Package: gfboost (via r-universe)

August 20, 2024

Type Package

Title Gradient-Free Gradient Boosting

Version 0.1.1

Maintainer Tino Werner <tino.werner1@uni-oldenburg.de>

Description Implementation of routines of the author's PhD thesis on gradient-free Gradient Boosting (Werner, Tino (2020) ``Gradient-Free Gradient Boosting'', URL '<https://oops.uni-oldenburg.de/id/eprint/4290>').

Depends mvtnorm, pcaPP, mboost

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 7.1.1

VignetteBuilder knitr

Suggests testthat, knitr, rmarkdown

NeedsCompilation no

Author Tino Werner [aut, cre, cph]

Repository CRAN

Date/Publication 2022-01-07 09:10:01 UTC

Contents

CMB *CMB aggregation function*

Description

Aggregates the selection frequencies of multiple SingBoost models. May be used with caution since there are not yet recommendations about good hyperparameters.

Usage

```
CMB(
  D,
  nsing,
  Bsing = 1,
  alpha = 1,
  singfam = Gaussian(),
  evalfam = Gaussian(),
  sing = FALSE,
  M = 10,
  m_iter = 100,
  kap = 0.1,
  LS = FALSE,
  best = 1,
  wagg,
  robagg = FALSE,
  lower = 0,
  ...
)
```

Arguments

D	Data matrix. Has to be an $n \times (p+1)-$dimensional data frame in the format (X, Y). The $X-$part must not contain an intercept column containing only ones since this column will be added automatically.
nsing	Number of observations (rows) used for the SingBoost submodels.
Bsing	Number of subsamples based on which the SingBoost models are validated. Default is 1. Not to confuse with parameter B for the Stability Selection.
alpha	Optional real number in $]0, 1]$. Defines the fraction of best SingBoost models used in the aggregation step. Default is 1 (use all models).

singfam	A SingBoost family. The SingBoost models are trained based on the corresponding loss function. Default is Gaussian() (squared loss).
evalfam	A SingBoost family. The SingBoost models are validated according to the corresponding loss function. Default is Gaussian() (squared loss).
sing	If sing=FALSE and the singfam family is a standard Boosting family that is contained in the package mboost, the CMB aggregation procedure is executed for the corresponding standard Boosting models.
M	An integer between 2 and m_iter. Indicates that in every $M-$th iteration, a singular iteration will be performed. Default is 10.
m_iter	Number of SingBoost iterations. Default is 100.
kap	Learning rate (step size). Must be a real number in $]0, 1]$. Default is 0.1 It is recommended to use a value smaller than 0.5.
LS	If a singfamily object that is already provided by mboost is used, the respective Boosting algorithm will be performed in the singular iterations if Ls is set to TRUE. Default is FALSE.
best	Needed in the case of localized ranking. The parameter K of the localized ranking loss will be computed by $best \cdot n$ (rounded to the next larger integer). Warning: If a parameter K is inserted into the LocRank family, it will be ignored when executing SingBoost.
wagg	Type of row weight aggregation. 'weights1' indicates that the selection frequencies of the (best) SingBoost models are averaged. 'weights2' respects the validation losses for each model and downweights the ones with higher validation losses.
robagg	Optional. If setting robagg=TRUE, the best SingBoost models are ignored when executing the aggregation to avoid inlier effects. Only reasonable in combination with lower.
lower	Optional argument. Only reasonable when setting robagg=TRUE. lower is a real number in $[0, 1[$ (a rather small number is recommended) and indicates that the aggregation ignores the SingBoost models with the best performances to avoid possible inlier effects.
...	Optional further arguments

Details

SingBoost is designed to detect variables that standard Boosting procedures may not but which may be relevant w.r.t. the target loss function. However, one may try to stabilize this "singular part" of the column measure by aggregating several SingBoost models in the sense that they are evaluated on a validation set and that the selection frequencies are averaged, maybe in a weighted manner according to the validation losses. Warning: This procedure does not replace a Stability Selection!

Value

Column measure	Aggregated column measure as $(p + 1)-$dimensional vector.
Selected variables	Names of the variables with positive aggregated column measure.

`Variables names`
: Names of all variables including the intercept.

`Row measure`
: Aggregated row measure as $n-$dimensional vector.

References

Werner, T., Gradient-Free Gradient Boosting, PhD Thesis, Carl von Ossietzky University Oldenburg, 2020

Examples

```
firis<-as.formula(Sepal.Length~.)
Xiris<-model.matrix(firis,iris)
Diris<-data.frame(Xiris[,-1],iris$Sepal.Length)
colnames(Diris)[6]<-"Y"
set.seed(19931023)
cmb1<-CMB(Diris,nsing=100,Bsing=50,alpha=0.8,singfam=Rank(),
evalfam=Rank(),sing=TRUE,M=10,m_iter=100,
kap=0.1,LS=TRUE,wagg='weights1',robagg=FALSE,lower=0)
cmb1
set.seed(19931023)
cmb2<-CMB(Diris,nsing=100,Bsing=50,alpha=0.8,singfam=Rank(),
evalfam=Rank(),sing=TRUE,M=2,m_iter=100,
kap=0.1,LS=TRUE,wagg='weights1',robagg=FALSE,lower=0)
cmb2[[1]]
set.seed(19931023)
cmb3<-CMB(Diris,nsing=100,Bsing=50,alpha=0.8,singfam=Rank(),
evalfam=Rank(),sing=TRUE,M=10,m_iter=100,
kap=0.1,LS=TRUE,wagg='weights2',robagg=FALSE,lower=0)
cmb3[[1]]
```

CMB.stabpath *CMB stability paths*

Description

Draws a Stability plot for CMB.

Usage

```
CMB.stabpath(
  D,
  nsing,
  Bsing = 1,
  alpha = 1,
  singfam = Gaussian(),
  evalfam = Gaussian(),
  sing = FALSE,
  Mseq,
```

```
  m_iter = 100,
  kap = 0.1,
  LS = FALSE,
  best = 1,
  wagg,
  robagg = FALSE,
  lower = 0,
  B,
  ncmb,
  ...
)
```

Arguments

D	Data matrix. Has to be an $n \times (p+1)-$dimensional data frame in the format (X, Y). The $X-$part must not contain an intercept column containing only ones since this column will be added automatically.
nsing	Number of observations (rows) used for the SingBoost submodels.
Bsing	Number of subsamples based on which the SingBoost models are validated. Default is 1. Not to confuse with parameter B for the Stability Selection.
alpha	Optional real number in $]0, 1]$. Defines the fraction of best SingBoost models used in the aggregation step. Default is 1 (use all models).
singfam	A SingBoost family. The SingBoost models are trained based on the corresponding loss function. Default is Gaussian() (squared loss).
evalfam	A SingBoost family. The SingBoost models are validated according to the corresponding loss function. Default is Gaussian() (squared loss).
sing	If sing=FALSE and the singfam family is a standard Boosting family that is contained in the package mboost, the CMB aggregation procedure is executed for the corresponding standard Boosting models.
Mseq	A vector of different values for M.
m_iter	Number of SingBoost iterations. Default is 100.
kap	Learning rate (step size). Must be a real number in $]0, 1]$. Default is 0.1 It is recommended to use a value smaller than 0.5.
LS	If a singfamily object that is already provided by mboost is used, the respective Boosting algorithm will be performed in the singular iterations if Ls is set to TRUE. Default is FALSE.
best	Needed in the case of localized ranking. The parameter K of the localized ranking loss will be computed by $best \cdot n$ (rounded to the next larger integer). Warning: If a parameter K is inserted into the LocRank family, it will be ignored when executing SingBoost.
wagg	Type of row weight aggregation. 'weights1' indicates that the selection frequencies of the (best) SingBoost models are averaged. 'weights2' respects the validation losses for each model and downweights the ones with higher validation losses.

robagg	Optional. If setting robagg=TRUE, the best SingBoost models are ignored when executing the aggregation to avoid inlier effects. Only reasonable in combination with lower.
lower	Optional argument. Only reasonable when setting robagg=TRUE. lower is a real number in $[0, 1[$ (a rather small number is recommended) and indicates that the aggregation ignores the SingBoost models with the best performances to avoid possible inlier effects.
B	Number of subsamples of size n_{cmb} of the training data for CMB aggregation.
ncmb	Number of samples used for CMB. Integer that must be smaller than the number of samples in Dtrain.
...	Optional further arguments

Value

relev	List of relevant variables (represented as their column number).
ind	Vector of relevant variables (represented as their column number).

References

Werner, T., Gradient-Free Gradient Boosting, PhD Thesis, Carl von Ossietzky University Oldenburg, 2020

CMB.Stabsel *Loss-adapted Stability Selection*

Description

Workhorse function for the Stability Selection variant where either a grid of thresholds or a grid of cardinalities is given so that the Boosting models are evaluated on a validation set according to all elements of the respective grid. The model which performs best is finally selected as stable model.

Usage

```
CMB.Stabsel(
  Dtrain,
  nsing,
  Bsing = 1,
  B = 100,
  alpha = 1,
  singfam = Gaussian(),
  evalfam = Gaussian(),
  sing = FALSE,
  M = 10,
  m_iter = 100,
  kap = 0.1,
  LS = FALSE,
```

```
  best = 1,
  wagg,
  gridtype,
  grid,
  Dvalid,
  ncmb,
  robagg = FALSE,
  lower = 0,
  singcoef = FALSE,
  Mfinal,
  ...
)
```

Arguments

Dtrain	Data matrix. Has to be an $n \times (p+1)-$dimensional data frame in the format (X, Y). The $X-$part must not contain an intercept column containing only ones since this column will be added automatically.
nsing	Number of observations (rows) used for the SingBoost submodels.
Bsing	Number of subsamples based on which the SingBoost models are validated. Default is 1. Not to confuse with parameter B for the Stability Selection.
B	Number of subsamples based on which the CMB models are validated. Default is 100. Not to confuse with Bsing for CMB.
alpha	Optional real number in $]0, 1]$. Defines the fraction of best SingBoost models used in the aggregation step. Default is 1 (use all models).
singfam	A SingBoost family. The SingBoost models are trained based on the corresponding loss function. Default is Gaussian() (squared loss).
evalfam	A SingBoost family. The SingBoost models are validated according to the corresponding loss function. Default is Gaussian() (squared loss).
sing	If sing=FALSE and the singfam family is a standard Boosting family that is contained in the package mboost, the CMB aggregation procedure is executed for the corresponding standard Boosting models.
M	An integer between 2 and m_iter. Indicates that in every $M-$th iteration, a singular iteration will be performed. Default is 10.
m_iter	Number of SingBoost iterations. Default is 100.
kap	Learning rate (step size). Must be a real number in $]0, 1]$. Default is 0.1 It is recommended to use a value smaller than 0.5.
LS	If a singfamily object that is already provided by mboost is used, the respective Boosting algorithm will be performed in the singular iterations if Ls is set to TRUE. Default is FALSE.
best	Needed in the case of localized ranking. The parameter K of the localized ranking loss will be computed by $best \cdot n$ (rounded to the next larger integer). Warning: If a parameter K is inserted into the LocRank family, it will be ignored when executing SingBoost.

wagg	Type of row weight aggregation. 'weights1' indicates that the selection frequencies of the (best) SingBoost models are averaged. 'weights2' respects the validation losses for each model and downweights the ones with higher validation losses.
gridtype	Choose between 'pigrid' and 'qgrid'.
grid	The grid for the thresholds (in $]0, 1]$) or the numbers of final variables (positive integers).
Dvalid	Validation data for selecting the optimal element of the grid and with it the best corresponding model.
ncmb	Number of samples used for CMB. Integer that must be smaller than the number of samples in Dtrain and higher than nsing.
robagg	Optional. If setting robagg=TRUE, the best SingBoost models are ignored when executing the aggregation to avoid inlier effects. Only reasonable in combination with lower.
lower	Optional argument. Only reasonable when setting robagg=TRUE. lower is a real number in $[0, 1[$ (a rather small number is recommended) and indicates that the aggregation ignores the SingBoost models with the best performances to avoid possible inlier effects.
singcoef	Default is FALSE. Then the coefficients for the candidate stable models are computed by standard linear regression (provided that the number of columns is smaller than the number of samples in the training set for each grid element). If set to TRUE, the coefficients are computed by SingBoost.
Mfinal	Optional. Necessary if singcoef=TRUE to determine the frequency of singular iterations in the SingBoost models.
...	Optional further arguments

Details

The Stability Selection in the packages stabs and mboost requires to fix two of three parameters which are the per-family error rate, the threshold and the number of variables which have to be selected in each model. Our Stability Selection is based on another idea. We also train Boosting models on subsamples but we use a validation step to determine the size of the optimal model. More precisely, if 'pigrid' is used as gridtype, the corresponding stable models for each threshold are computed by selecting all variables whose aggregated selection frequency exceeds the threshold. Then, these candidate stable models are validated according to the target loss function (inserted through evalfam) and the optimal one is finally selected. If 'qgrid' is used as gridtype, a vector of positive integers has to be entered instead of a vector of thresholds. The candidate stable models then consist of the best variables ordered by their aggregated selection frequencies, respectively. The validation step is the same.

Value

colind.opt	The column numbers of the variables that form the best stable model as a vector.
coeff.opt	The coefficients corresponding to the optimal stable model as a vector.
aggnu	Aggregated empirical column measure (i.e., selection frequencies) as a vector.
aggzeta	Aggregated empirical row measure (i.e., row weights) as a vector.

References

Werner, T., Gradient-Free Gradient Boosting, PhD Thesis, Carl von Ossietzky University Oldenburg, 2020

T. Hothorn, P. Bühlmann, T. Kneib, M. Schmid, and B. Hofner. mboost: Model-Based Boosting, 2017

B. Hofner and T. Hothorn. stabs: Stability Selection with Error Control, 2017.

B. Hofner, L. Boccuto, and M. Göker. Controlling false discoveries in high-dimensional situations: Boosting with stability selection. BMC Bioinformatics, 16(1):144, 2015.

N. Meinshausen and P. Bühlmann. Stability selection. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 72(4):417–473, 2010.

`CMB3S` *Column Measure Boosting with SingBoost and Stability Selection (CMB-3S)*

Description

Executes CMB and the loss-based Stability Selection.

Usage

```
CMB3S(
  Dtrain,
  nsing,
  Bsing = 1,
  B = 100,
  alpha = 1,
  singfam = Gaussian(),
  evalfam = Gaussian(),
  sing = FALSE,
  M = 10,
  m_iter = 100,
  kap = 0.1,
  LS = FALSE,
  best = 1,
  wagg,
  gridtype,
  grid,
  Dvalid,
  ncmb,
  robagg = FALSE,
  lower = 0,
  singcoef = FALSE,
  Mfinal = 10,
  ...
)
```

Arguments

Dtrain	Data matrix. Has to be an $n \times (p+1)-$dimensional data frame in the format (X, Y). The $X-$part must not contain an intercept column containing only ones since this column will be added automatically.
nsing	Number of observations (rows) used for the SingBoost submodels.
Bsing	Number of subsamples based on which the SingBoost models are validated. Default is 1. Not to confuse with parameter B for the Stability Selection.
B	Number of subsamples based on which the CMB models are validated. Default is 100. Not to confuse with Bsing for CMB.
alpha	Optional real number in $]0, 1]$. Defines the fraction of best SingBoost models used in the aggregation step. Default is 1 (use all models).
singfam	A SingBoost family. The SingBoost models are trained based on the corresponding loss function. Default is Gaussian() (squared loss).
evalfam	A SingBoost family. The SingBoost models are validated according to the corresponding loss function. Default is Gaussian() (squared loss).
sing	If sing=FALSE and the singfam family is a standard Boosting family that is contained in the package mboost, the CMB aggregation procedure is executed for the corresponding standard Boosting models.
M	An integer between 2 and m_iter. Indicates that in every $M-$th iteration, a singular iteration will be performed. Default is 10.
m_iter	Number of SingBoost iterations. Default is 100.
kap	Learning rate (step size). Must be a real number in $]0, 1]$. Default is 0.1 It is recommended to use a value smaller than 0.5.
LS	If a singfamily object that is already provided by mboost is used, the respective Boosting algorithm will be performed in the singular iterations if Ls is set to TRUE. Default is FALSE.
best	Needed in the case of localized ranking. The parameter K of the localized ranking loss will be computed by $best \cdot n$ (rounded to the next larger integer). Warning: If a parameter K is inserted into the LocRank family, it will be ignored when executing SingBoost.
wagg	Type of row weight aggregation. 'weights1' indicates that the selection frequencies of the (best) SingBoost models are averaged. 'weights2' respects the validation losses for each model and downweights the ones with higher validation losses.
gridtype	Choose between 'pigrid' and 'qgrid'.
grid	The grid for the thresholds (in $]0, 1]$) or the numbers of final variables (positive integers).
Dvalid	Validation data for selecting the optimal element of the grid and with it the best corresponding model.
ncmb	Number of samples used for CMB. Integer that must be smaller than the number of samples in Dtrain.

robagg	Optional. If setting robagg=TRUE, the best SingBoost models are ignored when executing the aggregation to avoid inlier effects. Only reasonable in combination with lower.
lower	Optional argument. Only reasonable when setting robagg=TRUE. lower is a real number in $[0, 1[$ (a rather small number is recommended) and indicates that the aggregation ignores the SingBoost models with the best performances to avoid possible inlier effects.
singcoef	Default is FALSE. Then the coefficients for the candidate stable models are computed by standard linear regression (provided that the number of columns is smaller than the number of samples in the training set for each grid element). If set to TRUE, the coefficients are computed by SingBoost.
Mfinal	Optional. Necessary if singcoef=TRUE to determine the frequency of singular iterations in the SingBoost models.
...	Optional further arguments

Details

See CMB and CMB.Stabsel.

Value

Final coefficients
: The coefficients corresponding to the optimal stable model as a vector.

Stable column measure
: Aggregated empirical column measure (i.e., selection frequencies) as a vector.

Selected columns
: The column numbers of the variables that form the best stable model as a vector.

Used row measure
: Aggregated empirical row measure (i.e., row weights) as a vector.

References

Werner, T., Gradient-Free Gradient Boosting, PhD Thesis, Carl von Ossietzky University Oldenburg, 2020

T. Hothorn, P. Bühlmann, T. Kneib, M. Schmid, and B. Hofner. mboost: Model-Based Boosting, 2017

B. Hofner and T. Hothorn. stabs: Stability Selection with Error Control, 2017.

B. Hofner, L. Boccuto, and M. Göker. Controlling false discoveries in high-dimensional situations: Boosting with stability selection. BMC Bioinformatics, 16(1):144, 2015.

N. Meinshausen and P. Bühlmann. Stability selection. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 72(4):417–473, 2010.

Examples

```
firis<-as.formula(Sepal.Length~.)
Xiris<-model.matrix(firis,iris)
Diris<-data.frame(Xiris[,-1],iris$Sepal.Length)
```

```
colnames(Diris)[6]<-"Y"
set.seed(19931023)
ind<-sample(1:150,120,replace=FALSE)
Dtrain<-Diris[ind,]
Dvalid<-Diris[-ind,]
set.seed(19931023)
cmb3s<-CMB3S(Dtrain,nsing=120,Dvalid=Dvalid,ncmb=120,Bsing=1,B=1,alpha=1,singfam=Gaussian()
,evalfam=Gaussian(),sing=FALSE,M=10,m_iter=100,kap=0.1,LS=FALSE,wagg='weights1',
gridtype='pigrid',grid=seq(0.8,0.9,1),robagg=FALSE,lower=0,singcoef=TRUE,Mfinal=10)
cmb3s$Fin
cmb3s$Stab
cmb3s$Sel
glmres4<-glmboost(Sepal.Length~.,iris[ind,])
coef(glmres4)
set.seed(19931023)
cmb3s1<-CMB3S(Dtrain,nsing=80,Dvalid=Dvalid,ncmb=100,Bsing=10,B=100,alpha=0.5,singfam=Gaussian(),
evalfam=Gaussian(),sing=FALSE,M=10,m_iter=100,kap=0.1,LS=FALSE,wagg='weights1',gridtype='pigrid',
grid=seq(0.8,0.9,1),robagg=FALSE,lower=0,singcoef=TRUE,Mfinal=10)
cmb3s1$Fin
cmb3s1$Stab
## This will may take around a minute
set.seed(19931023)
cmb3s2<-CMB3S(Dtrain,nsing=80,Dvalid=Dvalid,ncmb=100,Bsing=10,B=100,alpha=0.5,singfam=Rank(),
evalfam=Rank(),sing=TRUE,M=10,m_iter=100,kap=0.1,LS=TRUE,wagg='weights2',gridtype='pigrid',
grid=seq(0.8,0.9,1),robagg=FALSE,lower=0,singcoef=TRUE,Mfinal=10)
cmb3s2$Fin
cmb3s2$Stab
set.seed(19931023)
cmb3s3<-CMB3S(Dtrain,nsing=80,Dvalid=Dvalid,ncmb=100,Bsing=10,B=100,alpha=0.5,singfam=Huber(),
evalfam=Huber(),sing=FALSE,M=10,m_iter=100,kap=0.1,LS=FALSE,wagg='weights2',gridtype='pigrid',
grid=seq(0.8,0.9,1),robagg=FALSE,lower=0,singcoef=FALSE,Mfinal=10)
cmb3s3$Fin
cmb3s3$Stab
```

CV.CMB3S *Cross-validated version of CMB-3S*

Description

Cross-validates the whole loss-based Stability Selection by aggregating several stable models according to their performance on validation sets. Also computes a cross-validated test loss on a disjoint test set.

Usage

```
CV.CMB3S(
  D,
  nsing,
  Bsing = 1,
```

```
  B = 100,
  alpha = 1,
  singfam = Gaussian(),
  evalfam = Gaussian(),
  sing = FALSE,
  M = 10,
  m_iter = 100,
  kap = 0.1,
  LS = FALSE,
  best = 1,
  wagg,
  gridtype,
  grid,
  ncmb,
  CVind,
  targetfam = Gaussian(),
  print = TRUE,
  robagg = FALSE,
  lower = 0,
  singcoef = FALSE,
  Mfinal = 10,
  ...
)
```

Arguments

D	Data matrix. Has to be an $n \times (p+1)-$dimensional data frame in the format (X, Y). The $X-$part must not contain an intercept column containing only ones since this column will be added automatically.
nsing	Number of observations (rows) used for the SingBoost submodels.
Bsing	Number of subsamples based on which the SingBoost models are validated. Default is 1. Not to confuse with parameter B for the Stability Selection.
B	Number of subsamples based on which the CMB models are validated. Default is 100. Not to confuse with Bsing for CMB.
alpha	Optional real number in $]0, 1]$. Defines the fraction of best SingBoost models used in the aggregation step. Default is 1 (use all models).
singfam	A SingBoost family. The SingBoost models are trained based on the corresponding loss function. Default is Gaussian() (squared loss).
evalfam	A SingBoost family. The SingBoost models are validated according to the corresponding loss function. Default is Gaussian() (squared loss).
sing	If sing=FALSE and the singfam family is a standard Boosting family that is contained in the package mboost, the CMB aggregation procedure is executed for the corresponding standard Boosting models.
M	An integer between 2 and m_iter. Indicates that in every $M-$th iteration, a singular iteration will be performed. Default is 10.
m_iter	Number of SingBoost iterations. Default is 100.

kap	Learning rate (step size). Must be a real number in $]0, 1]$. Default is 0.1 It is recommended to use a value smaller than 0.5.
LS	If a singfamily object that is already provided by mboost is used, the respective Boosting algorithm will be performed in the singular iterations if Ls is set to TRUE. Default is FALSE.
best	Needed in the case of localized ranking. The parameter K of the localized ranking loss will be computed by $best \cdot n$ (rounded to the next larger integer). Warning: If a parameter K is inserted into the LocRank family, it will be ignored when executing SingBoost.
wagg	Type of row weight aggregation. 'weights1' indicates that the selection frequencies of the (best) SingBoost models are averaged. 'weights2' respects the validation losses for each model and downweights the ones with higher validation losses.
gridtype	Choose between 'pigrid' and 'qgrid'.
grid	The grid for the thresholds (in $]0, 1]$) or the numbers of final variables (positive integers).
ncmb	Number of samples used for CMB. Integer that must be smaller than the number of samples in D.
CVind	A list where each element contains a vector on length n (number of samples in the data matrix D) which contains the strings 'tr' (training set), 'v' (validation set) and 'te' (test set). This list can be easily generated using the function random_CVind.
targetfam	Target loss. Should be the same family as evalfam. Default is Gaussian() (squared loss).
print	If set to TRUE (default), the number of the currently finished outer cross-validation loop is printed.
robagg	Optional. If setting robagg=TRUE, the best SingBoost models are ignored when executing the aggregation to avoid inlier effects. Only reasonable in combination with lower.
lower	Optional argument. Only reasonable when setting robagg=TRUE. lower is a real number in $[0, 1[$ (a rather small number is recommended) and indicates that the aggregation ignores the SingBoost models with the best performances to avoid possible inlier effects.
singcoef	Default is FALSE. Then the coefficients for the candidate stable models are computed by standard linear regression (provided that the number of columns is smaller than the number of samples in the training set for each grid element). If set to TRUE, the coefficients are computed by SingBoost.
Mfinal	Optional. Necessary if singcoef=TRUE to determine the frequency of singular iterations in the SingBoost models.
...	Optional further arguments

Details

In CMB3S, a validation set is given based on which the optimal stable model is chosen. The CV.CMB3S function adds an outer cross-validation step such that both the training and the validation data sets

(and optionally the test data sets) are chosen randomly by disjointly dividing the initial data set. The aggregated stable models form an "ultra-stable" model. It is strongly recommended to use this function is a parallelized manner due to huge computation time.

Value

`Cross-validated loss`
: A vector containing the cross-validated test losses.

`Ultra-stable column measure`
: A vector containing the aggregated selection frequencies of the stable models.

References

Werner, T., Gradient-Free Gradient Boosting, PhD Thesis, Carl von Ossietzky University Oldenburg, 2020

---

genDataFromExamples *Data generation*

Description

Auxiliary function for generating simple artificial data sets with normally distributed coefficients and regressors. Note that we only report this function for reproducibility of the simulations from the PhD thesis of the author.

Usage

```
genDataFromExamples(
  p,
  n,
  s = 1,
  xmean = 0,
  betamean = 0,
  betasd = 1,
  snr = 2,
  rho = 0
)
```

Arguments

`p`	Number of variables (columns).
`n`	Number of observations (rows).
`s`	Sparsity. Real number between 0 and 1. `s=1` (default) leads to a coefficient vector without zero entries.
`xmean`	Mean of each of the normally distributed columns. Default is 0.
`betamean`	Mean of each of the normally distributed coefficients. Default is 0.

betasd	Standard deviation of the normally distributed coefficients. Default is 1.
snr	Signal to noise ratio. Real number greater than zero. Default is 2.
rho	Parameter for a Toeplitz covariance structure of the regressors. Real number between -1 and 1. Default is 0 which corresponds to uncorrelated columns.

Value

D	Data matrix (X, Y).
vars	A list of the relevant variables.

Examples

```
genDataFromExamples(10,25,0.3)
```

LocRank *Localized ranking familty*

Description

Gradient-free Gradient Boosting family for the localized ranking loss function including its fast computation.

Usage

```
LocRank(K)
```

Arguments

K	Indicates that we are interesting in the top K instances and their correct ordering. Must be an integer between 1 and the number n of observations.

Details

The localized ranking loss combines the hard and the weak ranking loss, i.e., it penalizes misrankings at the top of the list (the best K instances according to the response value) and "misclassification" in the sense that instances belonging to the top of the list are ranked lower and vice versa. The localized ranking loss already returns a normalized loss that can take values between 0 and 1. LocRank returns a family object as in the package mboost.

Value

A Boosting family object

References

Werner, T., Gradient-Free Gradient Boosting, PhD Thesis, Carl von Ossietzky University Oldenburg, 2020, Equation (5.2.5)

T. Hothorn, P. Bühlmann, T. Kneib, M. Schmid, and B. Hofner. mboost: Model-Based Boosting, 2017

Examples

```
{y<-c(-3, 10.3,-8, 12, 14,-0.5, 29,-1.1,-5.7, 119)
 yhat<-c(0.02, 0.6, 0.1, 0.47, 0.82, 0.04, 0.77, 0.09, 0.01, 0.79)
 LocRank(4)@risk(y,yhat)}
{y<-c(-3, 10.3,-8, 12, 14,-0.5, 29,-1.1,-5.7, 119)
 yhat<-c(0.02, 0.6, 0.1, 0.47, 0.82, 0.04, 0.77, 0.09, 0.01, 0.79)
 LocRank(5)@risk(y,yhat)}
```

---

path.singboost *Coefficient paths for SingBoost*

Description

Runs SingBoost but saves the coefficients paths. If no coefficient path plot is needed, just use `singboost`.

Usage

```
path.singboost(
  D,
  M = 10,
  m_iter = 100,
  kap = 0.1,
  singfamily = Gaussian(),
  best = 1,
  LS = FALSE
)
```

Arguments

`D`	Data matrix. Has to be an $n \times (p+1)-$dimensional data frame in the format (X, Y). The $X-$part must not contain an intercept column containing only ones since this column will be added automatically.
`M`	An integer between 2 and `m_iter`. Indicates that in every $M-$th iteration, a singular iteration will be performed. Default is 10.
`m_iter`	Number of SingBoost iterations. Default is 100.
`kap`	Learning rate (step size). Must be a real number in $]0, 1]$. Default is 0.1 It is recommended to use a value smaller than 0.5.
`singfamily`	A Boosting family corresponding to the target loss function. See .`mboost` for families corresponding to standard loss functions. May also use the loss functions for ranking losses provided in this package. Default is `Gaussian()` for which SingBoost is just standard L_2-Boosting.
`best`	Needed in the case of localized ranking. The parameter `K` of the localized ranking loss will be computed by $best \cdot n$ (rounded to the next larger integer). Warning: If a parameter `K` is inserted into the `LocRank` family, it will be ignored when executing SingBoost.

`LS` If a `singfamily` object that is already provided by `mboost` is used, the respective Boosting algorithm will be performed in the singular iterations if `Ls` is set to `TRUE`. Default is `FALSE`.

Value

`Selected variables`
Names of the selected variables.

`Coefficients` The selected coefficients as an $(p+1)-$dimensional vector (i.e., including the zeroes).

`Freqs` Selection frequencies and a matrix for intercept and coefficient paths, respectively.

`Intercept path` The intercept path as an $m_{iter}-$dimensional vector.

`Coefficient path`
The coefficient paths as a $2 \cdot m_{iter} \times 2-$dimensional matrix.

---

random.CVind *Cross validation index generator*

Description

Simple auxiliary function for randomly generating the indices for training, validation and test data for cross validation.

Usage

```
random.CVind(n, ncmb, nval, CV)
```

Arguments

`n` Number of observations (rows).

`ncmb` Number of training samples for the SingBoost models in CMB. Must be an integer between 1 and n.

`nval` Number of validation samples in the CMB aggregation procedure. Must be an integer between 1 and $n - n_{cmb} - 1$.

`CV` Number of cross validation steps. Must be a positive integer.

Details

The data set consists of n observations. n_{cmb} of them are used for the CMB aggregation procedure. Note that within CMB itself, only a subset of these observations may be used for SingBoost training. The Stability Selection is based on the validation set consisting of n_{val} observations. The cross-validated loss of the final model is evaluated on the test data set with $n - n_{cmb} - n_{val}$ observations. Clearly, all data sets need to be disjoint.

Value

CVind — List of row indices for training, validation and test data for each cross validation loop.

Rank — *Hard ranking family*

Description

Gradient-free Gradient Boosting family for the hard ranking loss function including its fast computation.

Usage

```
Rank()
```

Details

The hard ranking loss is used to compare different orderings, usually the true ordering of instances of a data set according to their responses with the predicted counterparts. The usage of the pcaPP package avoids the cumbersome computation that would require

$$fracn2(n-1)$$

comparisons. Rank returns a family object as in the package mboost.

Value

A Boosting family object

References

Werner, T., Gradient-Free Gradient Boosting, PhD Thesis, Carl von Ossietzky University Oldenburg, 2020, Equations (5.2.2) and (5.2.3)

T. Hothorn, P. Bühlmann, T. Kneib, M. Schmid, and B. Hofner. mboost: Model-Based Boosting, 2017

Examples

```
{y<-c(-3, 10.3,-8, 12, 14,-0.5, 29,-1.1,-5.7, 119)
 yhat<-c(0.02, 0.6, 0.1, 0.47, 0.82, 0.04, 0.77, 0.09, 0.01, 0.79)
 Rank()@risk(y,yhat)}
{x<-1:6
z<-6:1
Rank()@risk(x,z)}
{x<-1:6
z<-1:6
Rank()@risk(x,z)}
```

RejStep *CMB validation step*

Description

Validation step to combine different SingBoost models.

Usage

```
RejStep(
  D,
  nsing,
  Bsing = 1,
  ind,
  sing = FALSE,
  singfam = Gaussian(),
  evalfam = Gaussian(),
  M = 10,
  m_iter = 100,
  kap = 0.1,
  LS = FALSE,
  best = 1
)
```

Arguments

D	Data matrix. Has to be an $n \times (p+1)-$dimensional data frame in the format (X, Y). The $X-$part must not contain an intercept column containing only ones since this column will be added automatically.
nsing	Number of observations (rows) used for the SingBoost submodels.
Bsing	Number of subsamples based on which the SingBoost models are validated. Default is 1. Not to confuse with parameter B for the Stability Selection.
ind	Vector with indices for dividing the data set into training and validation data.
sing	If sing=FALSE and the singfam family is a standard Boosting family that is contained in the package mboost, the CMB aggregation procedure is executed for the corresponding standard Boosting models.
singfam	A SingBoost family. The SingBoost models are trained based on the corresponding loss function. Default is Gaussian() (squared loss).
evalfam	A SingBoost family. The SingBoost models are validated according to the corresponding loss function. Default is Gaussian() (squared loss).
M	An integer between 2 and m_iter. Indicates that in every $M-$th iteration, a singular iteration will be performed. Default is 10.
m_iter	Number of SingBoost iterations. Default is 100.
kap	Learning rate (step size). Must be a real number in $]0, 1]$. Default is 0.1 It is recommended to use a value smaller than 0.5.

LS	If a singfamily object that is already provided by mboost is used, the respective Boosting algorithm will be performed in the singular iterations if Ls is set to TRUE. Default is FALSE.
best	Needed in the case of localized ranking. The parameter K of the localized ranking loss will be computed by $best \cdot n$ (rounded to the next larger integer). Warning: If a parameter K is inserted into the LocRank family, it will be ignored when executing SingBoost.

Details

Divides the data set into a training and a validation set. The SingBoost models are computed on the training set and evaluated on the validation set based on the loss function corresponding to the selected Boosting family.

Value

loss	Vector of validation losses.
occ	Selection frequencies for each Boosting model.

References

Werner, T., Gradient-Free Gradient Boosting, PhD Thesis, Carl von Ossietzky University Oldenburg, 2020

---

singboost *SingBoost Boosting method*

Description

SingBoost is a Boosting method that can deal with complicated loss functions that do not allow for a gradient. SingBoost is based on L2-Boosting in its current implementation.

Usage

```
singboost(
  D,
  M = 10,
  m_iter = 100,
  kap = 0.1,
  singfamily = Gaussian(),
  best = 1,
  LS = FALSE
)
```

Arguments

D	Data matrix. Has to be an $n \times (p+1)-$dimensional data frame in the format (X, Y). The $X-$part must not contain an intercept column containing only ones since this column will be added automatically.
M	An integer between 2 and m_iter. Indicates that in every $M-$th iteration, a singular iteration will be performed. Default is 10.
m_iter	Number of SingBoost iterations. Default is 100.
kap	Learning rate (step size). Must be a real number in $]0, 1]$. Default is 0.1 It is recommended to use a value smaller than 0.5.
singfamily	A Boosting family corresponding to the target loss function. See .mboost for families corresponding to standard loss functions. May also use the loss functions for ranking losses provided in this package. Default is Gaussian() for which SingBoost is just standard L_2-Boosting.
best	Needed in the case of localized ranking. The parameter K of the localized ranking loss will be computed by $best \cdot n$ (rounded to the next larger integer). Warning: If a parameter K is inserted into the LocRank family, it will be ignored when executing SingBoost.
LS	If a singfamily object that is already provided by mboost is used, the respective Boosting algorithm will be performed in the singular iterations if Ls is set to TRUE. Default is FALSE.

Details

Gradient Boosting algorithms require convexity and differentiability of the underlying loss function. SingBoost is a Boosting algorithm based on L_2-Boosting that allows for complicated loss functions that do not need to satisfy these requirements. In fact, SingBoost alternates between standard L_2-Boosting iterations and singular iterations where essentially an empirical gradient step is executed in the sense that the baselearner that performs best, evaluated in the complicated loss, is selected in the respective iteration. The implementation is based on glmboost from the package mboost and using the L_2-loss in the singular iterations returns exactly the same coefficients as L_2-Boosting.

Value

Selected variables	Names of the selected variables.
Coefficients	The selected coefficients as an $(p+1)-$dimensional vector (i.e., including the zeroes).
Freqs	Selection frequencies and a matrix for intercept and coefficient paths, respectively.
VarCoef	Vector of the non-zero coefficients.

References

Werner, T., Gradient-Free Gradient Boosting, PhD Thesis, Carl von Ossietzky University Oldenburg, 2020

P. Bühlmann and B. Yu. Boosting with the l2 loss: Regression and Classification. Journal of the American Statistical Association, 98(462):324–339, 2003

T. Hothorn, P. Bühlmann, T. Kneib, M. Schmid, and B. Hofner. mboost: Model-Based Boosting, 2017

Examples

```
{glmres<-glmboost(Sepal.Length~.,iris)
glmres
attributes(varimp(glmres))$self
attributes(varimp(glmres))$var
firis<-as.formula(Sepal.Length~.)
Xiris<-model.matrix(firis,iris)
Diris<-data.frame(Xiris[,-1],iris$Sepal.Length)
colnames(Diris)[6]<-"Y"
coef(glmboost(Xiris,iris$Sepal.Length))
singboost(Diris)
singboost(Diris,LS=TRUE)}
{glmres2<-glmboost(Sepal.Length~Petal.Length+Sepal.Width:Species,iris)
finter<-as.formula(Sepal.Length~Petal.Length+Sepal.Width:Species-1)
Xinter<-model.matrix(finter,iris)
Dinter<-data.frame(Xinter,iris$Sepal.Length)
singboost(Dinter)
coef(glmres2)}
{glmres3<-glmboost(Xiris,iris$Sepal.Length,control=boost_control(mstop=250,nu=0.05))
coef(glmres3)
attributes(varimp(glmres3))$self
singboost(Diris,m_iter=250,kap=0.05)
singboost(Diris,LS=TRUE,m_iter=250,kap=0.05)}
{glmquant<-glmboost(Sepal.Length~.,iris,family=QuantReg(tau=0.75))
coef(glmquant)
attributes(varimp(glmquant))$self
singboost(Diris,singfamily=QuantReg(tau=0.75),LS=TRUE)
singboost(Diris,singfamily=QuantReg(tau=0.75),LS=TRUE,M=2)}
{singboost(Diris,singfamily=Rank(),LS=TRUE)
singboost(Diris,singfamily=Rank(),LS=TRUE,M=2)}
```

`singboost.plot` *Plot function for the SingBoost coefficient paths*

Description

Plot function for the SingBoost coefficient paths

Usage

```
singboost.plot(mod, M, m_iter, subnames = FALSE)
```

Arguments

mod	singboost object.
M	An integer between 2 and m_iter. Indicates that in every M-th iteration, a singular iteration will be performed. Default is 10.
m_iter	Number of SingBoost iterations. Default is 100.
subnames	Use it only if the variable names are of the form "letter plus number". Better just ignore it.

Value

Nothing. Plots SingBoost coefficient paths

Examples

```
{glmres<-glmboost(Sepal.Length~.,iris)
glmres
attributes(varimp(glmres))$self
attributes(varimp(glmres))$var
firis<-as.formula(Sepal.Length~.)
Xiris<-model.matrix(firis,iris)
Diris<-data.frame(Xiris[,-1],iris$Sepal.Length)
plot(glmres)
singpath<-path.singboost(Diris)
singboost.plot(singpath,10,100,subnames=FALSE)}
```

WeakRank *Weak ranking family*

Description

Gradient-free Gradient Boosting family for the weak ranking loss function.

Usage

```
WeakRank(K)
```

Arguments

K	Indicates that we are only interesting in the top K instances. Must be an integer between 1 and the number n of observations.

Details

The weak ranking loss may be regarded as a classification loss. The parameter K defines the top of the list, consisting of the best K instances according to their response values. Then the weak ranking loss penalizes "misclassification" in the sense that instances belonging to the top of the list are ranked lower and vice versa. WeakRank returns a family object as in the package mboost.

Value

A Boosting family object

References

Werner, T., Gradient-Free Gradient Boosting, PhD Thesis, Carl von Ossietzky University Oldenburg, 2020, Remark (5.2.1)

T. Hothorn, P. Bühlmann, T. Kneib, M. Schmid, and B. Hofner. mboost: Model-Based Boosting, 2017

Examples

```
{y<-c(-3, 10.3,-8, 12, 14,-0.5, 29,-1.1,-5.7, 119)
yhat<-c(0.02, 0.6, 0.1, 0.47, 0.82, 0.04, 0.77, 0.09, 0.01, 0.79)
WeakRank(4)@risk(y,yhat)}
{y<-c(-3, 10.3,-8, 12, 14,-0.5, 29,-1.1,-5.7, 119)
yhat<-c(0.02, 0.6, 0.1, 0.47, 0.82, 0.04, 0.77, 0.09, 0.01, 0.79)
WeakRank(5)@risk(y,yhat)}
```

WeakRankNorm *Weak ranking family (normalized)*

Description

Gradient-free Gradient Boosting family for the normalized weak ranking loss function.

Usage

```
WeakRankNorm(K)
```

Arguments

K	Indicates that we are only interesting in the top K instances. Must be an integer between 1 and the number n of observations.

Details

A more intuitive loss function than the weak ranking loss thanks to its normalization to a maximum value of 1. For example, if a number c of the top K instances has not been ranked at the top of the list, the normalized weak ranking loss is C/K. `WeakRankNorm` returns a family object as in the package `mboost`.

Value

A Boosting family object

References

Werner, T., Gradient-Free Gradient Boosting, PhD Thesis, Carl von Ossietzky University Oldenburg, 2020, Remark (5.2.4)

T. Hothorn, P. Bühlmann, T. Kneib, M. Schmid, and B. Hofner. mboost: Model-Based Boosting, 2017

Index