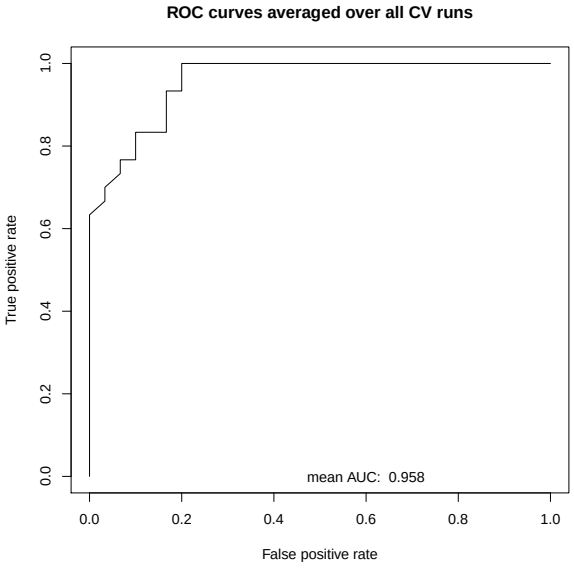

Figure 2: Receiver Operator Characteristic (ROC) curve for the random forest RF-Boruta algorithm cross-validation.

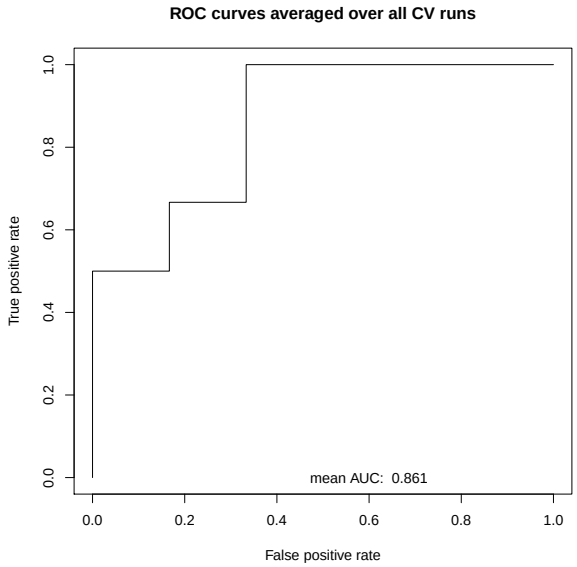

Figure 3: Receiver Operator Characteristic (ROC) curve for the SCAD-SVM algorithm cross-validation.

Here, 15 bootstrap sample sets are drawn (bstr=15) and feature Selection is done for each sample set. The group proportions are kept constant during the sample selection. Now, for each method, a separate importance graph can be generated:

```
> ## show an importance ranking for a single
> ## classification method
> bsres <- makeIG(retBS[[1]], SUBDIR=NULL, prob=.9)
```

This might be useful to inspect the drawn features for each method separately. The parameter *prob=.9* can be used to set the cutoff, how often a feature must cooccur with another feature, such that an edge is drawn between them. However, the general ranking of the importance of the features is done by generating the combined importance graph:

```
> ## create the combined importance graph for all methods
> ## and export the adjacency matrix containing the
> ## numbers of occuerrences of the features, as well
> ## as the top hits.
> res <- resultBS(retBS, DIR="bs", vlabel.cex = 3, filter = 8, saveres = FALSE)
```

There are several arguments which customise the look of the importance graph. In this call, the *vlabel.cex* argument defines the magnification factor

of the node labels (each node corresponds to one feature). The argument *filter* can be used to specify, how often a feature must cooccur with another, such that an edge is drawn between the two features. Figure 4 shows the result of the above call.

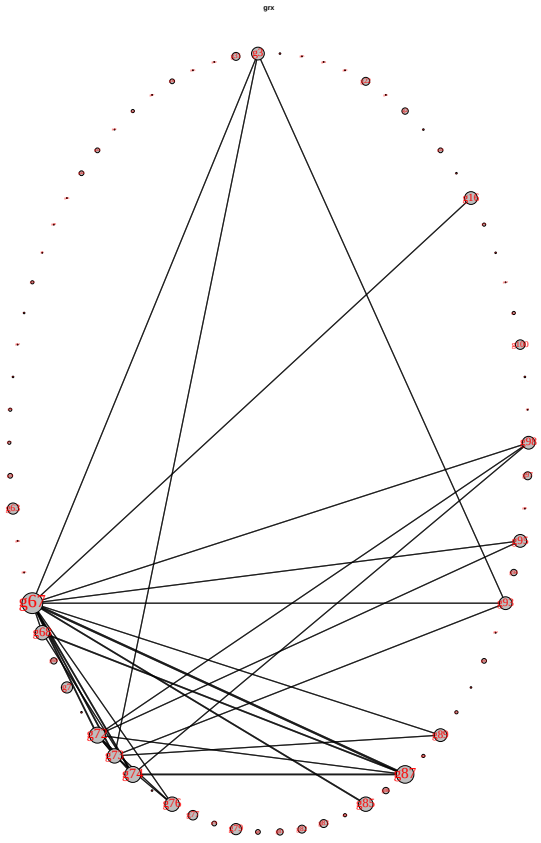

Figure 4: Importance graph generated by *resultBS*.

The graph can be customised more flexibly using the *importance_igraph* function directly:

```
> ## plot the importance graph directly. Gives more
> ## flexibility to adjust the graph
>
> ig <- importance_igraph(res$adj, main = "my test",
+         highlight = NULL,        layout="layout.ellipsis",
+                 pdf=NULL, pointsize=12, tk=FALSE,
+                 node.color="grey", node.filter=NULL,
+                 vlabel.cex=2, vlabel.cex.min=0.5, vlabel.cex.max=4,
```

```
+                 max_node_cex=8,
+          edge.width=2, filter=8, max_edge_cex=4, ewprop=8 )
```

Figure 5 shows the importance graph generated with the above call. Arguments *vlabel.cex*, *vlabel.cex.min* and *vlabel.cex.max* can be used to adjust the overall, minimum and maximum expansion factor for the node labels. *max_node_cex* controls the maximum expansion factor of the node size. The size of the nodes is always proportional to the absolute occurrence of the feature in the *bstr* bootstrapping sample sets. *edge.width* and *max_edge_cex* are used for setting the edge width and expansion factor for the edges, respectively, while *ewprop* is a proportionality factor controlling the decrease of edge with decreasing frequency, a higher value of *ewprop* means a fast reduction of edge with and thus less densly packed importance graph plot. Also consider the help pages for the respective functions, to learn about the remaining function arguments.

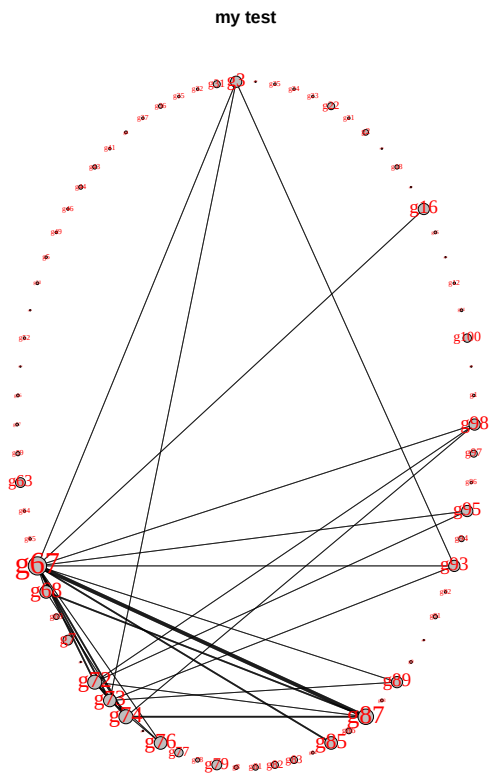

Figure 5: Importance graph generated by call to *importance_igraph* directly. Note how the edge widths are thicker but decreasing more rapidly as in figure 4, achieved by setting *max_edge_cex* and *ewprop* appropriately. Besides, the node label sizes are adjusted using *vlabel.cex*.

Session Information

The version number of R and packages loaded for generating the vignette were:

- R version 2.15.0 (2012-03-30), `x86_64-unknown-linux-gnu`
- Locale: `LC_CTYPE=en_US.UTF-8`, `LC_NUMERIC=C`, `LC_TIME=en_US.UTF-8`, `LC_COLLATE=en_US.UTF-8`, `LC_MONETARY=en_US.UTF-8`, `LC_MESSAGES=en_US.UTF-8`, `LC_PAPER=C`, `LC_NAME=C`, `LC_ADDRESS=C`, `LC_TELEPHONE=C`, `LC_MEASUREMENT=en_US.UTF-8`, `LC_IDENTIFICATION=C`
- Base packages: base, datasets, graphics, grDevices, methods, stats, utils
- Other packages: BiocInstaller 1.4.7
- Loaded via a namespace (and not attached): tools 2.15.0

References

[1] Natalia Becker, Grischa Toedt, Peter Lichter, and Axel Benner. Elastic scad as a novel penalization method for svm classification tasks in high-dimensional data. *BMC Bioinformatics*, 12:138, 2011.

[2] Miron B. Kursa and Witold R. Rudnicki. Feature selection with the boruta package. *Journal of Statistical Software*, 36(11):1–13, 9 2010.

[3] Robert Tibshirani, Trevor Hastie, Balasubramanian Narasimhan, and Gilbert Chu. Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proc Natl Acad Sci U S A*, 99(10):6567–6572, May 2002.

[4] Hao Helen Zhang, Jeongyoun Ahn, Xiaodong Lin, and Cheolwoo Park. Gene selection using support vector machines with non-convex penalty. *Bioinformatics*, 22(1):88–95, Jan 2006. SCAD paper.