Package ‘hybridEnsemble’

February 20, 2015

Type Package

Title Build, deploy and evaluate a Hybrid Ensemble

Version 0.1.1

Date 2013-03-23

Imports randomForest,kernelFactory,ada,rpart,ROCR,nnet,e1071,NMOF,GenSA,Rmalschains,pso,AUC,soma,genalg,reportr,nnls,quadprog,tabuSearch,glmnet

Author Michel Ballings, Dauwe Vercamer, and Dirk Van den Poel

Maintainer Michel Ballings <Michel.Ballings@GMail.com>

Description This package contains functions to build and deploy an ensemble consisting of six different sub-ensembles: Bagged Logistic Regressions, Random Forest, Stochastic AdaBoost, Kernel Factory, Bagged Neural Networks, Bagged Support Vector Machines. Functions to cross-validate the Hybrid Ensemble and plot and summarize the results are also provided. There is also a function to assess the importance of the predictors.

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2014-03-23 19:02:29

R topics documented:

Credit ................................................................. 2
CVhybridEnsemble .................................................. 2
hybridEnsemble ................................................... 7
hybridEnsembleNews .............................................. 12
importance.hybridEnsemble .................................. 13
plot.CVhybridEnsemble ....................................... 14
predict.hybridEnsemble ........................................ 16
summary.CVhybridEnsemble ................................... 17

Index 20
Credit

*Credit approval (Frank and Asuncion, 2010)*

**Description**

Credit contains credit card applications. The dataset has a good mix of continuous and categorical features.

**Usage**

data(Credit)

**Format**

A data frame with 690 observations and 38 predictors, and a binary criterion variable called `Response`.

**Details**

Missings are imputed and categorical variables are transformed to binary features.

**Source**


**References**

The original dataset can be downloaded at http://archive.ics.uci.edu/ml/datasets/Credit+Approval

**Examples**

data(Credit)
str(Credit)
table(Credit$Response)

---

CVhybridEnsemble

*Five times twofold cross-validation for the Hybrid Ensemble function*

**Description**

CVhybridEnsemble cross-validates (five times twofold) the Hybrid Ensemble function and computes performance statistics that can be plotted (plot.CVhybridEnsemble) and summarized (summary.CVhybridEnsemble).
Usage

CVhybridEnsemble(x = NULL, y = NULL, combine = NULL,
  eval.measure = "auc", verbose = FALSE, RF.ntree = 500, AB.iter = 500,
  AB.maxdepth = 3, KF.cp = 1, KF.rp = round(log(nrow(x), 10)),
  NN.rang = 0.1, NN.maxit = 10000, NN.size = c(5, 10, 20),
  NN.decay = c(0, 0.001, 0.01, 0.1), SV.gamma = 2^(-15:3),
  SV.cost = 2^(-5:13), SV.degree = c(2, 3), SV.kernel = c("radial",
  "sigmoid", "linear", "polynomial"), rbga.popSize = 42, rbga.iter = 500,
  rbga.mutationChance = 1/rbga.popSize, rbga.elitism = max(1,
  round(rbga.popSize * 0.05)), DEopt.nP = 20, DEopt.nG = 500,
  DEopt.F = 0.9314, DEopt.CR = 0.6938, GenSA.maxit = 500,
  GenSA.temperature = 0.5, GenSA.visiting.param = 2.7,
  GenSA.acceptance.param = -5, GenSA.max.call = 1e+07,
  malschains.popsize = 60, malschains.ls = "cmaes",
  malschains.istep = 300, malschains.effort = 0.5, malschains.alpha = 0.5,
  malschains.threshold = 1e-08, malschains.maxEvals = 500,
  psoptim.maxit = 500, psoptim.maxf = Inf, psoptim.abstol = -Inf,
  psoptim.reltol = 0, psoptim.s = 40, psoptim.k = 3, psoptim.p = 1 - 1
  /psoptim.s) ^ psoptim.k, psoptim.w = 1/(2 * log(2)), psoptim.c.p = 0.5
  +
  log(2), psoptim.c.g = 0.5 + log(2), soma.pathLength = 3,
  soma.stepLength = 0.11, soma.perturbationChance = 0.1,
  soma.minAbsoluteSep = 0, soma.minRelativeSep = 0.001,
  soma.nMigrations = 500, soma.populationSize = 10, tabu.iter = 500,
  tabu.listSize = c(5:12))

Arguments

x      A data frame of predictors. Categorical variables need to be transformed to
  binary (dummy) factors.

y      A factor of observed class labels (responses) with the only allowed values {0,1}.

combine Additional methods for combining the sub-ensembles. The simple mean, authority-
  based weighting and the single best are automatically provided since they are
  very efficient. Possible additional methods: Genetic Algorithm: "rbga", Diff-
  erential Evolutionary Algorithm: "DEopt", Generalized Simulated Annealing:
  "GenSA", Memetic Algorithm with Local Search Chains: "malschains", Par-
  ticle Swarm Optimization: "psoptim", Self-Organising Migrating Algorithm:
  "soma", Tabu Search Algorithm: "tabu", Non-negative binomial likelihood: "NNlog-
  lik", Goldfarb-Idnani Non-negative least squares: "GINNLS", Lawson-Hanson
  Non-negative least squares: "LHNNLS".

eval.measure Evaluation measure for the following combination methods: authority-based
  method, single best, "rbga", "DEopt", "GenSA", "malschains", "psoptim", "soma",
  "tabu". Default is the area under the receiver operator characteristic curve ‘auc’. The
  area under the sensitivity curve (‘sens’) and the area under the specificity curve
  (‘spec’) are also supported.

verbose TRUE or FALSE. Should information be printed to the screen while estimating
  the Hybrid Ensemble.

RF.ntr  ee Random Forest parameter. Number of trees to grow.
AB.iter  Stochastic AdaBoost parameter. Number of boosting iterations to perform.
AB.maxdepth Stochastic AdaBoost parameter. The maximum depth of any node of the final
tree, with the root node counted as depth 0.
KF.cp  Kernel Factory parameter. The number of column partitions.
KF.rp  Kernel Factory parameter. The number of row partitions.
NN.rang  Neural Network parameter. Initial random weights on [-rang, rang].
NN.maxit  Neural Network parameter. Maximum number of iterations.
NN.size  Neural Network parameter. Number of units in the single hidden layer.
NN.decay  Neural Network parameter. Weight decay.
SV.gamma  Support Vector Machines parameter. Width of the Gaussian for radial basis and
sigmoid kernel.
SV.cost  Support Vector Machines parameter. Penalty (soft margin constant).
SV.degree  Support Vector Machines parameter. Degree of the polynomial kernel.
SV.kernel  Support Vector Machines parameter. Kernels to try. Can be one or more of:
'radial','sigmoid','linear','polynomial'.
rbgaiers  Genetic Algorithm parameter. Number of iterations.
rbgamutationChance
Genetic Algorithm parameter. The chance that a gene in the chromosome mutates.
rbgaelitism  Genetic Algorithm parameter. Number of chromosomes that are kept into the
next generation.
DEopt.nG  Differential Evolutionary Algorithm parameter. Number of generations.
DEopt.F  Differential Evolutionary Algorithm parameter. Step size.
GenSA.maxit  Generalized Simulated Annealing. Maximum number of iterations.
GenSA.temperature  Generalized Simulated Annealing. Initial value for temperature.
GenSA.max.call  Generalized Simulated Annealing. Maximum number of calls of the objective
function.
malschains.ls  Memetic Algorithm with Local Search Chains parameter. Local search method.
malschains.istep  Memetic Algorithm with Local Search Chains parameter. Number of iterations
of the local search.
CVhybridEnsemble

malschains.effort
Memetic Algorithm with Local Search Chains parameter. Value between 0 and 1. The ratio between the number of evaluations for the local search and for the evolutionary algorithm. A higher effort means more evaluations for the evolutionary algorithm.

malschains.alpha
Memetic Algorithm with Local Search Chains parameter. Crossover BLX-alpha. Lower values (<0.3) reduce diversity and a higher value increases diversity.

malschains.threshold
Memetic Algorithm with Local Search Chains parameter. Threshold that defines how much improvement in the local search is considered to be no improvement.

malschains.maxEvals
Memetic Algorithm with Local Search Chains parameter. Maximum number of evaluations.

psoptim.maxit
Particle Swarm Optimization parameter. Maximum number of iterations.

psoptim.maxf
Particle Swarm Optimization parameter. Maximum number of function evaluations.

psoptim.abstol
Particle Swarm Optimization parameter. Absolute convergence tolerance.

psoptim.reltol
Particle Swarm Optimization parameter. Tolerance for restarting.

psoptim.s
Particle Swarm Optimization parameter. Swarm size.

psoptim.k
Particle Swarm Optimization parameter. Exponent for calculating number of informants.

psoptim.p
Particle Swarm Optimization parameter. Average percentage of informants for each particle.

psoptim.w
Particle Swarm Optimization parameter. Exploitation constant.

psoptim.c.p
Particle Swarm Optimization parameter. Local exploration constant.

psoptim.c.g
Particle Swarm Optimization parameter. Global exploration constant.

soma.pathLength
Self-Organising Migrating Algorithm parameter. Distance (towards the leader) that individuals may migrate.

soma.stepLength
Self-Organising Migrating Algorithm parameter. Granularity at which potential steps are evaluated.

soma.perturbationChance
Self-Organising Migrating Algorithm parameter. Probability that individual parameters are changed on any given step.

soma.minAbsoluteSep
Self-Organising Migrating Algorithm parameter. Smallest absolute difference between maximum and minimum cost function values. Below this minimum the algorithm will terminate.

soma.minRelativeSep
Self-Organising Migrating Algorithm parameter. Smallest relative difference between maximum and minimum cost function values. Below this minimum the algorithm will terminate.
soma.nMigrations

Self-Organising Migrating Algorithm parameter. Maximum number of migrations to complete.

soma.populationSize


tabu.iters

Number of iterations in the preliminary search of the algorithm.

tabu.listSize

Tabu list size.

Value

A list of class CVhybridEnsemble containing the following elements:

MEAN

For the simple mean combination method: A list containing the median and inter quartile range of the performance evaluations, the performance evaluations on each fold, and the predictions and response vectors for each fold.

AUTHORITY

For the authority combination method: A list containing the median and inter quartile range of the performance evaluations, the performance evaluations on each fold, and the predictions and response vectors for each fold.

SB

For the single best: A list containing the median and inter quartile range of the performance evaluations, the performance evaluations on each fold, and the predictions and response vectors for each fold.

eval.measure

The performance measure that was used

..and all the combination methods that are requested.

Author(s)

Authors: Michel Ballings, Dauwe Vercamer, and Dirk Van den Poel. Maintainer: <michel.ballings@gmail.com>

References

Ballings, M., Vercamer, D., Van den Poel, D., Hybrid Ensemble: Many Ensembles is Better Than One, Forthcoming.

See Also

hybridensemble, predict.hybridEnsemble, importance.hybridEnsemble, plot.CVhybridEnsemble, summary.CVhybridEnsemble

Examples

data(Credit)

## Not run:
CVhE <- CVhybridEnsemble(x=Credit[1:200, names(Credit) != 'Response'],
y=Credit$Response[1:200],
verbose=TRUE,
KF.rp=1,
RF.ntree=50,
**Binary classification with Hybrid Ensemble**

**Description**

`hybridensemble` builds an ensemble consisting of six different sub-ensembles: Bagged Logistic Regressions, Random Forest, Stochastic AdaBoost, Kernel Factory, Bagged Neural Networks, Bagged Support Vector Machines.

**Usage**

```r
hybridensemble(x = NULL, y = NULL, combine = NULL, eval.measure = "auc", verbose = FALSE, RF.ntree = 500, AB.iter = 500, AB.maxdepth = 3, KF.cp = 1, KF.rp = round(log(nrow(x), 10)), NN.rang = 0.1, NN.maxit = 10000, NN.size = c(5, 10, 20), NN.decay = c(0, 0.001, 0.01, 0.1), SV.gamma = 2^(-15:3), SV.cost = 2^(-5:13), SV.degree = c(2, 3), SV.kernel = c("radial", "sigmoid", "linear", "polynomial"), rbga.popSize = 42, rbga.iters = 300, rbga.mutationChance = 1/rbga.popSize, rbga.elitism = max(1, round(rilha.popSize * 0.05)), DEopt.nP = 20, DEopt.nG = 300, DEopt.F = 0.9314, DEopt.CR = 0.6938, GenSA.maxit = 300, GenSA.temperature = 0.5, GenSA.visiting.param = 2.7, GenSA.acceptance.param = -5, GenSA.max.call = 1e+07, malschains.popsize = 60, malschains.ls = "cmaes", malschains.istep = 300, malschains.effort = 0.5, malschains.alpha = 0.5, malschains.threshold = 1e-08, malschains.maxEvals = 300, psoptim.maxit = 300, psoptim.maxf = Inf, psoptim.abstol = -Inf, psoptim.reltol = 0, psoptim.s = 40, psoptim.k = 3, psoptim.p = 1 - (1 - 1/psoptim.s)*psoptim.k, psoptim.w = 1/(2 * log(2)), psoptim.c.p = 0.5 + log(2), psoptim.c.g = 0.5 + log(2), soma.pathLength = 3, soma.stepLength = 0.11, soma.perturbationChance = 0.1, soma.minAbsoluteSep = 0, soma.minRelativeSep = 0.001, soma.nMigrations = 300, soma.populationSize = 10, tabu.iters = 300, tabu.listSize = c(5:12))
```
**Arguments**

- **x**: A data frame of predictors. Categorical variables need to be transformed to binary (dummy) factors.

- **y**: A factor of observed class labels (responses) with the only allowed values {0,1}.


- **eval.measure**: Evaluation measure for the following combination methods: authority-based method, single best, "rbga", "DDeopt", "GenSA", "malschains", "psoptim", "soma", "tabu". Default is the area under the receiver operator characteristic curve 'auc'. The area under the sensitivity curve ('sens') and the area under the specificity curve ('spec') are also supported.

- **verbose**: TRUE or FALSE. Should information be printed to the screen while estimating the Hybrid Ensemble.

- **RF.ntree**: Random Forest parameter. Number of trees to grow.

- **AB.iter**: Stochastic AdaBoost parameter. Number of boosting iterations to perform.

- **AB.maxdepth**: Stochastic AdaBoost parameter. The maximum depth of any node of the final tree, with the root node counted as depth 0.

- **KF.cp**: Kernel Factory parameter. The number of column partitions.

- **KF.rp**: Kernel Factory parameter. The number of row partitions.

- **NN.rang**: Neural Network parameter. Initial random weights on [-rang, rang].

- **NN.maxit**: Neural Network parameter. Maximum number of iterations.

- **NN.size**: Neural Network parameter. Number of units in the single hidden layer.

- **NN.decay**: Neural Network parameter. Weight decay.

- **SV.gamma**: Support Vector Machines parameter. Width of the Gaussian for radial basis and sigmoid kernel.

- **SV.cst**: Support Vector Machines parameter. Penalty (soft margin constant).

- **SV.degree**: Support Vector Machines parameter. Degree of the polynomial kernel.

- **SV.kernel**: Support Vector Machines parameter. Kernels to try. Can be one or more of: 'radial', 'sigmoid', 'linear', 'polynomial'.


- **rbga.iters**: Genetic Algorithm parameter. Number of iterations.

- **rbga.mutationChance**: Genetic Algorithm parameter. The chance that a gene in the chromosome mutates.
rbga.elitism  Genetic Algorithm parameter. Number of chromosomes that are kept into the next generation.


dEopt.nG  Differential Evolutionary Algorithm parameter. Number of generations.

dEopt.F  Differential Evolutionary Algorithm parameter. Step size.


GenSA.maxit  Generalized Simulated Annealing. Maximum number of iterations.

GenSA.temperature  Generalized Simulated Annealing. Initial value for temperature.


GenSA.max.call  Generalized Simulated Annealing. Maximum number of calls of the objective function.


malschains.ls  Memetic Algorithm with Local Search Chains parameter. Local search method.

malschains.istep  Memetic Algorithm with Local Search Chains parameter. Number of iterations of the local search.

malschains.effort  Memetic Algorithm with Local Search Chains parameter. Value between 0 and 1. The ratio between the number of evaluations for the local search and for the evolutionary algorithm. A higher effort means more evaluations for the evolutionary algorithm.

malschains.alpha  Memetic Algorithm with Local Search Chains parameter. Crossover BLX-alpha. Lower values (<0.3) reduce diversity and a higher value increases diversity.

malschains.threshold  Memetic Algorithm with Local Search Chains parameter. Threshold that defines how much improvement in the local search is considered to be no improvement.

malschains.maxEvals  Memetic Algorithm with Local Search Chains parameter. Maximum number of evaluations.

psoptim.maxit  Particle Swarm Optimization parameter. Maximum number of iterations.

psoptim.maxf  Particle Swarm Optimization parameter. Maximum number of function evaluations.

psoptim.abstol  Particle Swarm Optimization parameter. Absolute convergence tolerance.

psoptim.reltol  Particle Swarm Optimization parameter. Tolerance for restarting.

psoptim.s  Particle Swarm Optimization parameter. Swarm size.

psoptim.k  Particle Swarm Optimization parameter. Exponent for calculating number of informants.
### Particle Swarm Optimization Parameters

- `psoptim.p`: Particle Swarm Optimization parameter. Average percentage of informants for each particle.
- `psoptim.w`: Particle Swarm Optimization parameter. Exploitation constant.
- `psoptim.c.p`: Particle Swarm Optimization parameter. Local exploration constant.
- `psoptim.c.g`: Particle Swarm Optimization parameter. Global exploration constant.

### Self-Organising Migrating Algorithm Parameters

- `soma.pathLength`: Self-Organising Migrating Algorithm parameter. Distance (towards the leader) that individuals may migrate.
- `soma.stepLength`: Self-Organising Migrating Algorithm parameter. Granularity at which potential steps are evaluated.
- `soma.perturbationChance`: Self-Organising Migrating Algorithm parameter. Probability that individual parameters are changed on any given step.
- `soma.minAbsoluteSep`: Self-Organising Migrating Algorithm parameter. Smallest absolute difference between maximum and minimum cost function values. Below this minimum the algorithm will terminate.
- `soma.minRelativeSep`: Self-Organising Migrating Algorithm parameter. Smallest relative difference between maximum and minimum cost function values. Below this minimum the algorithm will terminate.
- `soma.nMigrations`: Self-Organising Migrating Algorithm parameter. Maximum number of migrations to complete.
- `tabu.iters`: Number of iterations in the preliminary search of the algorithm.
- `tabu.listSize`: Tabu list size.

### Value

A list of class `hybridEnsemble` containing the following elements:

- **LR**: Bagged Logistic Regression model
- **LR.lambda**: Shrinkage parameter
- **RF**: Random Forest model
- **AB**: Stochastic AdaBoost model
- **KF**: Kernel Factory model
- **NN**: Neural Network model
- **SV**: Bagged Support Vector Machines model
- **SB**: A label denoting which sub-ensemble was the single best
- **weightsAUTHORITY**: The weights for the authority-based weighting method
combine Combination methods used
constants A vector denoting which predictors are constants
minima Minimum values of the predictors required for preprocessing the data for the Neural Network
scaling Range values of the predictors required for preprocessing the data for the Neural Network
NumID Vector indicating which predictors are numeric
calibratorLR The calibrator for the Bagged Logistic Regression model
calibratorRF The calibrator for the Random Forest model
calibratorAB The calibrator for the Stochastic AdaBoost model
calibratorKF The calibrator for the Kernel Factory model
calibratorNN The calibrator for the Neural Network model
calibratorSV The calibrator for the Bagged Support Vector Machines model
xVALIDATE Predictors of the validation sample
predictions The separate predictions by the six sub-ensembles
yVALIDATE Response variable of the validation sample
eval.measure The evaluation measure that was used

Author(s)
Authors: Michel Ballings, Dauwe Vercamer, and Dirk Van den Poel, Maintainer: <Michel.Ballings@GMail.com>

References
Ballings, M., Vercamer, D., Van den Poel, D., Hybrid Ensemble: Many Ensembles is Better Than One, Forthcoming.

See Also
predict.hybridEnsemble, importance.hybridEnsemble,CVhybridEnsemble,plot.CVhybridEnsemble,
summary.CVhybridEnsemble

Examples
data(Credit)

## Not run:
he <- hybridEnsemble(x=Credit[1:100, names(Credit) != 'Response'],
y=Credit$Response[1:100],
RF.ntree=50,
AB.iter=50,
NN.size=5,
NN.decay=0,
SV.gamma = 2^-15,
SV.cost = 2^-5,
SV.degree=2,
SV.kernelf='radial')

## End(Not run)

---

**hybridEnsembleNews**  
*Display the NEWS file*

**Description**

hybridEnsembleNews shows the NEWS file of the hybridEnsemble package.

**Usage**

hybridEnsembleNews()

**Value**

None.

**Author(s)**

Authors: Michel Ballings and Dirk Van den Poel, Maintainer: <Michel.Ballings@GMail.com>

**References**

Ballings, M., Vercamer, D., Van den Poel, D., Hybrid Ensemble: Many Ensembles is Better Than One, Forthcoming.

**See Also**

hybridEnsemble, predict.hybridEnsemble, importance.hybridEnsemble, CVhybridEnsemble, plot.CVhybridEnsemble, summary.CVhybridEnsemble

**Examples**

hybridEnsembleNews()
importance.hybridEnsemble

Importance method for hybridEnsemble objects

Description

Assess the importance of new data using a hybridEnsemble model. The importance is computed as follows. For each variable, compute the AUC of the model before permuting that variable and after. Next, subtract the latter from the former. This is called the decrease in AUC. If CV is greater than one, the mean is taken from all runs.

Usage

```r
## S3 method for class 'hybridEnsemble'
importance(x = NULL, xdata = NULL, ydata = NULL,
method = "MEAN", CV = 1, sort = TRUE)
```

Arguments

- `x` An object of class hybridEnsemble created by the function `hybridensemble`
- `xdata` A test data frame with the same predictors as in the training data
- `ydata` A test factor of observed class labels (responses) with the only allowed values `{0,1}`.
- `method` One of `"RBGA"` (Genetic Algorithm), `"DEOPT"` (Differential Evolution), `"GENSA"` (Generalized Simulated Annealing), `"MALSCHAINS"` (Memetic Algorithm), `"PSOPTIM"` (Particle Swarm), `"SOMA"` (Self Organizing Migrating Algorithm), `"TABU"` (Tabu Search), `"LHNLS"` (Lawson-Hanson Non-negative least squares), `"GINNLS"` (Goldfarb-Idnani Non-negative least squares), `"NNloglik"` (Non-negative binomial likelihood), `"MEAN"` (Simple Mean), `"SB"` (Single Best), `"AUTHORITY"` (Authority Based method)
- `CV` An integer indicating the number of cross-validation runs
- `sort` TRUE or FALSE. Should the predictors be sorted with the most important ones on top?

Value

A data frame with two columns: the variable name and the importance of the variable.

Author(s)

Authors: Michel Ballings, Dauwe Vercamer, and Dirk Van den Poel, Maintainer: <Michel.Ballings@GMail.com>

References

Ballings, M., Vercamer, D., Van den Poel, D., Hybrid Ensemble: Many Ensembles is Better Than One, Forthcoming.
plot.CVhybridEnsemble

See Also

hybridEnsemble, predict.hybridEnsemble, CVhybridEnsemble, plot.CVhybridEnsemble, summary.CVhybridEnsemble

Examples

data(Credit)

## Not run:
he <- hybridEnsemble(x=Credit[1:100, names(Credit) != 'Response'],
y=Credit$Response[1:100],
RF.ntree=50,
AB.iter=50,
NN.size=5,
NN.decay=0,
SV.gamma = 2^-15,
SV.cost = 2^-5,
SV.degree=2,
SV.kernel='radial')

importance(he,
    xdata=Credit[1:100, names(Credit) != 'Response'],
ydata=Credit$Response[1:100])

## End(Not run)

plot.CVhybridEnsemble  Plot the performance of the cross-validated Hybrid Ensemble

Description

This function plots the averaged ROC curve per combination method or the median predictive performance (Area under the ROC, sensitivity or specificity curve depending on what was used in the cvhybridensemble function).

Usage

## S3 method for class 'CVhybridEnsemble'
plot(x, y = NULL, ROCcurve = FALSE,
     averaging = "threshold", ...)

Arguments

x  An object of class CVhybridEnsemble
y  Not used
ROCcurve  TRUE or FALSE. Should the ROC curve be plotted or the median predictive performances?
averaging  For the ROC curve: "threshold" averaging, "horizontal" averaging, or "vertical" averaging.
...  Not used
Details

In the legend: 'RBGA' (Genetic Algorithm),'DEOPT' (Differential Evolution),'GENSA' (Generalized Simulated Annealing),'MALSCHAINS' (Memetic Algorithm),'PSOPTIM' (Particle Swarm),'SOMA' (Self Organizing Migrating Algorithm),'TABU' (Tabue Search),'LHNLS' (Lawson-Hanson Non-negative least squares),'GINNLS' (Goldfarb-Idnani Non-negative least squares),'NNloglik' (Non-negative binomial likelihood),'MEAN' (Simple Mean),'SB' (Single Best),'AUTHORITY' (Authority Based method). The letters next to SB denote the single best for all cross-validation runs: R=Random Forest, S=Bagged Support Vector Machines, K=Kernel Factory, A=AdaBoost, L=Bagged Logistic Regression, N=Bagged Neural Networks

Author(s)

Authors: Michel Ballings and Dirk Van den Poel, Maintainer: <michel.ballings@gmail.com>

References

Ballings, M., Vercamer, D., Van den Poel, D., Hybrid Ensemble: Many Ensembles is Better Than One, Forthcoming.

See Also

`hybridEnsemble`, `predict.hybridEnsemble`, `importance.hybridEnsemble`, `CVhybridEnsemble`, `summary.CVhybridEnsemble`

Examples

data(Credit)

## Not run:
CVhE <- CVhybridEnsemble(x=Credit[1:200],names(Credit) != 'Response'],
y=Credit$Response[1:200],
verbose=TRUE,
RF_ntree=50,
KF_rp=1,
AB_iter=50,
NN_size=5,
NN_decay=0,
SV_gamma = 2^-15,
SV_cost = 2^-5,
SV_degree=2,
SV_kernel='radial')

plot(x=CVhE,ROCcurve= FALSE)
plot(x=CVhE,ROCcurve= TRUE)

## End(Not run)
predict.hybridEnsemble

Predict method for hybridEnsemble objects

Description

Prediction of new data using a hybridEnsemble model.

Usage

```r
## S3 method for class 'hybridEnsemble'
predict(object, newdata, verbose = FALSE, ...)
```

Arguments

- `object`: An object of class hybridEnsemble created by the function `hybridEnsemble`
- `newdata`: A data frame with the same predictors as in the training data
- `verbose`: TRUE or FALSE. Should information be printed to the screen
- `...`: Not currently used

Value

A list containing the following vectors:

- `predMEAN`: Predictions combined by the simple mean
- `predSB`: Predictions by the single best
- `predAUTHORITY`: Predictions combined by authority

..and all the combination methods that are requested in the `hybridEnsemble` function.

Author(s)

Authors: Michel Ballings, Dauwe Vercamer, and Dirk Van den Poel, Maintainer: <Michel.Ballings@GMail.com>

References

Ballings, M., Vercamer, D., Van den Poel, D., Hybrid Ensemble: Many Ensembles is Better Than One, Forthcoming.

See Also

`hybridEnsemble`, `CVhybridEnsemble`, `importance.hybridEnsemble`, `plot.CVhybridEnsemble`, `summary.CVhybridEnsemble`
Examples

data(Credit)

## Not run:
he <- hybridEnsemble(x=Credit[,1:100], names(Credit) != 'Response'),
y=Credit$Response[1:100],
RF.ntree=50,
AB.iter=50,
NN.size=5,
NN.decay=0,
SV.gamma = 2^-15,
SV.cost = 2^-5,
SV.degree=2,
SV.kernel='radial')
predictions <- predict(he, newdata=Credit[,100], names(Credit) != 'Response'))

## End(Not run)

summary.CVhybridEnsemble

Summarize the performance of the cross-validated Hybrid Ensemble

Description

This function produces summary results per combination method.

Usage

## S3 method for class 'CVhybridEnsemble'
summary(object, name = '', stat = "median",
Latex = FALSE, toppart = FALSE, bottompart = FALSE, ...)

Arguments

object An object of class CVhybridEnsemble
name Name of the dataset. Default is blank.
stat 'median' or 'IQR' (inter quartile range) of the performance measure used in the
CVhybridEnsemble object
Latex TRUE or FALSE. If true LateX code is printed to the screen. Otherwise a data
frame.
toppart TRUE or FALSE. For the LateX table. Should the top part of the table be printed.
Useful for concatenating multiple runs of the summary function (see examples).
bottompart TRUE or FALSE. For the LateX table. Should the bottom part of the table be
printed. Useful for concatenating multiple runs of the summary function (see examples).
... Not used
Details
In the output: 'RBGA' (Genetic Algorithm),'DEOPT' (Differential Evolution),'GENSA' (General-
ized Simulated Annealing),'MALSCAHEADS' (Memetic Algorithm),'PSOPTIMA' (Particle Swarm),'SOMA' (Self Organizing Migrating Algorithm),'TABU' (Tabue Search),'LHNNLS' (Lawson-Hanson Non-
negative least squares),'GINNLS' (Goldfarb-Idnani Non-negative least squares),'NNloglik' (Non-
negative binomial likelihood),'MEAN' (Simple Mean),'SB' (Single Best),'AUTHORITY' (Author-
ity Based method). SB names denote the single best for all cross-validation runs: R=Random Forest, S=Bagged Support Vector Machines, K=Kernel Factory, A=AdaBoost, L=Bagged Logistic Regres-
sion, N=Bagged Neural Networks

Author(s)
Authors: Michel Ballings and Dirk Van den Poel, Maintainer: <michel.Ballings@GMail.com>

References
Ballings, M., Vercamer, D., Van den Poel, D., Hybrid Ensemble: Many Ensembles is Better Than
One, Forthcoming.

See Also
hybridEnsemble, predict.hybridEnsemble, importance.hybridEnsemble, CVhybridEnsemble,
plot.CVhybridEnsemble

Examples
data(Credit)

## Not run:
CVhE <- CVhybridEnsemble(x=Credit[1:200],names(Credit) != 'Response'],
y=Credit$Response[1:200],
verbose=TRUE,
RF.ntree=50,
KF.rp=1,
AB.iter=50,
NN.size=5,
NN.decay=8,
SV.gamma = 2^-15,
SV.cost = 2^-5,
SV.degree=2,
SV.kernel='radial')

summary(object=CVhE,stat='median')
summary(object=CVhE,stat='IQR')

#LaTeX table
#This code example shows how toppart and bottompart can be convenient if you want
#to concatenate multiple datasets (here six time the same dataset).
#Paste the output of this code in your LateX document:
cat(
    summary(object=CVhE ,name="Credit", LaTeX=TRUE, toppart=TRUE),
summary.CVhybridEnsemble

summary(object=CVhE, name="Credit", LaTeX=TRUE),
summary(object=CVhE, name="Credit", LaTeX=TRUE),
summary(object=CVhE, name="Credit", LaTeX=TRUE),
summary(object=CVhE, name="Credit", LaTeX=TRUE),
summary(object=CVhE, name="Credit", LaTeX=TRUE, bottompart=TRUE)  

## End(Not run)
Index

*Topic **datasets**
  Credit, 2

Credit, 2
CVHybridEnsemble, 2, 11, 12, 14–16, 18

hybridEnsemble, 6, 7, 12, 14–16, 18
hybridEnsembleNews, 12

importance.hybridEnsemble, 6, 11, 12, 13, 15, 16, 18

plot.CVHybridEnsemble, 2, 6, 11, 12, 14, 14, 16, 18
predict.hybridEnsemble, 6, 11, 12, 14, 15, 16, 18

summary.CVHybridEnsemble, 2, 6, 11, 12, 14–16, 17