Package 'AUCRF'

February 19, 2015

Type Package

Title Variable Selection with Random Forest and the Area Under the Curve

Version 1.1

Date 2012-03-19

Author Victor Urrea, M.Luz Calle

Maintainer Victor Urrea <victor.urrea@uvic.cat>

Depends R (>= 2.11.0), randomForest

Description Variable selection using Random Forest based on optimizing the area-under-the ROC curve (AUC) of the Random Forest.

License GPL (>= 2)

LazyLoad yes

Repository CRAN

Date/Publication 2012-03-19 11:12:24

NeedsCompilation no

R topics documented:

`AUCRF` *Variable Selection with Random Forest and the Area Under the Curve*

Description

AUCRF is an algorithm for variable selection using Random Forest based on optimizing the area-under-the ROC curve (AUC) of the Random Forest. The proposed strategy implements a backward elimination process based on the initial ranking of the variables.

Usage

```
AUCRF(formula, data, k0 = 1, pdel = 0.2, ranking=c(”MDG”,”MDA”), ...)
```

Arguments

`formula` an object of class `formula`: a symbolic description of the model to be fitted. The details of model specification are given in `Details`.

`data` a data frame containing the variables in the model. Dependent variable must be a binary variable defined as `factor` and codified as `1` for positives (e.g. cases) and `0` for negatives (e.g. controls).

`k0` number of remaining variables for stopping the backward elimination process. By default `k0=1`.

`pdel` fraction of remaining variables to be removed in each step. By default `pdel=0.2`. If `pdel=0`, only one variable is removed each time.

`ranking` specifies the importance measure provided by `randomForest` for ranking the variables. There are two options MDG (by default) for MeanDecreaseGini and MDA for MeanDecreaseAccuracy.

`...` optional parameters to be passed to the `randomForest` function. If no arguments are specified, default arguments of `randomForest` function will be used.

Details

The AUC-RF algorithm is described in detail in Calle et. al.(2011). The following is a summary:

Ranking and AUC of the initial set:
Perform a random forest using all predictor variables and the response, as specified in the `formula` argument, and compute the AUC of the random forest. Based on the selected measure of importance (by default MDG), obtain a ranking of predictors.

Elimination process:
Based on the variables ranking, remove the less important variables (fraction of variables specified in `pdel` argument). Perform a new random forest with the remaining variables and compute its AUC. This step is iterated until the number of remaining variables is less or equal than `k0`.

Optimal set:
The optimal set of predictive variables is considered the one giving rise to the Random Forest with the highest OOB-AUCopt. The number of selected predictors is denoted by Kopt

Value

An object of class `AUCRF`, which is a list with the following components:

`call` the original call to `AUCRF`.

`data` the `data` argument.

`ranking` the ranking of predictors based on the importance measure.

`Xopt` optimal set of predictors obtained.

`OOB-AUCopt` AUC obtained for the optimal set of predictors.

`Kopt` size of the optimal set of predictors obtained.

`AUCcurve` values of AUC obtained for each set of predictors evaluated in the elimination process.

`RFopt` the `randomForest` adjusted with the optimal set.

References

Calle ML, Urrea V, Boulesteix A-L, Malats N (2011) "AUC-RF: A new strategy for genomic profiling with Random Forest". Human Heredity. (In press)

See Also

`OptimalSet`, `AUCRFcv`, `randomForest`.

Examples

```
# load the included example dataset. This is a simulated case/control study
# data set with 4000 patients (2000 cases / 2000 controls) and 1000 SNPs,
# where the  first 10 SNPs have a direct association with the outcome:
data(exampleData)

# call AUCRF process: (it may take some time)
# fit <- AUCRF(Y~., data=exampleData)

# The result of this example is included for illustration purpose:

data(fit)
summary(fit)
plot(fit)

# Additional randomForest parameters can be included, otherwise default
# parameters of randomForest function will be used:
# fit <- AUCRF(Y~., data=exampleData, ntree=1000, nodesize=20)
```

AUCRFcv *Repeated cross validation of the AUC-RF process.*

Description

Performes a repeated cross validation analysis and computes the probability of selection for each variable.

Usage

```
AUCRFcv(x, nCV = 5, M = 20)
```

Arguments

`x`	an object of class `AUCRF`.
`nCV`	number of folds in cross validation. By default a 5-fold cross validation is performed.
`M`	number of cross validation repetitions.

Details

The results of this repeated cross validation analysis are (1) a corrected estimation of the predictive accuracy of the selected variables and (2) an estimate of the probability of selection for each variable.
The AUC-RF algorithm is exhaustively described in Calle et. al.(2011).

Value

The same `AUCRF` object passed (see `AUCRF`) as argument but updated with the following components:

`cvAUC`	mean of AUC values in test datasets of the optimal sets of predictors.
`Psel`	probability of selection of each variable as the proportion of times that is selected by AUC-RF method.

References

Calle ML, Urrea V, Boulesteix A-L, Malats N (2011) "AUC-RF: A new strategy for genomic profiling with Random Forest". Human Heredity. (In press)

See Also

`OptimalSet`, `AUCRF`, `randomForest`.

Examples

```
# Next steps take some time

# load included AUCRF result example:
# data(fit)

# call AUCRFcv process:
# fitCV <- AUCRFcv(fit)

# The result of this example is included:

data(fitCV)
summary(fitCV)
plot(fitCV)
```

`OptimalSet` *AUCRF optimal set selection.*

Description

Returns the optimal set of predictive variables selected by the AUC-RF method.

Usage

```
OptimalSet(object)
```

Arguments

`object` an object of class `AUCRF` as the result of `AUCRF` or `AUCRFcv` functions.

Value

A data.frame with the selected variables ordered by the initial ranking, their importance values (initial ranking) and, if available, the probability of selection value measured by `AUCRFcv` function.

See Also

`AUCRF`, `AUCRFcv`.

Examples

```
data(fitCV)
OptimalSet(fitCV)
```

plot.AUCRF *Plot Method for AUCRF*

Description

The plot method for `AUCRF` objects.

Usage

```
  ## S3 method for class 'AUCRF'
plot(x, which=c("auc","ranking","psel"), showOpt=TRUE, digits=4,
       maxvars=NULL, ...)
```

Arguments

`x` an object of class `AUCRF` as the result of `AUCRF` or `AUCRFcv` functions.

`which` specifies the information to plot. There are three options: (1) `"auc"` (by default) to plot the curve of AUCs in the backwards elimination process, (2) `"ranking"` to plot the importance measure in initial ranking of each variable, and (3) `"psel"` to plot the probability of selection of each variable. The `"psel"` option is only available if a cross validation is performed by `AUCRFcv` function.
For option (1), `showOpt` and `digits` arguments can be specified for more details (see below).
For options (2) and (3), the number of variables to plot and their order preference can be specified by `maxvars` and `order` arguments, respectively (see below).

`showOpt` (only applied if `"auc"` option is specified in `wich` argument). If `showOpt=TRUE`, the optimal subset is emphasised in the plot.

`digits` (only applied if `"auc"` option is specified in `wich` argument and `showOpt` or `showThres` are `TRUE`). Specifies the number of decimal digits for representing the optimal AUC in the plot.

`maxvars` (only applied if `"ranking"` or `"psel"` options are specified in `wich` argument). Number of variables to include in the plot. The specified number of variables with highest importance measure (initial ranking) will be plotted. If `maxvars=NULL` (by default) the selected variables will be plotted.
(For large number of variables, their names can be illegible in the plot)

`...` other graphical parameters (see `par`).

Examples

```
data(fitCV)

# Plotting the AUC in the AUCRF backward elimination process:
plot(fitCV)

# Plotting the probability of selection of the selected variables:
plot(fitCV, wich="psel")
```

```
# Plotting the 20 variables with highest probability of selection:
plot(fitCV, wich="psel", maxvars=20)
```

Index