Package ‘enpls’

May 18, 2019

Type Package
Title Ensemble Partial Least Squares Regression
Version 6.1
Maintainer Nan Xiao <me@nanx.me>
Description An algorithmic framework for measuring feature importance,
outlier detection, model applicability domain evaluation,
and ensemble predictive modeling with (sparse)
partial least squares regressions.
License GPL-3 | file LICENSE
URL https://nanx.me/enpls/, https://github.com/nanxstats/enpls
BugReports https://github.com/nanxstats/enpls/issues
Depends R (>= 3.0.2)
LazyData TRUE
VignetteBuilder knitr
Imports pls, spls, foreach, doParallel, ggplot2, reshape2, plotly
Suggests knitr, rmarkdown
Encoding UTF-8
RoxygenNote 6.1.1
NeedsCompilation no
Author Nan Xiao [aut, cre] (<https://orcid.org/0000-0002-0250-5673>),
Dong-Sheng Cao [aut],
Miao-Zhu Li [aut],
Qing-Song Xu [aut]
Repository CRAN
Date/Publication 2019-05-18 04:30:02 UTC
R topics documented:

alkanes ......................................................... 2
cv.enpls .......................................................... 3
cv.enspls .......................................................... 5
enpls.ad ........................................................... 6
enpls.fit ........................................................... 8
enpls.fs ........................................................... 9
enpls.mae .......................................................... 10
enpls.od ........................................................... 11
enpls.rmse .......................................................... 12
enpls.rmsle .......................................................... 13
enspls.ad ........................................................... 13
enspls.fit ........................................................... 15
enspls.fs ........................................................... 17
enspls.od ........................................................... 18
logd1k ............................................................... 19
plot.cv.enpls ...................................................... 20
plot.cv.enspls ..................................................... 21
plot.enpls.ad ....................................................... 22
plot.enpls.fs ....................................................... 23
plot.enpls.od ....................................................... 24
plot.enspls.ad ...................................................... 25
plot.enspls.fs ...................................................... 26
plot.enspls.od ...................................................... 27
predict.enpls.fit ................................................... 28
predict.enspls.fit ................................................... 29
print.cv.enpls ...................................................... 30
print.cv.enspls ..................................................... 31
print.enpls.ad ....................................................... 32
print.enpls.fit ....................................................... 33
print.enpls.fs ....................................................... 34
print.enpls.od ....................................................... 35
print.enspls.ad ...................................................... 35
print.enspls.fit ...................................................... 36
print.enspls.fs ...................................................... 37
print.enspls.od ...................................................... 38

Index 40

<table>
<thead>
<tr>
<th>alkanes</th>
<th>Methylalkanes Retention Index Dataset</th>
</tr>
</thead>
</table>

Description

Methylalkanes retention index dataset from Liang et, al.
Usage
data("alkanes")

Format
A list with 2 components:

• x - data frame with 207 rows (samples) and 21 columns (predictors)
• y - numeric vector of length 207 (response)

Details
This dataset contains 207 methylalkanes’ chromatographic retention index (y) which have been modeled by 21 molecular descriptors (x).

Molecular descriptor types:

• Chi path, cluster and path/cluster indices
• Kappa shape indices
• E-state indices
• Molecular electricity distance vector index

References

Examples
data("alkanes")
str(alkanes)

cv.enpls

Cross Validation for Ensemble Partial Least Squares Regression

Description
K-fold cross validation for ensemble partial least squares regression.

Usage
cv.enpls(x, y, nfolds = 5L, verbose = TRUE, ...)
cv.enpls

Arguments

x  Predictor matrix.
y  Response vector.
nfolds  Number of cross-validation folds, default is 5. Note that this is the CV folds for the ensemble PLS model, not the individual PLS models. To control the CV folds for single PLS models, please use the argument cvfolds.
verbose  Shall we print out the progress of cross-validation?
...  Arguments to be passed to enpls.fit.

Value

A list containing:

• ypred - a matrix containing two columns: real y and predicted y
• residual - cross validation result (y.pred - y.real)
• RMSE - RMSE
• MAE - MAE
• Rsquare - Rsquare

Note

To maximize the probability that each observation can be selected in the test set (thus the prediction uncertainty can be measured), please try setting a large reptimes.

Author(s)

Nan Xiao <https://nanx.me>

See Also

See enpls.fit for ensemble partial least squares regressions.

Examples

data("alkanes")
x <- alkanes$x
y <- alkanes$y

set.seed(42)
cvfit <- cv.enpls(x, y, reptimes = 10)
print(cvfit)
plot(cvfit)
cv.enspls

**Cross Validation for Ensemble Sparse Partial Least Squares Regression**

### Description

K-fold cross validation for ensemble sparse partial least squares regression.

### Usage

```r
cv.enspls(x, y, nfolds = 5L, verbose = TRUE, ...)
```

### Arguments

- **x**: Predictor matrix.
- **y**: Response vector.
- **nfolds**: Number of cross-validation folds, default is 5L. Note that this is the CV folds for the ensemble sparse PLS model, not the individual sparse PLS models. To control the CV folds for single sparse PLS models, please use the argument `cvfolds`.
- **verbose**: Shall we print out the progress of cross-validation?
- **...**: Arguments to be passed to `enspls.fit`.

### Value

A list containing:

- `ypred`: a matrix containing two columns: real y and predicted y
- `residual`: cross validation result (y.pred - y.real)
- `RMSE`: RMSE
- `MAE`: MAE
- `Rsquare`: Rsquare

### Note

To maximize the probability that each observation can be selected in the test set (thus the prediction uncertainty can be measured), please try setting a large `reptimes`.

### Author(s)

Nan Xiao <https://nanx.me>

### See Also

See `enspls.fit` for ensemble sparse partial least squares regressions.
Examples

```r
# This example takes one minute to run
## Not run:
data("logd1k")
x <- logd1k$x
y <- logd1k$y
set.seed(42)
cvfit <- cv.enspls(x, y, reptimes = 10)
print(cvfit)
plot(cvfit)
## End(Not run)
```

Description

Model applicability domain evaluation with ensemble partial least squares.

Usage

```r
enpls.ad(x, y, xtest, ytest, maxcomp = NULL, cvfolds = 5L,
  space = c("sample", "variable"), method = c("mc", "boot"),
  reptimes = 500L, ratio = 0.8, parallel = 1L)
```

Arguments

- `x` Predictor matrix of the training set.
- `y` Response vector of the training set.
- `xtest` List, with the i-th component being the i-th test set’s predictor matrix (see example code below).
- `ytest` List, with the i-th component being the i-th test set’s response vector (see example code below).
- `maxcomp` Maximum number of components included within each model. If not specified, will use the maximum number possible (considering cross-validation and special cases where n is smaller than p).
- `cvfolds` Number of cross-validation folds used in each model for automatic parameter selection, default is 5.
- `space` Space in which to apply the resampling method. Can be the sample space ("sample") or the variable space ("variable").
- `method` Resampling method. "mc" (Monte-Carlo resampling) or "boot" (bootstrap). Default is "mc".
reptimes  Number of models to build with Monte-Carlo resampling or bootstrapping.
ratio    Sampling ratio used when method = "mc".
parallel Integer. Number of CPU cores to use. Default is 1 (not parallelized).

Value
A list containing:

- tr.error.mean - absolute mean prediction error for training set
- tr.error.median - absolute median prediction error for training set
- tr.error.sd  - prediction error sd for training set
- tr.error.matrix - raw prediction error matrix for training set
- te.error.mean - list of absolute mean prediction error for test set(s)
- te.error.median - list of absolute median prediction error for test set(s)
- te.error.sd  - list of prediction error sd for test set(s)
- te.error.matrix - list of raw prediction error matrix for test set(s)

Note
Note that for space = "variable", method could only be "mc", since bootstrapping in the variable space will create duplicated variables, and that could cause problems.

Author(s)
Nan Xiao <https://nanx.me>

Examples
```r
data("alkanes")
x <- alkanes$x
y <- alkanes$y

# training set
x.tr <- x[1:100, ]
y.tr <- y[1:100]

# two test sets
x.te <- list(
    "test.1" = x[101:150, ],
    "test.2" = x[151:207, ]
)
y.te <- list(
    "test.1" = y[101:150],
    "test.2" = y[151:207]
)

set.seed(42)
ad <- enpls.ad(
    x.tr, y.tr, x.te, y.te,
```
space = "variable", method = "mc",  
   ratio = 0.9, reptimes = 50 
)
print(ad)
plot(ad)  
# the interactive plot requires a HTML viewer
## Not run:
plot(ad, type = "interactive")
## End(Not run)

enpls.fit  

Ensemble Partial Least Squares Regression  

Description  

Ensemble partial least squares regression.

Usage  

enpls.fit(x, y, maxcomp = NULL, cvfolds = 5L, reptimes = 500L, 
   method = c("mc", "boot"), ratio = 0.8, parallel = 1L)

Arguments  

x  
Predictor matrix.

y  
Response vector.

maxcomp  
Maximum number of components included within each model. If not specified, will use the maximum number possible (considering cross-validation and special cases where n is smaller than p).

cvfolds  
Number of cross-validation folds used in each model for automatic parameter selection, default is 5.

reptimes  
Number of models to build with Monte-Carlo resampling or bootstrapping.

method  
Resampling method. "mc" (Monte-Carlo resampling) or "boot" (bootstrapping). Default is "mc".

ratio  
Sampling ratio used when method = "mc".

parallel  
Integer. Number of CPU cores to use. Default is 1 (not parallelized).

Value  

A list containing all partial least squares model objects.

Author(s)  

Nan Xiao <https://nanx.me>
See Also

See `enpls.fs` for measuring feature importance with ensemble partial least squares regressions. See `enpls.od` for outlier detection with ensemble partial least squares regressions.

Examples

```r
data("alkanes")
x <- alkanes$x
y <- alkanes$y

set.seed(42)
fit <- enpls.fit(x, y, reptimes = 50)
print(fit)
predict(fit, newx = x)
```

```
enpls.fs  Ensemble Partial Least Squares for Measuring Feature Importance

Description

Measuring feature importance with ensemble partial least squares.

Usage

`enpls.fs(x, y, maxcomp = NULL, cvfolds = 5L, reptimes = 500L, method = c("mc", "boot"), ratio = 0.8, parallel = 1L)`

Arguments

- `x`: Predictor matrix.
- `y`: Response vector.
- `maxcomp`: Maximum number of components included within each model. If not specified, will use the maximum number possible (considering cross-validation and special cases where n is smaller than p).
- `cvfolds`: Number of cross-validation folds used in each model for automatic parameter selection, default is 5.
- `reptimes`: Number of models to build with Monte-Carlo resampling or bootstrapping.
- `method`: Resampling method. "mc" (Monte-Carlo resampling) or "boot" (bootstrapping). Default is "mc".
- `ratio`: Sampling ratio used when method = "mc".
- `parallel`: Integer. Number of CPU cores to use. Default is 1 (not parallelized).
Value
A list containing two components:

• variable.importance - a vector of variable importance
• coefficient.matrix - original coefficient matrix

Author(s)
Nan Xiao <https://nanx.me>

See Also
See enpls.od for outlier detection with ensemble partial least squares regressions. See enpls.fit for fitting ensemble partial least squares regression models.

Examples
```r
data("alkanes")
x <- alkanes$x
y <- alkanes$y
set.seed(42)
fs <- enpls.fs(x, y, reptimes = 50)
print(fs)
plot(fs)
```

---

enpls.mae  

Mean Absolute Error (MAE)

Description
Mean Absolute Error (MAE)

Usage

```r
enpls.mae(yreal, ypred)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>yreal</td>
<td>true response vector</td>
</tr>
<tr>
<td>ypred</td>
<td>predicted response vector</td>
</tr>
</tbody>
</table>

Value

MAE

Author(s)
Nan Xiao <https://nanx.me>
Ensemble Partial Least Squares for Outlier Detection

Description
Outlier detection with ensemble partial least squares.

Usage
enpls.od(x, y, maxcomp = NULL, cvfolds = 5L, reptimes = 500L, 
method = c("mc", "boot"), ratio = 0.8, parallel = 1L)

Arguments

x     Predictor matrix.
y     Response vector.
maxcomp Maximum number of components included within each model. If not specified, 
         will use the maximum number possible (considering cross-validation and special 
         cases where n is smaller than p).
cvfolds Number of cross-validation folds used in each model for automatic parameter 
         selection, default is 5.
reptimes Number of models to build with Monte-Carlo resampling or bootstrapping.
method Resampling method. "mc" (Monte-Carlo resampling) or "boot" (bootstrap) 
         (bootstrapping). Default is "mc".
ratio Sampling ratio used when method = "mc".
parallel Integer. Number of CPU cores to use. Default is 1 (not parallelized).

Value
A list containing four components:

- error.mean - error mean for all samples (absolute value)
- error.median - error median for all samples
- error.sd - error sd for all samples
- predict.error.matrix - the original prediction error matrix

Note
To maximize the probability that each observation can be selected in the test set (thus the prediction 
uncertainty can be measured), please try setting a large reptimes.

Author(s)
Nan Xiao <https://nanx.me>
See Also

See `enpls.fs` for measuring feature importance with ensemble partial least squares regressions. See `enpls.fit` for fitting ensemble partial least squares regression models.

Examples

data("alkanes")
x <- alkanes$x
y <- alkanes$y

set.seed(42)
od <- enpls.od(x, y, reptimes = 50)
print(od)
plot(od)
plot(od, criterion = "sd")

---

### enpls.rmse

**Root Mean Squared Error (RMSE)**

#### Description

Compute Root Mean Squared Error (RMSE).

#### Usage

```r
enpls.rmse(yreal, ypred)
```

#### Arguments

- `yreal`  
  true response vector

- `ypred`  
  predicted response vector

#### Value

RMSE

#### Author(s)

Nan Xiao <https://nanx.me>
enpls.rmsle

Root Mean Squared Logarithmic Error (RMSLE)

Description

Root Mean Squared Logarithmic Error (RMSLE)

Usage

enpls.rmsle(yreal, ypred)

Arguments

yreal true response vector
ypred predicted response vector

Value

RMSLE

Author(s)

Nan Xiao <https://nanx.me>

enspls.ad

Ensemble Sparse Partial Least Squares for Model Applicability Domain Evaluation

Description

Model applicability domain evaluation with ensemble sparse partial least squares.

Usage

enspls.ad(x, y, xtest, ytest, maxcomp = 5L, cvfolds = 5L, alpha = seq(0.2, 0.8, 0.2), space = c("sample", "variable"), method = c("mc", "boot"), reptimes = 500L, ratio = 0.8, parallel = 1L)
Arguments

x  Predictor matrix of the training set.
y  Response vector of the training set.
xtest  List, with the i-th component being the i-th test set’s predictor matrix (see example code below).
ytest  List, with the i-th component being the i-th test set’s response vector (see example code below).
maxcomp  Maximum number of components included within each model. If not specified, will use 5 by default.
cvfolds  Number of cross-validation folds used in each model for automatic parameter selection, default is 5.
alpha  Parameter (grid) controlling sparsity of the model. If not specified, default is seq(0.2, 0.8, 0.2).
space  Space in which to apply the resampling method. Can be the sample space ("sample") or the variable space ("variable").
method  Resampling method. "mc" (Monte-Carlo resampling) or "boot" (bootstrapping). Default is "mc".
reptimes  Number of models to build with Monte-Carlo resampling or bootstrapping.
ratio  Sampling ratio used when method = "mc".
parallel  Integer. Number of CPU cores to use. Default is 1 (not parallelized).

Value

A list containing:

- tr.error.mean - absolute mean prediction error for training set
- tr.error.median - absolute median prediction error for training set
- tr.error.sd - prediction error sd for training set
- tr.error.matrix - raw prediction error matrix for training set
- te.error.mean - list of absolute mean prediction error for test set(s)
- te.error.median - list of absolute median prediction error for test set(s)
- te.error.sd - list of prediction error sd for test set(s)
- te.error.matrix - list of raw prediction error matrix for test set(s)

Note

Note that for space = "variable", method could only be "mc", since bootstrapping in the variable space will create duplicated variables, and that could cause problems.

Author(s)

Nan Xiao <https://nanx.me>
Examples

data("logd1k")
# remove low variance variables
x <- logd1k$x[-c(17, 52, 59)]
y <- logd1k$y

# training set
x.tr <- x[1:300, ]
y.tr <- y[1:300]

# two test sets
x.te <- list(
  "test.1" = x[301:400, ],
  "test.2" = x[401:500, ]
)
y.te <- list(
  "test.1" = y[301:400],
  "test.2" = y[401:500]
)

set.seed(42)
ad <- enspls.fit(x.tr, y.tr, x.te, y.te, maxcomp = 3, alpha = c(0.3, 0.6, 0.9),
  space = "variable", method = "mc",
  ratio = 0.8, reptimes = 10)
print(ad)
plot(ad)
# the interactive plot requires a HTML viewer
## Not run:
plot(ad, type = "interactive")
## End(Not run)

enspls.fit

Ensemble Sparse Partial Least Squares Regression

Description

Ensemble sparse partial least squares regression.

Usage

enspls.fit(x, y, maxcomp = 5L, cvfolds = 5L, alpha = seq(0.2, 0.8,
  0.2), reptimes = 500L, method = c("mc", "boot"), ratio = 0.8,
  parallel = 1L)
Arguments

- `x`: Predictor matrix.
- `y`: Response vector.
- `maxcomp`: Maximum number of components included within each model. If not specified, will use 5 by default.
- `cvfolds`: Number of cross-validation folds used in each model for automatic parameter selection, default is 5.
- `alpha`: Parameter (grid) controlling sparsity of the model. If not specified, default is `seq(0.2, 0.8, 0.2)`.
- `reptimes`: Number of models to build with Monte-Carlo resampling or bootstrapping.
- `method`: Resampling method. "mc" (Monte-Carlo resampling) or "boot" (bootstrapping). Default is "mc".
- `ratio`: Sampling ratio used when `method` = "mc".
- `parallel`: Integer. Number of CPU cores to use. Default is 1 (not parallelized).

Value

A list containing all sparse partial least squares model objects.

Author(s)

Nan Xiao <https://nanx.me>

See Also

See `enspls.fs` for measuring feature importance with ensemble sparse partial least squares regressions. See `enspls.od` for outlier detection with ensemble sparse partial least squares regressions.

Examples

data("logd1k")
x <- logd1k$x
y <- logd1k$y

set.seed(42)
fit <- enspls.fit(
x, y, reptimes = 5, maxcomp = 3, alpha = c(0.3, 0.6, 0.9)
)
print(fit)
predict(fit, newx = x)
Description

Measuring feature importance with ensemble sparse partial least squares.

Usage

```r
enspls.fs(x, y, maxcomp = 5L, cvfolds = 5L, alpha = seq(0.2, 0.8, 0.2), reptimes = 500L, method = c("mc", "boot"), ratio = 0.8, parallel = 1L)
```

Arguments

- `x`: Predictor matrix.
- `y`: Response vector.
- `maxcomp`: Maximum number of components included within each model. If not specified, will use 5 by default.
- `cvfolds`: Number of cross-validation folds used in each model for automatic parameter selection, default is 5.
- `alpha`: Parameter (grid) controlling sparsity of the model. If not specified, default is `seq(0.2, 0.8, 0.2)`.
- `reptimes`: Number of models to build with Monte-Carlo resampling or bootstrapping.
- `method`: Resampling method. "mc" (Monte-Carlo resampling) or "boot" (bootstrapping). Default is "mc".
- `ratio`: Sampling ratio used when `method = "mc"`.
- `parallel`: Integer. Number of CPU cores to use. Default is 1 (not parallelized).

Value

A list containing two components:

- `variable.importance` - a vector of variable importance
- `coefficient.matrix` - original coefficient matrix

Author(s)

Nan Xiao <https://nanx.me>

See Also

See `enspls.od` for outlier detection with ensemble sparse partial least squares regressions. See `enspls.fit` for fitting ensemble sparse partial least squares regression models.
Examples

```r
data("logd1k")
x <- logd1k$x
y <- logd1k$y
set.seed(42)
fs <- enspls.fs(x, y, reptimes = 5, maxcomp = 2)
print(fs, nvar = 10)
plot(fs, nvar = 10)
plot(fs, type = "boxplot", limits = c(0.05, 0.95), nvar = 10)
```

---

**enspls.od**  
**Ensemble Sparse Partial Least Squares for Outlier Detection**

---

**Description**

Outlier detection with ensemble sparse partial least squares.

**Usage**

```r
enspls.od(x, y, maxcomp = 5L, cvfolds = 5L, alpha = seq(0.2, 0.8, 0.2), reptimes = 500L, method = c("mc", "boot"), ratio = 0.8, parallel = 1L)
```

**Arguments**

- **x**  
  Predictor matrix.

- **y**  
  Response vector.

- **maxcomp**  
  Maximum number of components included within each model. If not specified, will use 5 by default.

- **cvfolds**  
  Number of cross-validation folds used in each model for automatic parameter selection, default is 5.

- **alpha**  
  Parameter (grid) controlling sparsity of the model. If not specified, default is seq(0.2, 0.8, 0.2).

- **reptimes**  
  Number of models to build with Monte-Carlo resampling or bootstrapping.

- **method**  
  Resampling method. "mc" (Monte-Carlo resampling) or "boot" (bootstrapping). Default is "mc".

- **ratio**  
  Sampling ratio used when method = "mc".

- **parallel**  
  Integer. Number of CPU cores to use. Default is 1 (not parallelized).
Value

A list containing four components:

- error.mean - error mean for all samples (absolute value)
- error.median - error median for all samples
- error.sd - error sd for all samples
- predict.error.matrix - the original prediction error matrix

Note

To maximize the probability that each observation can be selected in the test set (thus the prediction uncertainty can be measured), please try setting a large reptimes.

Author(s)

Nan Xiao <https://nanx.me>

See Also

See `enspls.fs` for measuring feature importance with ensemble sparse partial least squares regressions. See `enspls.fit` for fitting ensemble sparse partial least squares regression models.

Examples

```r
data("logd1k")
x <- logd1k$x
y <- logd1k$y

set.seed(42)
od <- enspls.od(
x, y,
  reptimes = 5, maxcomp = 3,
  alpha = c(0.3, 0.6, 0.9)
)
plot(od, prob = 0.1)
plot(od, criterion = "sd", sdtimes = 1)
```

Description

Distribution coefficients at pH 7.4 (logD7.4) dataset from Wang et al.

Usage

data(logd1k)
Format

A list with 2 components:

• x - data frame with 1,000 rows (samples) and 80 columns (predictors)
• y - numeric vector of length 1,000 (response)

The first 1000 compounds in the original dataset were selected.

Details

This dataset contains distribution coefficients at pH 7.4 (logD7.4) for 1,000 compounds, and 80 molecular descriptors computed with RDKit.

References


Examples

data(logd1k)
str(logd1k)

Description

Plot cv.enpls object

Usage

## S3 method for class 'cv.enpls'
plot(x, xlim = NULL, ylim = NULL, alpha = 0.8,
     main = NULL, ...)

Arguments

x An object of class cv.enpls.
xlim x Vector of length 2 - x axis limits of the plot.
ylim y Vector of length 2 - y axis limits of the plot.
alpha An alpha transparency value for points, a real number in (0, 1].
main Plot title, not used currently.
... Additional graphical parameters, not used currently.
Author(s)
Nan Xiao <https://nanx.me>

See Also
See `cv.enpls` for cross-validation of ensemble partial least squares regression models.

Examples
data("alkanes")
x <- alkanes$x
y <- alkanes$y
set.seed(42)
cvfit <- cv.enpls(x, y, reptimes = 10)
plot(cvfit)

plot.cv.enpls

Description
Plot cv.enpls object

Usage
## S3 method for class 'cv.enpls'
plot(x, xlim = NULL, ylim = NULL, alpha = 0.8,
     main = NULL, ...)

Arguments

  x           An object of class cv.enpls.
xlim         x Vector of length 2 - x axis limits of the plot.
ylim         y Vector of length 2 - y axis limits of the plot.
alpha        An alpha transparency value for points, a real number in (0, 1].
main         Plot title, not used currently.
...          Additional graphical parameters, not used currently.

Author(s)
Nan Xiao <https://nanx.me>

See Also
See `cv.enpls` for cross-validation of ensemble sparse partial least squares regression models.
Examples

```r
# This example takes one minute to run
## not run:
data("logd1k")
x <- logd1k$x
y <- logd1k$y

set.seed(42)
cvfit <- cv.enspls(x, y, reptimes = 10)
plot(cvfit)

## End(not run)
```

---

**plot.enpls.ad**

*Plot enpls.ad object*

### Description

Plot enpls.ad object

### Usage

```r
## S3 method for class 'enpls.ad'
plot(x, type = c("static", "interactive"),
     main = NULL, ...)
```

### Arguments

- **x**: An object of class enpls.ad.
- **type**: Plot type. Can be "static" or "interactive".
- **main**: Plot title, not used currently.
- **...**: Additional graphical parameters, not used currently.

### Author(s)

Nan Xiao &lt;https://nanx.me&gt;

### See Also

See enpls.ad for model applicability domain evaluation with ensemble partial least squares regressions.
plot.enpls.fs  

**Examples**

```r
data("alkanes")
x <- alkanes$x
y <- alkanes$y

# training set
x.tr <- x[1:100, ]
y.tr <- y[1:100]

# two test sets
x.te <- list("test.1" = x[101:150, ],
             "test.2" = x[151:207, ]
           )
y.te <- list("test.1" = y[101:150],
             "test.2" = y[151:207]
           )

set.seed(42)
ad <- enpls.ad(
x.tr, y.tr, x.te, y.te,
   space = "variable", method = "mc",
   ratio = 0.9, reptimes = 50
)
plot(ad)
```

# the interactive plot requires a HTML viewer
## Not run:
plot(ad, type = "interactive")
## End(Not run)

---

**plot.enpls.fs**  
*Plot enpls.fs object*

**Description**

Plot enpls.fs object

**Usage**

```r
## S3 method for class 'enpls.fs'
plot(x, nvar = NULL, type = c("dotplot", "boxplot"),
    limits = c(0, 1), main = NULL, ...)
```

**Arguments**

- `x`  
  An object of class enpls.fs.
- `nvar`  
  Number of top variables to show. Ignored if `sort = FALSE`. 
type  
limits  
main  
...  

Author(s)

Nan Xiao <https://nanx.me>

See Also

See enpls.fs for measuring feature importance with ensemble partial least squares regressions.

Examples

data("alkanes")
x <- alkanes$x
y <- alkanes$y

set.seed(42)
fs <- enpls.fs(x, y, reptimes = 50)
plot(fs)
plot(fs, nvar = 10)
plot(fs, type = "boxplot")
plot(fs, type = "boxplot", limits = c(0.05, 0.95))

plot.enpls.od  

Description

Plot enpls.od object

Usage

## S3 method for class 'enpls.od'
plot(x, criterion = c("quantile", "sd"),
     prob = 0.05, sdtimes = 3L, alpha = 1, main = NULL, ...)

Arguments

x  
criterion  
prob  
sdtimes  
alpha  
main  
...  

Additional graphical parameters for plot.
plot.enspls.ad

Author(s)
Nan Xiao <https://nanx.me>

See Also
See enpls.od for outlier detection with ensemble partial least squares regressions.

Examples

data("alkanes")
x <- alkanes$x
y <- alkanes$y

set.seed(42)
od <- enpls.od(x, y, reptimes = 50)
plot(od, criterion = "quantile")
plot(od, criterion = "sd")

Description
Plot enspls.ad object

Usage
## S3 method for class 'enspls.ad'
plot(x, type = c("static", "interactive"),
     main = NULL, ...)

Arguments
x An object of class enspls.ad.
type Plot type. Can be "static" or "interactive".
main Plot title.
... Additional graphical parameters for plot.

Author(s)
Nan Xiao <https://nanx.me>

See Also
See enpls.ad for model applicability domain evaluation with ensemble sparse partial least squares regressions.
Examples

data("logd1k")
# remove low variance variables
x <- logd1k$x[, -c(17, 52, 59)]
y <- logd1k$y

# training set
x.tr <- x[1:300,]
y.tr <- y[1:300]

# two test sets
x.te <- list(
  "test.1" = x[301:400,],
  "test.2" = x[401:500,]
)
y.te <- list(
  "test.1" = y[301:400],
  "test.2" = y[401:500]
)

set.seed(42)
ad <- enspls.ad(
  x.tr, y.tr, x.te, y.te,
  maxcomp = 3, alpha = c(0.3, 0.6, 0.9),
  space = "variable", method = "mc",
  ratio = 0.8, reptimes = 10
)
plot(ad)
# the interactive plot requires a HTML viewer
## Not run:
plot(ad, type = "interactive")
## End(Not run)

---

plot.enspls.fs

Plot enspls.fs object

Description

Plot enspls.fs object

Usage

## S3 method for class 'enspls.fs'
plot(x, nvar = NULL, type = c("dotplot", "boxplot"), limits = c(0, 1), main = NULL, ...)
Arguments

x          An object of class enspls.fs.
nvar       Number of top variables to show. Ignored if sort = FALSE.
type       Plot type, can be "dotplot" or "boxplot".
limits     Vector of length 2. Set boxplot limits (in quantile) to remove the extreme outlier coefficients.
main       Plot title, not used currently.
...        Additional graphical parameters, not used currently.

Author(s)

Nan Xiao <https://nanx.me>

See Also

See enspls.fs for measuring feature importance with ensemble sparse partial least squares regressions.

Examples

data("logd1k")
x <- logd1k$x
y <- logd1k$y

set.seed(42)
fs <- enspls.fs(x, y, reptimes = 5, maxcomp = 2)
plot(fs, nvar = 10)
plot(fs, type = "boxplot", limits = c(0.05, 0.95), nvar = 10)
predict.enpls.fit

Message

Arguments

- \texttt{x} \hspace{1cm} \text{An object of class enpls.od.}
- \texttt{criterion} \hspace{1cm} \text{Criterion of being classified as an outlier, can be "quantile" or "sd".}
- \texttt{prob} \hspace{1cm} \text{Quantile probability as the cut-off value.}
- \texttt{sdtimes} \hspace{1cm} \text{Times of standard deviation as the cut-off value.}
- \texttt{alpha} \hspace{1cm} \text{An alpha transparency value for points, a real number in (0, 1].}
- \texttt{main} \hspace{1cm} \text{Plot title.}
- \texttt{...} \hspace{1cm} \text{Additional graphical parameters for \texttt{plot}.}

Author(s)

Nan Xiao <https://nanx.me>

See Also

See \texttt{enspls.od} for outlier detection with ensemble sparse partial least squares regressions.

Examples

```r
data("logd1k")
x <- logd1k$x
y <- logd1k$y

set.seed(42)
od <- enspls.od(x, y, reptimes = 4, maxcomp = 2)
plot(od, criterion = "quantile", prob = 0.1)
plot(od, criterion = "sd", sdtimes = 1)
```

predict.enpls.fit \hspace{1cm} \textit{Make Predictions from a Fitted Ensemble Partial Least Squares Model}

Description

Make predictions on new data by fitted enpls.fit object.

Usage

```r
## S3 method for class 'enpls.fit'
predict(object, newx, method = c("mean", "median"),
    ...
)
```

Arguments

- \texttt{object} \hspace{1cm} \text{An object of class enpls.fit.}
- \texttt{newx} \hspace{1cm} \text{New data to predict with.}
- \texttt{method} \hspace{1cm} \text{Use "mean" or "median" to create the final prediction.}
- \texttt{...} \hspace{1cm} \text{Additional parameters for \texttt{predict}.}
Value

A numeric vector containing the predicted values.

Author(s)

Nan Xiao <https://nanx.me>

See Also

See `enpls.fit` for fitting ensemble partial least squares regression models.

Examples

data("alkanes")
x <- alkanes$x
y <- alkanes$y

set.seed(42)
fit <- enpls.fit(x, y, reptimes = 50)
y.pred <- predict(fit, newx = x)
plot(y, y.pred, xlim = range(y), ylim = range(y))
abline(a = 0L, b = 1L)
y.pred.med <- predict(fit, newx = x, method = "median")
plot(y, y.pred.med, xlim = range(y), ylim = range(y))
abline(a = 0L, b = 1L)
Value
A numeric vector containing the predicted values.

Author(s)
Nan Xiao <https://nanx.me>

See Also
See `enspls.fit` for fitting ensemble sparse partial least squares regression models.

Examples
```r
data("logd1k")
x <- logd1k$x
y <- logd1k$y

set.seed(42)
fit <- enspls.fit(x, y, reptimes = 5, maxcomp = 2)
y.pred <- predict(fit, newx = x)
plot(y, y.pred, xlab = "x", ylab = "y")
abline(a = 0L, b = 1L)

y.pred.med <- predict(fit, newx = x, method = "median")
plot(y, y.pred.med, xlab = "x", ylab = "y")
abline(a = 0L, b = 1L)
```

See Also

See `cv.enpls` for cross-validation of ensemble partial least squares regression models.

Examples

data(\texttt{alkanes})
x <- alkanes$x
y <- alkanes$y

set.seed(42)
cvfit <- cv.enpls(x, y, reptimes = 10)
cvfit

print.cv.enpls  \hspace{1cm} \textit{Print cv.enpls Object}

Description

Print cv.enpls object.

Usage

\texttt{## S3 method for class 'cv.enpls'}
\texttt{print(x, \ldots)}

Arguments

\texttt{x} \hspace{1cm} An object of class \texttt{cv.enpls}.

\texttt{\ldots} \hspace{1cm} Additional parameters for \texttt{print}.

Author(s)

Nan Xiao <https://nan.x.me>

See Also

See \texttt{cv.enpls} for cross-validation of ensemble sparse partial least squares regression models.

Examples

\# This example takes one minute to run
\# Not run:
data(\texttt{logd1k})
x <- logd1k$x
y <- logd1k$y

set.seed(42)
cvfit <- cv.enpls(x, y, reptimes = 10)
print(cvfit)

## End(Not run)

---

**print.enpls.ad**

---

**Description**

Print enpls.ad object.

**Usage**

```r
## S3 method for class 'enpls.ad'
print(x, ...)
```

**Arguments**

- `x` An object of class `enpls.ad`.
- `...` Additional parameters for `print`.

**Author(s)**

Nan Xiao &lt;https://nanx.me&gt;

**See Also**

See `enpls.ad` for model applicability domain evaluation with ensemble partial least squares regressions.

**Examples**

```r
data("alkanes")
x <- alkanes$x
y <- alkanes$y

# training set
x.tr <- x[1:100,]
y.tr <- y[1:100]

# two test sets
x.te <- list(
  "test.1" = x[101:150,],
  "test.2" = x[151:207,]
)
y.te <- list(
  "test.1" = y[101:150],
  "test.2" = y[151:207]
)
```
print.enpls.fit

set.seed(42)
ad <- enpls.ad(
  x.tr, y.tr, x.te, y.te,
  space = "variable", method = "mc",
  ratio = 0.9, reptimes = 50
)
ad

print.enpls.fit

Print Fitted Ensemble Partial Least Squares Object

Description

Print coefficients of each model in the enpls.fit object.

Usage

## S3 method for class 'enpls.fit'
print(x, ...)

Arguments

x An object of class enpls.fit.

... Additional parameters for print.

Author(s)

Nan Xiao <https://nanx.me>

See Also

See enpls.fit for fitting ensemble partial least squares regression models.

Examples

data("alkanes")
x <- alkanes$x
y <- alkanes$y

set.seed(42)
fit <- enpls.fit(x, y, reptimes = 50)
fit
**Description**

Print enpls.fs object.

**Usage**

```r
## S3 method for class 'enpls.fs'
print(x, sort = TRUE, nvar = NULL, ...)
```

**Arguments**

- `x`: An object of class `enpls.fs`.
- `sort`: Should the variables be sorted in decreasing order of importance?
- `nvar`: Number of top variables to show. Ignored if `sort = FALSE`.
- `...`: Additional parameters for `print`.

**Author(s)**

Nan Xiao <https://nanx.me>

**See Also**

See `enpls.fs` for measuring feature importance with ensemble partial least squares regressions.

**Examples**

```r
data(alkanes)
x <- alkanes$x
y <- alkanes$y

set.seed(42)
fs <- enpls.fs(x, y, reptimes = 100)
print(fs)
print(fs, nvar = 10L)
```
print.enpls.od  

Print enpls.od Object

Description
Print enpls.od object.

Usage
## S3 method for class 'enpls.od'
print(x, ...)

Arguments
x  An object of class enpls.od.
...  Additional parameters for print.

Author(s)
Nan Xiao <https://nanx.me>

See Also
See enpls.od for outlier detection with ensemble partial least squares regressions.

Examples
data("alkanes")
x <- alkanes$x
y <- alkanes$y

set.seed(42)
od <- enpls.od(x, y, reptimes = 40)
od

print.enspls.ad  

Print enspls.ad Object

Description
Print enspls.ad object.

Usage
## S3 method for class 'enspls.ad'
print(x, ...)


print.enspls.fit

Arguments

- **x**
  - An object of class `enspls.ad`.

- **...**
  - Additional parameters for `print`.

Author(s)

Nan Xiao <https://nanx.me>

See Also

See `enspls.ad` for model applicability domain evaluation with ensemble sparse partial least squares regressions.

Examples

```r
data("logd1k")
# remove low variance variables
x <- logd1k$x[,-c(17, 52, 59)]
y <- logd1k$y

# training set
x.tr <- x[1:300, ]
y.tr <- y[1:300]

# two test sets
x.te <- list(
  "test.1" = x[301:400, ],
  "test.2" = x[401:500, ]
)
y.te <- list(
  "test.1" = y[301:400],
  "test.2" = y[401:500]
)

set.seed(42)
ad <- enspls.ad(
  x.tr, y.tr, x.te, y.te,
  maxcomp = 3, alpha = c(0.3, 0.6, 0.9),
  space = "variable", method = "mc",
  ratio = 0.8, reptimes = 10
)
print(ad)
```

---

**print.enspls.fit**

*Print Fitted Ensemble Sparse Partial Least Squares Object*

Description

Print coefficients of each model in the enspls.fit object.
print.enspls.fs

Usage

## S3 method for class 'enspls.fit'
print(x, ...)

Arguments

x An object of class enspls.fit.

... Additional parameters for print.

Author(s)

Nan Xiao <https://nanx.me>

See Also

See enspls.fit for fitting ensemble sparse partial least squares regression models.

Examples

data("logd1k")
x <- logd1k$x
y <- logd1k$y

set.seed(42)
fit <- enspls.fit(
  x, y,
  reptimes = 5, maxcomp = 3,
  alpha = c(0.3, 0.6, 0.9)
)
print(fit)

print.enspls.fs  Print enspls.fs Object

Description

Print enspls.fs object.

Usage

## S3 method for class 'enspls.fs'
print(x, sort = TRUE, nvar = NULL, ...)

Arguments

x An object of class enspls.fs.

sort Should the variables be sorted in decreasing order of importance?

nvar Number of top variables to show. Ignored if sort = FALSE.

... Additional parameters for print.
Author(s)
Nan Xiao <https://nanx.me>

See Also
See enspls.fs for measuring feature importance with ensemble sparse partial least squares regressions.

Examples
```r
data("logd1k")
x <- logd1k$x
y <- logd1k$y

set.seed(42)
fs <- enspls.fs(
  x, y,
  reptimes = 5, maxcomp = 3,
  alpha = c(0.3, 0.6, 0.9)
)
print(fs, nvar = 10L)
```

---

**print.enspls.od**  
*Print enspls.od Object*

Description
Print enspls.od object.

Usage
```r
## S3 method for class 'enspls.od'
print(x, ...)
```

Arguments
- **x**  
  An object of class enspls.od.
- **...**  
  Additional parameters for print.

Author(s)
Nan Xiao <https://nanx.me>

See Also
See enspls.od for outlier detection with ensemble sparse partial least squares regressions.
Examples

data("logd1k")
x <- logd1k$x
y <- logd1k$y

set.seed(42)
od <- enspls.od(
  x, y,
  reptimes = 5, maxcomp = 3,
  alpha = c(0.3, 0.6, 0.9)
)
print(od)
Index

alkanes, 2

cv.enpls, 3, 21, 31

cv.enplsfs, 5, 21, 31

enpls.ad, 6, 22, 32

enpls.fit, 4, 8, 10, 12, 29, 33

enpls.fs, 9, 9, 12, 24, 34

enpls.mae, 10

enpls.od, 9, 10, 11, 25, 35

enpls.rmse, 12

enpls.rmsle, 13

enspls.ad, 13, 25, 36

enspls.fit, 5, 15, 17, 19, 30, 37

enspls.fs, 16, 17, 19, 27, 38

enspls.od, 16, 17, 18, 28, 38

logd1k, 19

plot, 24, 25, 28

plot.cv.enpls, 20

plot.cv.enplsfs, 21

plot.enpls.ad, 22

plot.enpls.fs, 23

plot.enpls.od, 24

plot.enpls.ad, 25

plot.enpls.fs, 26

plot.enpls.od, 27

predict, 28, 29

predict.enpls.fit, 28

predict.enspls.fit, 29

print, 30–38

print.cv.enpls, 30

print.cv.enplsfs, 31

print.enpls.ad, 32

print.enpls.fit, 33

print.enpls.fs, 34

print.enpls.od, 35

print.enspls.ad, 35

print.enspls.fit, 36

print.enspls.fs, 37

print.enspls.od, 38