Package 'enpls'

September 27, 2017

Type Package

Title Ensemble Partial Least Squares Regression

Version 5.9

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://enpls.org, https://github.com/road2stat/enpls

BugReports https://github.com/road2stat/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 6.0.1.9000

NeedsCompilation no

Author Nan Xiao [aut, cre],
Dong-Sheng Cao [aut],
Miao-Zhu Li [aut],
Qing-Song Xu [aut]

Repository CRAN

Date/Publication 2017-09-27 20:13:25 UTC

R topics documented:

enpls-package *Ensemble (Sparse) Partial Least Squares Regressions*

Description

An algorithmic framework for measuring feature importance, outlier detection, model applicability evaluation, and ensemble predictive modeling with (sparse) partial least squares regressions.

Details

Browse the package vignette via `vignette("enpls")`.

Package:	enpls
Type:	Package
License:	GPL-3

alkanes *Methylalkanes Retention Index Dataset*

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

Yi-Zeng Liang, Da-Lin Yuan, Qing-Song Xu, and Olav Martin Kvalheim. "Modeling based on subspace orthogonal projections for QSAR and QSPR research." *Journal of Chemometrics* 22, no. 1 (2008): 23–35.

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, ...)
```

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 probablity 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

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

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 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 probablity 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

```
# 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)
```

---

enpls.ad *Ensemble Partial Least Squares for Model Applicability Domain Evaluation*

Description

Model applicability domain evaluation with ensemble partial least squares.

Usage

```
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" (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("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

```
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

```
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

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

Arguments

yreal	true response vector
ypred	predicted response vector

Value

MAE

Author(s)

Nan Xiao <https://nanx.me>

enpls.od *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"` (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 probablity 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

```
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.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)
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)
```

`enspls.fs` *Ensemble Sparse Partial Least Squares for Measuring Feature Importance*

Description

Measuring feature importance with ensemble sparse partial least squares.

Usage

```
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

```
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

```
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 probablity 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

```
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)
```

---

logd1k *logD7.4 Data for 1,000 Compounds*

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

Jian-Bing Wang, Dong-Sheng Cao, Min-Feng Zhu, Yong-Huan Yun, Nan Xiao, and Yi-Zeng Liang. "In silico evaluation of logD7.4 and comparison with other prediction methods." *Journal of Chemometrics* 29, no. 7 (2015): 389–398.

Examples

```
data(logd1k)
str(logd1k)
```

---

plot.cv.enpls *Plot cv.enpls object*

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.enspls *Plot cv.enspls object*

Description

Plot cv.enspls object

Usage

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

Arguments

x	An object of class cv.enspls.
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.enspls for cross-validation of ensemble sparse partial least squares regression models.

Examples

```
# 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

```
## 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 <https://nanx.me>

See Also

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

Examples

```
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

```
## 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`	Plot type. `"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 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 *Plot enpls.od object*

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

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")
```

---

plot.enspls.ad *Plot enspls.ad object*

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 enspls.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)
```

---

plot.enspls.od *Plot enspls.od object*

Description

Plot enspls.od object

Usage

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

Arguments

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

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 = 4, maxcomp = 2)
plot(od, criterion = "quantile", prob = 0.1)
plot(od, criterion = "sd", sdtimes = 1)
```

---

predict.enpls.fit *Make Predictions from a Fitted Ensemble Partial Least Squares Model*

Description

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

Usage

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

Arguments

object	An object of class enpls.fit.
newx	New data to predict with.
method	Use "mean" or "median" to create the final prediction.
...	Additional parameters for 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)
```

`predict.enspls.fit` *Make Predictions from a Fitted Sparse Ensemble Partial Least Squares Model*

Description

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

Usage

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

Arguments

`object`	An object of class `enspls.fit`.
`newx`	New data to predict with.
`method`	Use `"mean"` or `"median"` to create the final prediction.
`...`	Additional parameters for `predict`.

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

```
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, 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)
```

---

print.cv.enpls *Print cv.enpls Object*

Description

Print cv.enpls object.

Usage

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

Arguments

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

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)
print(cvfit)
```

`print.cv.enspls` *Print cv.enspls Object*

Description

Print cv.enspls object.

Usage

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

Arguments

`x`	An object of class `cv.enspls`.
`...`	Additional parameters for `print`.

Author(s)

Nan Xiao <https://nanx.me>

See Also

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

Examples

```
# 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)
## End(Not run)
```

`print.enpls.ad` *Print enpls.ad Object*

Description

Print enpls.ad object.

Usage

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

Arguments

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

Author(s)

Nan Xiao <https://nanx.me>

See Also

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

Examples

```
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)
```

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)
print(fit)
```

print.enpls.fs *Print enpls.fs Object*

Description

Print enpls.fs object.

Usage

```
## 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

```
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)
print(od)
```

---

print.enspls.ad *Print enspls.ad Object*

Description

Print enspls.ad object.

Usage

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

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

```
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.

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

```
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

```
## 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