Package 'bootPLS'

September 24, 2024

Version 1.0.1

Date 2024-09-23

Depends R (>= 3.5.0)

Imports boot, doParallel, foreach, plsRglm, pls, spls, bipartite, mvtnorm

Suggests knitr, markdown, plsdof, prettydoc, rmarkdown

Title Bootstrap Hyperparameter Selection for PLS Models and Extensions

Author Frederic Bertrand [cre, aut] (<`https://orcid.org/0000-0002-0837-8281`>),
Jeremy Magnanensi [aut],
Myriam Maumy-Bertrand [aut] (<`https://orcid.org/0000-0002-4615-1512`>)

Maintainer Frederic Bertrand <`frederic.bertrand@utt.fr`>

Description Several implementations of non-parametric stable bootstrap-based techniques to determine the numbers of components for Partial Least Squares linear or generalized linear regression models as well as and sparse Partial Least Squares linear or generalized linear regression models. The package collects techniques that were published in a book chapter (Magnanensi et al. 2016, 'The Multiple Facets of Partial Least Squares and Related Methods', <doi:10.1007/978-3-319-40643-5_18>) and two articles (Magnanensi et al. 2017, 'Statistics and Computing', <doi:10.1007/s11222-016-9651-4>) and (Magnanensi et al. 2021, 'Frontiers in Applied Mathematics and Statistics', <doi:10.3389/fams.2021.693126>).

License GPL-3

Encoding UTF-8

Classification/MSC 62N01, 62N02, 62N03, 62N99

LazyData true

VignetteBuilder knitr

RoxygenNote 7.3.2

URL `https://fbertran.github.io/bootPLS/`,
`https://github.com/fbertran/bootPLS/`

BugReports `https://github.com/fbertran/bootPLS/issues/`

NeedsCompilation no

Repository CRAN

Date/Publication 2024-09-24 11:10:08 UTC

Contents

`coefs.plsR.adapt.ncomp` *Bootstrap (Y,X) for the coefficients with number of components updated for each resampling.*

Description

Bootstrap (Y,X) for the coefficients with number of components updated for each resampling.

Usage

```
coefs.plsR.adapt.ncomp(
  dataset,
  i,
  R = 1000,
  ncpus = 1,
  parallel = "no",
  verbose = FALSE
)
```

Arguments

`dataset`	Dataset to use.
`i`	Vector of resampling.
`R`	Number of resamplings to find the number of components.

`ncpus` integer: number of processes to be used in parallel operation: typically one would chose this to the number of available CPUs.

`parallel` The type of parallel operation to be used (if any). If missing, the default is taken from the option "boot.parallel" (and if that is not set, "no").

`verbose` Suppress information messages.

Value

Numeric vector: first value is the number of components, the remaining values are the coefficients the variables computed for that number of components.

Author(s)

Jérémy Magnanensi, Frédéric Bertrand
`<frederic.bertrand@utt.fr>`
`https://fbertran.github.io/homepage/`

References

A new bootstrap-based stopping criterion in PLS component construction, J. Magnanensi, M. Maumy-Bertrand, N. Meyer and F. Bertrand (2016), in The Multiple Facets of Partial Least Squares and Related Methods, doi:10.1007/9783319406435_18

A new universal resample-stable bootstrap-based stopping criterion for PLS component construction, J. Magnanensi, F. Bertrand, M. Maumy-Bertrand and N. Meyer, (2017), Statistics and Computing, 27, 757–774. doi:10.1007/s1122201696514

New developments in Sparse PLS regression, J. Magnanensi, M. Maumy-Bertrand, N. Meyer and F. Bertrand, (2021), Frontiers in Applied Mathematics and Statistics, doi:10.3389/fams.2021.693126
.

Examples

```
set.seed(314)
ncol=5
xran=matrix(rnorm(30*ncol),30,ncol)
coefs.plsR.adapt.ncomp(xran,sample(1:30))

coefs.plsR.adapt.ncomp(xran,sample(1:30),ncpus=2,parallel="multicore")
```

`coefs.plsR.CSim` *Bootstrap (Y,T) functions for PLSR*

Description

Bootstrap (Y,T) functions for PLSR

Usage

```
coefs.plsR.CSim(dataset, i)
```

Arguments

`dataset`	Dataset with tt
`i`	Index for resampling

Value

Coefficient of the last variable in the linear regression `lm(dataset[i,1] ~ dataset[,-1] - 1)` computed using bootstrap resampling.

Author(s)

Jérémy Magnanensi, Frédéric Bertrand
`<frederic.bertrand@utt.fr>`
`https://fbertran.github.io/homepage/`

References

A new bootstrap-based stopping criterion in PLS component construction, J. Magnanensi, M. Maumy-Bertrand, N. Meyer and F. Bertrand (2016), in The Multiple Facets of Partial Least Squares and Related Methods, doi:10.1007/9783319406435_18

A new universal resample-stable bootstrap-based stopping criterion for PLS component construction, J. Magnanensi, F. Bertrand, M. Maumy-Bertrand and N. Meyer, (2017), Statistics and Computing, 27, 757–774. doi:10.1007/s1122201696514

New developments in Sparse PLS regression, J. Magnanensi, M. Maumy-Bertrand, N. Meyer and F. Bertrand, (2021), Frontiers in Applied Mathematics and Statistics, doi:10.3389/fams.2021.693126
.

Examples

```
set.seed(314)
xran=matrix(rnorm(150),30,5)
coefs.plsR.CSim(xran,sample(1:30))
```

`coefs.plsRglm.CSim` *Bootstrap (Y,T) function for PLSGLR*

Description

A function passed to `boot` to perform bootstrap.

Usage

```
coefs.plsRglm.CSim(
  dataRepYtt,
  ind,
  nt,
  modele,
  family = NULL,
  maxcoefvalues,
  ifbootfail
)
```

Arguments

`dataRepYtt`	Dataset with tt components to resample
`ind`	indices for resampling
`nt`	number of components to use
`modele`	type of modele to use, see plsRglm. Not used, please specify the family instead.
`family`	glm family to use, see plsRglm
`maxcoefvalues`	maximum values allowed for the estimates of the coefficients to discard those coming from singular bootstrap samples
`ifbootfail`	value to return if the estimation fails on a bootstrap sample

Value

estimates on a bootstrap sample or `ifbootfail` value if the bootstrap computation fails.

Numeric vector of the components computed using a bootstrap resampling.

Author(s)

Jérémy Magnanensi, Frédéric Bertrand
`<frederic.bertrand@utt.fr>`
`https://fbertran.github.io/homepage/`

References

A new bootstrap-based stopping criterion in PLS component construction, J. Magnanensi, M. Maumy-Bertrand, N. Meyer and F. Bertrand (2016), in The Multiple Facets of Partial Least Squares and Related Methods, doi:10.1007/9783319406435_18

A new universal resample-stable bootstrap-based stopping criterion for PLS component construction, J. Magnanensi, F. Bertrand, M. Maumy-Bertrand and N. Meyer, (2017), Statistics and Computing, 27, 757–774. doi:10.1007/s1122201696514

New developments in Sparse PLS regression, J. Magnanensi, M. Maumy-Bertrand, N. Meyer and F. Bertrand, (2021), Frontiers in Applied Mathematics and Statistics, doi:10.3389/fams.2021.693126
.

Examples

```
set.seed(314)
library(plsRglm)
data(aze_compl, package="plsRglm")
Xaze_compl<-aze_compl[,2:34]
yaze_compl<-aze_compl$y
dataset <- cbind(y=yaze_compl,Xaze_compl)
modplsglm <- plsRglm::plsRglm(y~.,data=dataset,4,modele="pls-glm-family",family=binomial)
dataRepYtt <- cbind(y = modplsglm$RepY, modplsglm$tt)
coefs.plsRglm.CSim(dataRepYtt, sample(1:nrow(dataRepYtt)), 4,
family = binomial, maxcoefvalues=10, ifbootfail=0)
```

coefs.sgpls.CSim *Bootstrap (Y,T) function for plsRglm*

Description

A function passed to `boot` to perform bootstrap.

Usage

```
coefs.sgpls.CSim(
  dataRepYtt,
  ind,
  nt,
  modele,
  family = binomial,
  maxcoefvalues,
  ifbootfail
)
```

Arguments

`dataRepYtt`	Dataset with tt components to resample
`ind`	indices for resampling
`nt`	number of components to use
`modele`	type of modele to use, see plsRglm. Not used, please specify the family instead.
`family`	glm family to use, see plsRglm
`maxcoefvalues`	maximum values allowed for the estimates of the coefficients to discard those coming from singular bootstrap samples
`ifbootfail`	value to return if the estimation fails on a bootstrap sample

Value

Numeric vector of the components computed using a bootstrap resampling or `ifbootfail` value if the bootstrap computation fails.

Author(s)

Jérémy Magnanensi, Frédéric Bertrand
`<frederic.bertrand@utt.fr>`
`https://fbertran.github.io/homepage/`

References

A new bootstrap-based stopping criterion in PLS component construction, J. Magnanensi, M. Maumy-Bertrand, N. Meyer and F. Bertrand (2016), in The Multiple Facets of Partial Least Squares and Related Methods, doi:10.1007/9783319406435_18

A new universal resample-stable bootstrap-based stopping criterion for PLS component construction, J. Magnanensi, F. Bertrand, M. Maumy-Bertrand and N. Meyer, (2017), Statistics and Computing, 27, 757–774. doi:10.1007/s1122201696514

New developments in Sparse PLS regression, J. Magnanensi, M. Maumy-Bertrand, N. Meyer and F. Bertrand, (2021), Frontiers in Applied Mathematics and Statistics, doi:10.3389/fams.2021.693126
.

Examples

```
set.seed(4619)
xran=cbind(rbinom(30,1,.2),matrix(rnorm(150),30,5))
coefs.sgpls.CSim(xran, ind=sample(1:nrow(xran)),
maxcoefvalues=1e5, ifbootfail=rep(NA,3))
```

datasim *Simulated dataset for gamma family based PLSR*

Description

This dataset provides a simulated dataset for gamma family based PLSR that was created with the simul_data_UniYX_gamma function.

Format

A data frame with 200 observations on the following 8 variables.

Ygamma a numeric vector

X1 a numeric vector

X2 a numeric vector

X3 a numeric vector

X4 a numeric vector

X5 a numeric vector

X6 a numeric vector

X7 a numeric vector

X8 a numeric vector

Examples

```
data(datasim)
X_datasim_train <- datasim[1:140,2:8]
y_datasim_train <- datasim[1:140,1]
X_datasim_test <- datasim[141:200,2:8]
y_datasim_test <- datasim[141:200,1]
rm(X_datasim_train,y_datasim_train,X_datasim_test,y_datasim_test)
```

nbcomp.bootplsR *Non-parametric (Y,T) Bootstrap for selecting the number of components in PLSR models*

Description

Provides a wrapper for the bootstrap function boot from the boot R package.
Implements non-parametric bootstraps for PLS Regression models by (Y,T) resampling to select the number of components.

Usage

```
nbcomp.bootplsR(
  Y,
  X,
  R = 500,
  sim = "ordinary",
  ncpus = 1,
  parallel = "no",
  typeBCa = TRUE,
  verbose = TRUE
)
```

Arguments

`Y`	Vector of response.
`X`	Matrix of predictors.
`R`	The number of bootstrap replicates. Usually this will be a single positive integer. For importance resampling, some resamples may use one set of weights and others use a different set of weights. In this case `R` would be a vector of integers where each component gives the number of resamples from each of the rows of weights.
`sim`	A character string indicating the type of simulation required. Possible values are `"ordinary"` (the default), `"balanced"`, `"permutation"`, or `"antithetic"`.
`ncpus`	integer: number of processes to be used in parallel operation: typically one would chose this to the number of available CPUs.
`parallel`	The type of parallel operation to be used (if any). If missing, the default is taken from the option "boot.parallel" (and if that is not set, "no").
`typeBCa`	Compute BCa type intervals ?
`verbose`	Display info during the run of algorithm?

Details

More details on bootstrap techniques are available in the help of the `boot` function.

Value

A numeric, the number of components selected by the bootstrap.

Author(s)

Jérémy Magnanensi, Frédéric Bertrand
`<frederic.bertrand@utt.fr>`
`https://fbertran.github.io/homepage/`

References

A new bootstrap-based stopping criterion in PLS component construction, J. Magnanensi, M. Maumy-Bertrand, N. Meyer and F. Bertrand (2016), in The Multiple Facets of Partial Least Squares and Related Methods, doi:10.1007/9783319406435_18

A new universal resample-stable bootstrap-based stopping criterion for PLS component construction, J. Magnanensi, F. Bertrand, M. Maumy-Bertrand and N. Meyer, (2017), Statistics and Computing, 27, 757–774. doi:10.1007/s1122201696514

New developments in Sparse PLS regression, J. Magnanensi, M. Maumy-Bertrand, N. Meyer and F. Bertrand, (2021), Frontiers in Applied Mathematics and Statistics, doi:10.3389/fams.2021.693126 .

Examples

```
data(pine, package="plsRglm")
Xpine<-pine[,1:10]
ypine<-log(pine[,11])
res <- nbcomp.bootplsR(ypine, Xpine)
nbcomp.bootplsR(ypine, Xpine, typeBCa=FALSE)

nbcomp.bootplsR(ypine, Xpine, typeBCa=FALSE, verbose=FALSE)
try(nbcomp.bootplsR(ypine, Xpine, sim="permutation"))
nbcomp.bootplsR(ypine, Xpine, sim="permutation", typeBCa=FALSE)
```

---

nbcomp.bootplsRglm *Non-parametric (Y,T) Bootstrap for selecting the number of components in PLS GLR models*

Description

Provides a wrapper for the bootstrap function boot from the boot R package.
Implements non-parametric bootstraps for PLS Generalized Linear Regression models by (Y,T) resampling to select the number of components.

Usage

```
nbcomp.bootplsRglm(
  object,
  typeboot = "boot_comp",
  R = 250,
  statistic = coefs.plsRglm.CSim,
  sim = "ordinary",
  stype = "i",
  stabvalue = 1e+06,
    ...
)
```

Arguments

`object` An object of class `plsRmodel` to bootstrap

`typeboot` The type of bootstrap. (`typeboot="boot_comp"`) for (Y,T) bootstrap to select components. Defaults to (`typeboot="boot_comp"`).

`R` The number of bootstrap replicates. Usually this will be a single positive integer. For importance resampling, some resamples may use one set of weights and others use a different set of weights. In this case `R` would be a vector of integers where each component gives the number of resamples from each of the rows of weights.

`statistic` A function which when applied to data returns a vector containing the statistic(s) of interest. `statistic` must take at least two arguments. The first argument passed will always be the original data. The second will be a vector of indices, frequencies or weights which define the bootstrap sample. Further, if predictions are required, then a third argument is required which would be a vector of the random indices used to generate the bootstrap predictions. Any further arguments can be passed to statistic through the `...` argument.

`sim` A character string indicating the type of simulation required. Possible values are `"ordinary"` (the default), `"balanced"`, `"permutation"`, or `"antithetic"`.

`stype` A character string indicating what the second argument of `statistic` represents. Possible values of stype are `"i"` (indices - the default), `"f"` (frequencies), or `"w"` (weights).

`stabvalue` A value to hard threshold bootstrap estimates computed from atypical resamplings. Especially useful for Generalized Linear Models.

`...` Other named arguments for `statistic` which are passed unchanged each time it is called. Any such arguments to `statistic` should follow the arguments which `statistic` is required to have for the simulation. Beware of partial matching to arguments of `boot` listed above.

Details

More details on bootstrap techniques are available in the help of the `boot` function.

Value

An object of class `"boot"`. See the Value part of the help of the function `boot`.

Author(s)

Jérémy Magnanensi, Frédéric Bertrand
`<frederic.bertrand@utt.fr>`
`https://fbertran.github.io/homepage/`

References

A new bootstrap-based stopping criterion in PLS component construction, J. Magnanensi, M. Maumy-Bertrand, N. Meyer and F. Bertrand (2016), in The Multiple Facets of Partial Least Squares and Related Methods, doi:10.1007/9783319406435_18

A new universal resample-stable bootstrap-based stopping criterion for PLS component construction, J. Magnanensi, F. Bertrand, M. Maumy-Bertrand and N. Meyer, (2017), Statistics and Computing, 27, 757–774. doi:10.1007/s1122201696514

New developments in Sparse PLS regression, J. Magnanensi, M. Maumy-Bertrand, N. Meyer and F. Bertrand, (2021), Frontiers in Applied Mathematics and Statistics, doi:10.3389/fams.2021.693126
.

Examples

```
set.seed(314)
library(plsRglm)
data(aze_compl, package="plsRglm")
Xaze_compl<-aze_compl[,2:34]
yaze_compl<-aze_compl$y
dataset <- cbind(y=yaze_compl,Xaze_compl)
modplsglm <- plsRglm::plsRglm(y~.,data=dataset,10,modele="pls-glm-family", family = binomial)

comp_aze_compl.bootYT <- nbcomp.bootplsRglm(modplsglm, R=250)
boxplots.bootpls(comp_aze_compl.bootYT)
confints.bootpls(comp_aze_compl.bootYT)
plots.confints.bootpls(confints.bootpls(comp_aze_compl.bootYT),typeIC = "BCa")

comp_aze_compl.permYT <- nbcomp.bootplsRglm(modplsglm, R=250, sim="permutation")
boxplots.bootpls(comp_aze_compl.permYT)
confints.bootpls(comp_aze_compl.permYT, typeBCa=FALSE)
plots.confints.bootpls(confints.bootpls(comp_aze_compl.permYT, typeBCa=FALSE))
```

nbcomp.bootsgpls *Number of components for SGPLS using (Y,T) bootstrap*

Description

Number of components for SGPLS using (Y,T) bootstrap

Usage

```
nbcomp.bootsgpls(
  x,
  y,
  fold = 10,
  eta,
  R,
  scale.x = TRUE,
  maxnt = 10,
  plot.it = TRUE,
  br = TRUE,
  ftype = "iden",
  typeBCa = TRUE,
  stabvalue = 1e+06,
  verbose = TRUE
)
```

Arguments

`x` Matrix of predictors.

`y` Vector or matrix of responses.

`fold` Number of fold for cross-validation

`eta` Thresholding parameter. eta should be between 0 and 1.

`R` Number of resamplings.

`scale.x` Scale predictors by dividing each predictor variable by its sample standard deviation?

`maxnt` Maximum number of components allowed in a spls model.

`plot.it` Plot the results.

`br` Apply Firth's bias reduction procedure?

`ftype` Type of Firth's bias reduction procedure. Alternatives are "iden" (the approximated version) or "hat" (the original version). Default is "iden".

`typeBCa` Include computation for BCa type interval.

`stabvalue` A value to hard threshold bootstrap estimates computed from atypical resamplings.

`verbose` Additionnal information on the algorithm.

Value

List of four: error matrix, eta optimal, K optimal and the matrix of results.

Author(s)

Jérémy Magnanensi, Frédéric Bertrand
`<frederic.bertrand@utt.fr>`
`https://fbertran.github.io/homepage/`

References

A new bootstrap-based stopping criterion in PLS component construction, J. Magnanensi, M. Maumy-Bertrand, N. Meyer and F. Bertrand (2016), in The Multiple Facets of Partial Least Squares and Related Methods, doi:10.1007/9783319406435_18

A new universal resample-stable bootstrap-based stopping criterion for PLS component construction, J. Magnanensi, F. Bertrand, M. Maumy-Bertrand and N. Meyer, (2017), Statistics and Computing, 27, 757–774. doi:10.1007/s1122201696514

New developments in Sparse PLS regression, J. Magnanensi, M. Maumy-Bertrand, N. Meyer and F. Bertrand, (2021), Frontiers in Applied Mathematics and Statistics, doi:10.3389/fams.2021.693126 .

Examples

```
set.seed(4619)
data(prostate, package="spls")
nbcomp.bootsgpls((prostate$x)[,1:30], prostate$y, R=250, eta=0.2, maxnt=1, typeBCa = FALSE)

set.seed(4619)
data(prostate, package="spls")
nbcomp.bootsgpls(prostate$x, prostate$y, R=250, eta=c(0.2,0.6), typeBCa = FALSE)
```

`nbcomp.bootsgpls.para` *Number of components for SGPLS using (Y,T) bootstrap (parallel version)*

Description

Number of components for SGPLS using (Y,T) bootstrap (parallel version)

Usage

```
nbcomp.bootsgpls.para(
  x,
  y,
  fold = 10,
  eta,
  R,
  scale.x = TRUE,
  maxnt = 10,
  br = TRUE,
  ftype = "iden",
  ncpus = 1,
  plot.it = TRUE,
  typeBCa = TRUE,
  stabvalue = 1e+06,
  verbose = TRUE
)
```

Arguments

`x` Matrix of predictors.

`y` Vector or matrix of responses.

`fold` Number of fold for cross-validation.

`eta` Thresholding parameter. eta should be between 0 and 1.

`R` Number of resamplings.

`scale.x` Scale predictors by dividing each predictor variable by its sample standard deviation?

`maxnt` Maximum number of components allowed in a spls model.

`br` Apply Firth's bias reduction procedure?

`ftype` Type of Firth's bias reduction procedure. Alternatives are "iden" (the approximated version) or "hat" (the original version). Default is "iden".

`ncpus` Number of cpus for parallel computing.

`plot.it` Plot the results.

`typeBCa` Include computation for BCa type interval.

`stabvalue` A value to hard threshold bootstrap estimates computed from atypical resamplings.

`verbose` Additionnal information on the algorithm.

Value

List of four: error matrix, eta optimal, K optimal and the matrix of results.

Author(s)

Jérémy Magnanensi, Frédéric Bertrand
`<frederic.bertrand@utt.fr>`
`https://fbertran.github.io/homepage/`

References

A new bootstrap-based stopping criterion in PLS component construction, J. Magnanensi, M. Maumy-Bertrand, N. Meyer and F. Bertrand (2016), in The Multiple Facets of Partial Least Squares and Related Methods, doi:10.1007/9783319406435_18

A new universal resample-stable bootstrap-based stopping criterion for PLS component construction, J. Magnanensi, F. Bertrand, M. Maumy-Bertrand and N. Meyer, (2017), Statistics and Computing, 27, 757–774. doi:10.1007/s1122201696514

New developments in Sparse PLS regression, J. Magnanensi, M. Maumy-Bertrand, N. Meyer and F. Bertrand, (2021), Frontiers in Applied Mathematics and Statistics, doi:10.3389/fams.2021.693126
.

Examples

```
set.seed(4619)
data(prostate, package="spls")
nbcomp.bootsgpls.para((prostate$x)[,1:30], prostate$y, R=250, eta=0.2, maxnt=1, typeBCa = FALSE)

set.seed(4619)
data(prostate, package="spls")
nbcomp.bootsgpls.para(prostate$x, prostate$y, R=250, eta=c(0.2,0.6), typeBCa = FALSE)
```

nbcomp.bootspls *Title*

Description

Title

Usage

```
nbcomp.bootspls(
  x,
  y,
  fold = 10,
  eta,
  R = 500,
  maxnt = 10,
  kappa = 0.5,
  select = "pls2",
  fit = "simpls",
  scale.x = TRUE,
  scale.y = FALSE,
  plot.it = TRUE,
  typeBCa = TRUE,
  verbose = TRUE
)
```

Arguments

`x` Matrix of predictors.

`y` Vector or matrix of responses.

`fold` Number of fold for cross-validation

`eta` Thresholding parameter. eta should be between 0 and 1.

`R` Number of resamplings.

`maxnt` Maximum number of components allowed in a spls model.

`kappa` Parameter to control the effect of the concavity of the objective function and the closeness of original and surrogate direction vectors. kappa is relevant only when responses are multivariate. kappa should be between 0 and 0.5. Default is 0.5.

`select` PLS algorithm for variable selection. Alternatives are "pls2" or "simpls". Default is "pls2".

`fit` PLS algorithm for model fitting. Alternatives are "kernelpls", "widekernelpls", "simpls", or "oscorespls". Default is "simpls".

`scale.x` Scale predictors by dividing each predictor variable by its sample standard deviation?

`scale.y` Scale responses by dividing each response variable by its sample standard deviation?

`plot.it` Plot the results.

`typeBCa` Include computation for BCa type interval.

`verbose` Displays information on the algorithm.

Value

list of 3: mspemat matrix of results, eta.opt numeric value, K.opt numeric value)

Author(s)

Jérémy Magnanensi, Frédéric Bertrand
`<frederic.bertrand@utt.fr>`
`https://fbertran.github.io/homepage/`

References

A new bootstrap-based stopping criterion in PLS component construction, J. Magnanensi, M. Maumy-Bertrand, N. Meyer and F. Bertrand (2016), in The Multiple Facets of Partial Least Squares and Related Methods, doi:10.1007/9783319406435_18

A new universal resample-stable bootstrap-based stopping criterion for PLS component construction, J. Magnanensi, F. Bertrand, M. Maumy-Bertrand and N. Meyer, (2017), Statistics and Computing, 27, 757–774. doi:10.1007/s1122201696514

New developments in Sparse PLS regression, J. Magnanensi, M. Maumy-Bertrand, N. Meyer and F. Bertrand, (2021), Frontiers in Applied Mathematics and Statistics, doi:10.3389/fams.2021.693126
.

Examples

```
set.seed(314)
data(pine, package = "plsRglm")
Xpine<-pine[,1:10]
ypine<-log(pine[,11])
nbcomp.bootspls(x=Xpine,y=ypine,eta=.2, maxnt=1)
```

```
set.seed(314)
data(pine, package = "plsRglm")
Xpine<-pine[,1:10]
ypine<-log(pine[,11])
nbcomp.bootspls.para(x=Xpine,y=ypine,eta=c(.2,.6))
```

nbcomp.bootspls.para *Title*

Description

Title

Usage

```
nbcomp.bootspls.para(
  x,
  y,
  fold = 10,
  eta,
  R = 500,
  maxnt = 10,
  kappa = 0.5,
  select = "pls2",
  fit = "simpls",
  scale.x = TRUE,
  scale.y = FALSE,
  plot.it = TRUE,
  typeBCa = TRUE,
  ncpus = 1,
  verbose = TRUE
)
```

Arguments

`x`	Matrix of predictors.
`y`	Vector or matrix of responses.
`fold`	Number of fold for cross-validation
`eta`	Thresholding parameter. eta should be between 0 and 1.
`R`	Number of resamplings.
`maxnt`	Maximum number of components allowed in a spls model.
`kappa`	Parameter to control the effect of the concavity of the objective function and the closeness of original and surrogate direction vectors. kappa is relevant only when responses are multivariate. kappa should be between 0 and 0.5. Default is 0.5.

`select` PLS algorithm for variable selection. Alternatives are "pls2" or "simpls". Default is "pls2".

`fit` PLS algorithm for model fitting. Alternatives are "kernelpls", "widekernelpls", "simpls", or "oscorespls". Default is "simpls".

`scale.x` Scale predictors by dividing each predictor variable by its sample standard deviation?

`scale.y` Scale responses by dividing each response variable by its sample standard deviation?

`plot.it` Plot the results.

`typeBCa` Include computation for BCa type interval.

`ncpus` Number of cpus for parallel computing.

`verbose` Displays information on the algorithm.

Value

list of 3: mspemat matrix of results, eta.opt numeric value, K.opt numeric value)

Author(s)

Jérémy Magnanensi, Frédéric Bertrand
`<frederic.bertrand@utt.fr>`
`https://fbertran.github.io/homepage/`

References

A new bootstrap-based stopping criterion in PLS component construction, J. Magnanensi, M. Maumy-Bertrand, N. Meyer and F. Bertrand (2016), in The Multiple Facets of Partial Least Squares and Related Methods, doi:10.1007/9783319406435_18

A new universal resample-stable bootstrap-based stopping criterion for PLS component construction, J. Magnanensi, F. Bertrand, M. Maumy-Bertrand and N. Meyer, (2017), Statistics and Computing, 27, 757–774. doi:10.1007/s1122201696514

New developments in Sparse PLS regression, J. Magnanensi, M. Maumy-Bertrand, N. Meyer and F. Bertrand, (2021), Frontiers in Applied Mathematics and Statistics, doi:10.3389/fams.2021.693126
.

Examples

```
set.seed(314)
data(pine, package = "plsRglm")
Xpine<-pine[,1:10]
ypine<-log(pine[,11])
nbcomp.bootspls.para(x=Xpine,y=ypine,eta=.2, maxnt=1)

set.seed(314)
data(pine, package = "plsRglm")
```

```
Xpine<-pine[,1:10]
ypine<-log(pine[,11])
nbcomp.bootspls.para(x=Xpine,y=ypine,eta=c(.2,.6))
```

`permcoefs.plsR.CSim` *Permutation bootstrap (Y,T) function for PLSR*

Description

Permutation bootstrap (Y,T) function for PLSR

Usage

```
permcoefs.plsR.CSim(dataset, i)
```

Arguments

`dataset`	Dataset with tt
`i`	Index for resampling

Value

Coefficient of the last variable in the linear regression `lm(dataset[i,1] ~ dataset[,-1] - 1)` computed using permutation resampling.

Author(s)

Jérémy Magnanensi, Frédéric Bertrand
`<frederic.bertrand@utt.fr>`
`https://fbertran.github.io/homepage/`

References

A new bootstrap-based stopping criterion in PLS component construction, J. Magnanensi, M. Maumy-Bertrand, N. Meyer and F. Bertrand (2016), in The Multiple Facets of Partial Least Squares and Related Methods, doi:10.1007/9783319406435_18

A new universal resample-stable bootstrap-based stopping criterion for PLS component construction, J. Magnanensi, F. Bertrand, M. Maumy-Bertrand and N. Meyer, (2017), Statistics and Computing, 27, 757–774. doi:10.1007/s1122201696514

New developments in Sparse PLS regression, J. Magnanensi, M. Maumy-Bertrand, N. Meyer and F. Bertrand, (2021), Frontiers in Applied Mathematics and Statistics, doi:10.3389/fams.2021.693126
.

Examples

```
set.seed(314)
xran=matrix(rnorm(150),30,5)
permcoefs.plsR.CSim(xran,sample(1:30))
```

---

permcoefs.plsRglm.CSim *Permutation bootstrap (Y,T) function for PLSGLR*

Description

A function passed to boot to perform bootstrap.

Usage

```
permcoefs.plsRglm.CSim(
  dataRepYtt,
  ind,
  nt,
  modele,
  family = NULL,
  maxcoefvalues,
  ifbootfail
)
```

Arguments

dataRepYtt	Dataset with tt components to resample
ind	indices for resampling
nt	number of components to use
modele	type of modele to use, see plsRglm. Not used, please specify the family instead.
family	glm family to use, see plsRglm
maxcoefvalues	maximum values allowed for the estimates of the coefficients to discard those coming from singular bootstrap samples
ifbootfail	value to return if the estimation fails on a bootstrap sample

Value

estimates on a bootstrap sample or ifbootfail value if the bootstrap computation fails.

Numeric vector of the components computed using a permutation resampling.

Author(s)

Jérémy Magnanensi, Frédéric Bertrand
<frederic.bertrand@utt.fr>
https://fbertran.github.io/homepage/

References

A new bootstrap-based stopping criterion in PLS component construction, J. Magnanensi, M. Maumy-Bertrand, N. Meyer and F. Bertrand (2016), in The Multiple Facets of Partial Least Squares and Related Methods, doi:10.1007/9783319406435_18

A new universal resample-stable bootstrap-based stopping criterion for PLS component construction, J. Magnanensi, F. Bertrand, M. Maumy-Bertrand and N. Meyer, (2017), Statistics and Computing, 27, 757–774. doi:10.1007/s1122201696514

New developments in Sparse PLS regression, J. Magnanensi, M. Maumy-Bertrand, N. Meyer and F. Bertrand, (2021), Frontiers in Applied Mathematics and Statistics, doi:10.3389/fams.2021.693126
.

Examples

```
set.seed(314)
library(plsRglm)
data(aze_compl, package="plsRglm")
Xaze_compl<-aze_compl[,2:34]
yaze_compl<-aze_compl$y
dataset <- cbind(y=yaze_compl,Xaze_compl)
modplsglm <- plsRglm::plsRglm(y~.,data=dataset,4,modele="pls-glm-logistic")
dataRepYtt <- cbind(y = modplsglm$RepY, modplsglm$tt)
permcoefs.plsRglm.CSim(dataRepYtt, sample(1:nrow(dataRepYtt)), 4,
family = binomial, maxcoefvalues=10, ifbootfail=0)
```

---

permcoefs.sgpls.CSim *Permutation Bootstrap (Y,T) function for plsRglm*

Description

Permutation Bootstrap (Y,T) function for plsRglm

Usage

```
permcoefs.sgpls.CSim(
  dataRepYtt,
  ind,
  nt,
  modele,
  family = binomial,
  maxcoefvalues,
  ifbootfail
)
```

Arguments

`dataRepYtt` Dataset with tt components to resample

`ind` indices for resampling

`nt` number of components to use

`modele` type of modele to use, see plsRglm. Not used, please specify the family instead.

`family` glm family to use, see plsRglm

`maxcoefvalues` maximum values allowed for the estimates of the coefficients to discard those coming from singular bootstrap samples

`ifbootfail` value to return if the estimation fails on a bootstrap sample

Value

Numeric vector of the components computed using a bootstrap resampling or `ifbootfail` value if the bootstrap computation fails.

Author(s)

Jérémy Magnanensi, Frédéric Bertrand
`<frederic.bertrand@utt.fr>`
`https://fbertran.github.io/homepage/`

References

A new bootstrap-based stopping criterion in PLS component construction, J. Magnanensi, M. Maumy-Bertrand, N. Meyer and F. Bertrand (2016), in The Multiple Facets of Partial Least Squares and Related Methods, doi:10.1007/9783319406435_18

A new universal resample-stable bootstrap-based stopping criterion for PLS component construction, J. Magnanensi, F. Bertrand, M. Maumy-Bertrand and N. Meyer, (2017), Statistics and Computing, 27, 757–774. doi:10.1007/s1122201696514

New developments in Sparse PLS regression, J. Magnanensi, M. Maumy-Bertrand, N. Meyer and F. Bertrand, (2021), Frontiers in Applied Mathematics and Statistics, doi:10.3389/fams.2021.693126
.

Examples

```
set.seed(4619)
xran=cbind(rbinom(30,1,.2),matrix(rnorm(150),30,5))
permcoefs.sgpls.CSim(xran, ind=sample(1:nrow(xran)), maxcoefvalues=1e5,
ifbootfail=rep(NA,3))
```

signpred2 *Graphical assessment of the stability of selected variables*

Description

This function is based on the visweb function from the bipartite package.

Usage

```
signpred2(
  matbin,
  pred.lablength = max(sapply(rownames(matbin), nchar)),
  labsize = 1,
  plotsize = 12
)
```

Arguments

matbin	Matrix with 0 or 1 entries. Each row per predictor and a column for every model. 0 means the predictor is not significant in the model and 1 that, on the contrary, it is significant.
pred.lablength	Maximum length of the predictors labels. Defaults to full label length.
labsize	Size of the predictors labels.
plotsize	Global size of the graph.

Value

A plot window.

Author(s)

Bernd Gruber with minor modifications from Frédéric Bertrand
<frederic.bertrand@math.unistra.fr>
https://fbertran.github.io/homepage/

References

Vazquez, P.D., Chacoff, N.,P. and Cagnolo, L. (2009) Evaluating multiple determinants of the structure of plant-animal mutualistic networks. *Ecology*, 90:2039-2046.

See Also

See Also visweb

Examples

```
set.seed(314)
simbin <- matrix(rbinom(200,3,.2),nrow=20,ncol=10)
signpred2(simbin)
```

---

simul_data_UniYX_gamma

Data generating function for univariate gamma plsR models

Description

This function generates a single univariate gamma response value $Ygamma$ and a vector of explanatory variables $(X_1, \ldots, X_{totdim})$ drawn from a model with a given number of latent components.

Usage

```
simul_data_UniYX_gamma(totdim, ncomp, jvar, lvar, link = "inverse", offset = 0)
```

Arguments

totdim	Number of columns of the X vector (from ncomp to hardware limits)
ncomp	Number of latent components in the model (to use noise, select ncomp=3)
jvar	First variance parameter
lvar	Second variance parameter
link	Character specification of the link function in the mean model (mu). Currently, "inverse", "log" and "identity" are supported. Alternatively, an object of class "link-glm" can be supplied.
offset	Offset on the linear scale

Details

This function should be combined with the replicate function to give rise to a larger dataset. The algorithm used is a modification of a port of the one described in the article of Li which is a multivariate generalization of the algorithm of Naes and Martens.

Value

vector	$(Ygamma, X_1, \ldots, X_{totdim})$

Author(s)

Jeremy Magnanensi, Frédéric Bertrand
<frederic.bertrand@utt.fr>
https://fbertran.github.io/homepage/

Jérémy Magnanensi, Frédéric Bertrand
<frederic.bertrand@utt.fr>
https://fbertran.github.io/homepage/

References

T. Naes, H. Martens, Comparison of prediction methods for multicollinear data, Commun. Stat., Simul. 14 (1985) 545-576.
Morris, Elaine B. Martin, Model selection for partial least squares regression, Chemometrics and Intelligent Laboratory Systems 64 (2002), 79-89, doi:10.1016/S01697439(02)000515.

A new bootstrap-based stopping criterion in PLS component construction, J. Magnanensi, M. Maumy-Bertrand, N. Meyer and F. Bertrand (2016), in The Multiple Facets of Partial Least Squares and Related Methods, doi:10.1007/9783319406435_18

A new universal resample-stable bootstrap-based stopping criterion for PLS component construction, J. Magnanensi, F. Bertrand, M. Maumy-Bertrand and N. Meyer, (2017), Statistics and Computing, 27, 757–774. doi:10.1007/s1122201696514

New developments in Sparse PLS regression, J. Magnanensi, M. Maumy-Bertrand, N. Meyer and F. Bertrand, (2021), Frontiers in Applied Mathematics and Statistics, doi:10.3389/fams.2021.693126
.

See Also

simul_data_UniYX

Examples

```
set.seed(314)
ncomp=rep(3,100)
totdimpos=7:50
totdim=sample(totdimpos,100,replace=TRUE)
l=3.01
#for (l in seq(3.01,15.51,by=0.5)) {
j=3.01
#for (j in seq(3.01,9.51,by=0.5))  {
i=44
#for ( i in 1:100){
set.seed(i)
totdimi<-totdim[i]
ncompi<-ncomp[i]
datasim <- t(replicate(200,simul_data_UniYX_gamma(totdimi,ncompi,j,l)))
#}
#}
#}
pairs(datasim)
rm(i,j,l,totdimi,ncompi,datasim)
```

Index