Package ‘upsdr’

April 30, 2024

Type Package

Title Unified Principal Sufficient Dimension Reduction Package

Version 1.0.0

Author Jungmin Shin, Andreas Artemiou, Seung Jun Shin

Maintainer Jungmin Shin <jungminshin@korea.ac.kr>

Description
A unified and user-friendly framework for applying the principal sufficient dimension reduction methods for both linear and nonliear cases. The package has an extendable power by varying loss functions for the SVM, even for an user-defined arbitrary function, unless those are convex and differentiable everywhere over the support. Also, it provides a realtime sufficient dimension reduction update procedure using the principal least squares SVM.

License GPL-2

Encoding UTF-8

Imports stats, graphics, svmpath(>= 0.970)

Suggests testthat

RoxygenNote 7.2.3

Roxygen list (markdown = TRUE, roclets = c (``namespace", ``rd",
``srr::srr_stats_roclet"))

R topics documented:

BostonHousing *Boston house price data The Boston Housing Dataset is a derived from information collected by the U.S. Census Service concerning housing in the area of Boston MA*

Description

- crim per capita crime rate by town
- zn proportion of residential land zoned for lots over 25,000 sq.ft.
- indus proportion of non-retail business acres per town
- chas Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
- nox nitric oxides concentration (parts per 10 million)
- rm average number of rooms per dwelling
- age proportion of owner-occupied units built prior to 1940
- dis weighted distances to five Boston employment centres
- rad index of accessibility to radial highways
- tax full-value property-tax rate per $10,000
- ptratio pupil-teacher ratio by town
- b 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
- stat percent lower status of the population
- mdev Median value of owner-occupied homes in $1000's

Usage

```
data(BostonHousing)
```

Format

A data frame with 506 rows and 14 variables.

Author(s)

Jungmin Shin, <jungminshin@korea.ac.kr>, Andreas Artemiou <artemiou@uol.ac.cy>, Seung Jun Shin, <sjshin@korea.ac.kr>

References

Harrison, D. and Rubinfeld, D.L. (1978) *Hedonic prices and the demand for clean air, J. Environ. Economics & Management, vol.5, 81-102.*
Belsley, Kuh & Welsch. (1980) *Regression diagnostics, pp.244-261, Wiley.*

crBIC *Order estimation via BIC-type criterion*

Description

Estimation of a structural dimensionality. Choose the k which maximizes a Gn value

Usage

```
crBIC(obj, rho)
```

Arguments

obj	The psdr object
rho	parameter for BIC criterion

Value

Estimated BIC-type Gn scores for determining the optimal structural dimension will be returned

Author(s)

Jungmin Shin, <jungminshin@korea.ac.kr>, Andreas Artemiou <artemiou@uol.ac.cy>, Seung Jun Shin, <sjshin@korea.ac.kr>

References

Li, B., Artemiou, A. and Li, L. (2011) *Principal support vector machines for linear and nonlinear sufficient dimension reduction, Annals of Statistics 39(6): 3182–3210.*

See Also

psdr plot.Gn

Examples

```
set.seed(1234)
n <- 200; p <- 5; H <- 10; lambda <- 0.001; eps <- 1.0e-6;
max.iter <- 10; h <- 1.0e-5; delta <- 2*1.0e-1;
init.theta <- rnorm(sd=1,n=p)
x <- matrix(rnorm(n*p, 0, 1), n, p)
err <- rnorm(n, 0, .2)
B <- matrix(0, p, 2)
B[1,1] <- 1; B[2,2] <- 1
x1 <- x %*% B[,1]
x2 <- x %*% B[,2]
fx <-  x1/(0.5 + (x2 + 1)^2)
y <- fx + err
obj <- psdr(x, y, init.theta, H,lambda, delta, h, eps, max.iter, loss="svm")
d.hat <- crBIC(obj, rho=0.005); d.hat
```

get.psi *Kernel function generator*

Description

Generate a kernel function

Usage

```
get.psi(x, y, k = floor(length(y)/3))
```

Arguments

`x`	data matrix
`y`	either continuous or (+1,-1) typed binary response vector
`k`	number of basis functions for a kernel trick, floor(length(y)/3) is default

Value

generate the kernel function $\phi(\mathbf{x})$

`w`	the first d leading eigenvectors of the matrix $\mathbf{QKQ}$ in Li B. et. al., (2011)
`l`	the first d leading eigenvectors of the matrix $\mathbf{QKQ}$
`scaled.x`	scaled $\mathbf{x}$
`bw`	bandwidth of the kernel function
`k`	parameter for the kernel, `floor(length(n)/3)` is applied as a default

Author(s)

Jungmin Shin, <`jungminshin@korea.ac.kr`>, Andreas Artemiou <`artemiou@uol.ac.cy`>, Seung Jun Shin, <`sjshin@korea.ac.kr`>

See Also

`npsdr`, `phix`

npsdr *A unified Principal sufficient dimension reduction method via kernel trick*

Description

Principal Sufficient Dimension Reduction method

Usage

```
npsdr(
  x,
  y,
  H = NULL,
  h = NULL,
  lambda = NULL,
  delta = NULL,
  k = floor(length(y)/3),
  eps = 1e-04,
  max.iter = NULL,
  loss = NULL,
  a = NULL,
  c = NULL
)
```

Arguments

x	data matrix
y	either continuous or (+1,-1) typed binary response vector
H	the number of slices. default value is 10
h	very small interval for calculating numerical derivatives for a given arbitrary loss function
lambda	hyperparameter for the loss function. default value is 0.1
delta	learning rate for gradient descent method. default value is 0.1
k	number of basis functions for a kernel trick, floor(length(y)/3) is default
eps	threshold for stopping iteration with respect to the magnitude of derivative, default value is 1.0e-4
max.iter	maximum iteration number for the optimization process. default value is 30
loss	pre-specified loss functions are "svm", "logit","l2svm","wsvm", and etc., and user-defined loss function object also can be used formed by inside double (or single) quotation mark
a	the first hyperparameter for the LUM loss function
c	second hyperparameter for the LUM loss function

Value

An object with S3 class "npsdr". Details are listed below.

evectors	Eigenvectors of the Mn, the first d leading eigenvectors consists the basis of the central subspace
evalues	Eigenvalues of the Mn
obj.psi	result of function `get.psi` is returned

Author(s)

Jungmin Shin, <jungminshin@korea.ac.kr>, Andreas Artemiou <artemiou@uol.ac.cy>, Seung Jun Shin, <sjshin@korea.ac.kr>

References

Artemiou, A. and Dong, Y. (2016) *Sufficient dimension reduction via principal lq support vector machine, Electronic Journal of Statistics 10: 783–805.*
Artemiou, A., Dong, Y. and Shin, S. J. (2021) *Real-time sufficient dimension reduction through principal least squares support vector machines, Pattern Recognition 112: 107768.*
Kim, B. and Shin, S. J. (2019) *Principal weighted logistic regression for sufficient dimension reduction in binary classification, Journal of the Korean Statistical Society 48(2): 194–206.*
Li, B., Artemiou, A. and Li, L. (2011) *Principal support vector machines for linear and nonlinear sufficient dimension reduction, Annals of Statistics 39(6): 3182–3210.*
Soale, A.-N. and Dong, Y. (2022) *On sufficient dimension reduction via principal asymmetric least squares, Journal of Nonparametric Statistics 34(1): 77–94.*
Wang, C., Shin, S. J. and Wu, Y. (2018) *Principal quantile regression for sufficient dimension reduction with heteroscedasticity, Electronic Journal of Statistics 12(2): 2114–2140.*
Shin, S. J., Wu, Y., Zhang, H. H. and Liu, Y. (2017) *Principal weighted support vector machines for sufficient dimension reduction in binary classification, Biometrika 104(1): 67–81.*
Li, L. (2007) *Sparse sufficient dimension reduction, Biometrika 94(3): 603–613.*

See Also

`npsdrx`, `phix`, `get.psi`, `print`

Examples

```
set.seed(1)
n <- 200;
p <- 5;
H <- 20;
lambda <- 0.1
eps <- 1.0e-5
max.iter <- 30
h <- 1.0e-6; delta <- 5*1.0e-1
x <- matrix(rnorm(n*p, 0, 2), n, p)
err <- rnorm(n, 0, .2)
B <- matrix(0, p, 2)
B[1,1] <- 1; B[2,2] <- 1
x1 <- x %*% B[,1]
x2 <- x %*% B[,2]
fx <-  x1/(0.5 + (x2 + 1)^2)
y <- c(fx + err) # response
y.binary <- sign(y)
npsdr(x, y, H, h, lambda, delta, k=floor(length(y)/3), eps,
               max.iter, loss="svm")

##real data: Wisconsin diagnostic breast cancer data
data(wisc)
x.wisc <- matrix(unlist(wisc[,-c(1,2)]), ncol = 30)
y.wisc <- 2*as.numeric(as.factor(unlist(wisc[,2]))) - 3
set.seed(123)
my.obj <- npsdr(x.wisc,y.wisc,H=20,lambda=0.01,delta=0.5, k=floor(length(y.wisc)/3),
               eps, max.iter=30,loss="wlogit")
x.nsvm <- phix(x.wisc, my.obj)
boxplot.default(x.nsvm[y.wisc == 1,1], x.nsvm[y.wisc != 1,1], xlab = "Y", axes = FALSE,
               ylab = expression(hat(phi)[1](x)))
axis(1, seq(0.5, 2.5, by = 0.5), c(NA, "+1", NA, "-1", NA)); axis(2, las = 1)
```

npsdrx *Predict a response variable for the new explanatory variables*

Description

Returning new $\mathbf{X}$ via the estimated nonlinear kernel.

Usage

```
npsdrx(object, newdata, d = 2)
```

Arguments

`object`	The object from function `npsdr`
`newdata`	new data $\mathbf{X}$
`d`	structural dimensionality. d=2 is default.

Value

the value of the estimated nonlinear mapping $\phi(\cdot)$ is applied to newdata X with dimension d is returned.

Author(s)

Jungmin Shin, <jungminshin@korea.ac.kr>, Andreas Artemiou <artemiou@uol.ac.cy>, Seung Jun Shin, <sjshin@korea.ac.kr>

See Also

npsdr

Examples

```
set.seed(1)
n <- 200;
p <- 5;
H <- 20;
lambda <- 0.1
eps <- 1.0e-5
max.iter <- 10
h <- 1.0e-6; delta <- 5*1.0e-1
x <- matrix(rnorm(n*p, 0, 2), n, p)
err <- rnorm(n, 0, .2)
B <- matrix(0, p, 2)
B[1,1] <- 1; B[2,2] <- 1
x1 <- x %*% B[,1]
x2 <- x %*% B[,2]
fx <-  x1/(0.5 + (x2 + 1)^2)
y <- c(fx + err) # response
y.binary <- sign(y)
new.x <- matrix(rnorm(n*p, 0, 2), n, p)
obj <- npsdr(x, y, H, h, lambda, delta, k=floor(length(y)/3), eps, max.iter, loss="svm")
npsdrx(object=obj, newdata=new.x, d = 2)
```

phix — *A function of yielding nonlinear mapping of $\mathbf{x}$ onto the reduced dimension*

Description

A function of yielding nonlinear mapping of $\mathbf{x}$ onto the reduced dimension

Usage

```
phix(value, object, d = 2)
```

Arguments

value	data matrix $\mathbf{x}$
object	the object from function 'npsdr'
d	the structural dimensionality. The default is $d = 2$

Value

An object with S3 class "npsdr". Details are listed below.Return the value of a nonlinear Mapping $\mathbf{x}$ to the reduced dimension of rank 2 via the kernel function $\phi(\mathbf{x})$

w	the first d leading eigenvectors of the matrix $\mathbf{QKQ}$ in Li B. et. al., (2011)
l	the first d leading eigenvectors of the matrix $\mathbf{QKQ}$
scaled.x	scaled $\mathbf{x}$
bw	bandwidth of the kernel function
k	parameter for the kernel, `floor(length(n)/3)` is applied as a default

Author(s)

Jungmin Shin, <jungminshin@korea.ac.kr>, Andreas Artemiou <artemiou@uol.ac.cy>, Seung Jun Shin, <sjshin@korea.ac.kr>

References

Li, B., Artemiou, A. and Li, L. (2011) *Principal support vector machines for linear and nonlinear sufficient dimension reduction, Annals of Statistics 39(6): 3182–3210.*

See Also

npsdr

Examples

```
set.seed(1)
n <- 200;
p <- 5;
H <- 20;
lambda <- 0.1
eps <- 1.0e-5
max.iter <- 10
h <- 1.0e-6; delta <- 5*1.0e-1
x <- matrix(rnorm(n*p, 0, 2), n, p)
err <- rnorm(n, 0, .2)
B <- matrix(0, p, 2)
B[1,1] <- 1; B[2,2] <- 1
x1 <- x %*% B[,1]
x2 <- x %*% B[,2]
fx <-  x1/(0.5 + (x2 + 1)^2)
y <- c(fx + err) # response
y.binary <- sign(y)
obj <- npsdr(x, y, H, h, lambda, delta, k=floor(length(y)/3), eps,
              max.iter, loss="svm")
phix(value=x, object=obj, d=2)
```

plot.Gn *plot Gn scores from a "Gn" object*

Description

Produces a Gn score line plot making easy to verify the maximum.

Usage

```
## S3 method for class 'Gn'
plot(x, ...)
```

Arguments

x	object from crBIC
...	Other graphical parameters to plot

Value

Estimated Gn scores for a optimal structural dimension will be returned

Author(s)

Jungmin Shin, <jungminshin@korea.ac.kr>, Andreas Artemiou <artemiou@uol.ac.cy>, Seung Jun Shin, <sjshin@korea.ac.kr>

References

Li, B., Artemiou, A. and Li, L. (2011) *Principal support vector machines for linear and nonlinear sufficient dimension reduction, Annals of Statistics 39(6): 3182–3210.*

See Also

psdr crBIC

Examples

```
set.seed(1234)
n <- 200; p <- 5; H <- 10; lambda <- 0.001; eps <- 1.0e-6;
max.iter <- 10; h <- 1.0e-5; delta <- 2*1.0e-1;
init.theta <- rnorm(sd=1,n=p)
x <- matrix(rnorm(n*p, 0, 1), n, p)
err <- rnorm(n, 0, .2)
B <- matrix(0, p, 2)
B[1,1] <- 1; B[2,2] <- 1
x1 <- x %*% B[,1]
x2 <- x %*% B[,2]
fx <-  x1/(0.5 + (x2 + 1)^2)
y <- fx + err
obj <- psdr(x, y, init.theta, H,lambda, h, delta, eps, max.iter, loss="svm")
d.hat <- crBIC(obj, rho=0.005)
plot(d.hat)
```

psdr *Unified Principal sufficient dimension reduction methods*

Description

A unified and user-friendly R package for applying the principal sufficient dimension reduction methods for both linear and nonlinear ,and regression and classification context. The package has an extendable power by varying loss functions for the SVM, even for an user-defined arbitrary function, unless those are convex and differentiable everywhere over the support. Also, it provides a realtime sufficient dimension reduction update procedure using the principal least squares SVM.

Usage

```
psdr(
  x,
  y,
  init = NULL,
  H = NULL,
  lambda = NULL,
  delta = NULL,
  h = 1e-05,
  eps = 1e-05,
  max.iter = NULL,
  loss = NULL,
  a = NULL,
  c = NULL,
  stochastic = FALSE,
  plot = FALSE
)
```

Arguments

`x` input matrix, of dimension `nobs` `x` `nvars`; each row is an observation vector. Requirement: `nvars`>1; in other words, `x` should have 2 or more columns.

`y` response variable, either can be continuous variable or (+1,-1) coded binary response vector

`init` initial coefficient vector of which dimension matches the `nvar` of $\mathbf{X}$. If it is not specified, random vector from standard normal distribution is applied by default

`H` the number of slices and probabilities equally spaced in $(0, 1)$. default value is 10

`lambda` the cost parameter for the svm loss function. The default value is 0.1

`delta` learning rate for the gradient descent algorithm. The default value is 0.1

`h` very small interval for calculating numerical derivatives for a given arbitrary loss function. The default is 1.0e-5

`eps` the threshold for stopping iteration with respect to the magnitude of the change of the derivative. The default value is 1.0e-5

`max.iter` maximum iteration number for the optimization process. default value is 30

`loss` pre-specified loss functions are "svm", "logit","l2svm","wsvm", and etc., and user-defined loss function object also can be used formed by inside double (or single) quotation mark

`a` the first hyperparameter for the LUM loss function

`c` the second hyperparameter for the LUM loss function

`stochastic` If `TRUE` then the stochastic gradient descent algorithm will be implemented to optimize the loss function. The default is FALSE

`plot` If `TRUE` then it produces scatter plots of Y versus $\hat{B^{\top}}_j\mathbf{X}$. j can be specified by the user with $j = 2$ as a default. The default is FALSE

Details

Details on `loss` option:

The argument `loss` determines a specific loss function for SVM and the corresponding SDR method. For example, `loss="ls"` means that the user can do SDR with least square SVM. The package provides several pre-embeded functions. 1. for regression problem: `loss="svm"` is the hinge loss, `loss="logstic"` is the logistic loss, `loss="l2svm"` is the squared hinge loss, `loss="LUM"` is for the large margin unified loss, `loss="asymls"` is for asymmetric least square loss 2. Also the corresponding weighted loss functions are included, such as, `loss="wsvm"` , `loss="wlogistic"`, `loss="l2wsvm"`, `loss="wLUM"` and `loss="wls"`, which mean weighted hinge loss, weighted logistic loss, weighted squared hinge loss, weighted LUM loss and weighted least square loss, respectively.

Not only function `psdr` includes popular loss functions, but also, it is designed for working with user defined arbitrary convex loss function that is claimed through the argument `loss`. Two examples of the usage of user-defined losses are presented below (`m` represents a margin):

`myLogistic <- function(m,...){rslt <- log(1+exp(-m)) return(rslt)}`,

`myLS <- function(m,...){rslt <- (1-m)^2 return(rslt)}`.

Users can define their own loss function in advance, and apply those functions to `psdr` like `loss="myLogistic"` or `loss="myLS"`.

Value

An object with S3 class "psdr". Details are listed below.

`Mn`	The estimated working matrix, which is obtained by the cumulative outer product of the estimated parameters over H
`evalues`	Eigenvalues of the Mn
`evectors`	Eigenvectors of the Mn, the first leading d eigenvectors consists the basis of the central subspace

Author(s)

Jungmin Shin, <jungminshin@korea.ac.kr>, Andreas Artemiou <artemiou@uol.ac.cy>, Seung Jun Shin, <sjshin@korea.ac.kr>

References

Artemiou, A. and Dong, Y. (2016) *Sufficient dimension reduction via principal lq support vector machine, Electronic Journal of Statistics 10: 783–805.*
Artemiou, A., Dong, Y. and Shin, S. J. (2021) *Real-time sufficient dimension reduction through principal least squares support vector machines, Pattern Recognition 112: 107768.*
Kim, B. and Shin, S. J. (2019) *Principal weighted logistic regression for sufficient dimension reduction in binary classification, Journal of the Korean Statistical Society 48(2): 194–206.*
Li, B., Artemiou, A. and Li, L. (2011) *Principal support vector machines for linear and nonlinear sufficient dimension reduction, Annals of Statistics 39(6): 3182–3210.*
Soale, A.-N. and Dong, Y. (2022) *On sufficient dimension reduction via principal asymmetric least squares, Journal of Nonparametric Statistics 34(1): 77–94.*
Wang, C., Shin, S. J. and Wu, Y. (2018) *Principal quantile regression for sufficient dimension reduction with heteroscedasticity, Electronic Journal of Statistics 12(2): 2114–2140.*
Shin, S. J., Wu, Y., Zhang, H. H. and Liu, Y. (2017) *Principal weighted support vector machines for sufficient dimension reduction in binary classification, Biometrika 104(1): 67–81.*
Li, L. (2007) *Sparse sufficient dimension reduction, Biometrika 94(3): 603–613.*

See Also

`plot`, `print`, `crBIC`

Examples

```
set.seed(1)
n <- 200;
p <- 5;
H <- 10;
lambda <- 0.1
eps <- 1.0e-5
max.iter <- 30
init.theta <- rnorm(p,0,1)
h <- 1.0e-5; delta <- 0.5
x <- matrix(rnorm(n*p, 0, 2), n, p)
err <- rnorm(n, 0, .2)
B <- matrix(0, p, 2)
B[1,1] <- 1; B[2,2] <- 1
x1 <- x %*% B[,1]
x2 <- x %*% B[,2]
fx <-  x1/(0.5 + (x2 + 1)^2)
```

```
y <- c(fx + err) # response
my.hinge <- function(m,...){
  rslt <- (1-m)*(as.numeric((1-m) > 0))
  return(rslt)
}
obj <- psdr(x, y, init.theta, H, lambda, delta, h, eps, max.iter, loss="svm")
psdr(x, y, init.theta, H,lambda, delta, h, eps, max.iter, loss="my.hinge")
print(obj)
plot(obj)

##real data: Boston housing data
data("BostonHousing")
attach(BostonHousing)
BostonHousing <- BostonHousing[BostonHousing$crim < 3.2 , -c(4,9)]
X <- BostonHousing[,-12]
Y <- BostonHousing[,"medv"]
p <- ncol(X); H <- 20; lambda <- 0.1; eps <- 1.0e-5;
max.iter <- 100; h <- 1.0e-5; delta <- 2*1.0e-1;
set.seed(1); init.theta <- rnorm(sd=1,n=p)
rslt <- psdr(X, Y, init.theta, H,lambda, h, delta, eps, max.iter, loss="svm")
value.lsvm <- rslt$values
lsvm <- round(rslt$vectors,3)
X <- as.matrix(X)
x.lsvm <- X %*% lsvm
plot(x.lsvm[,1],Y , type = "p", xlab = expression(hat(b)[1]^T*X), ylab="medv", cex=1)
lines(lowess( x.lsvm[,1], Y), col="red", lwd=2)
plot(x.lsvm[,2], Y, type = "p", xlab = expression(hat(b)[2]^T*X), ylab="medv", cex=1);
lines(lowess(x.lsvm[,2], Y), col="blue", lwd=2)

##real data: Wisconsin diagnostic breast cancer data
data(wisc)
x.wisc <- matrix(unlist(wisc[,-c(1,2)]), ncol = 30)
y.wisc <- 2*as.numeric(as.factor(unlist(wisc[,2]))) - 3
init.theta <- rnorm(dim(x.wisc)[2],0,1)
wisc.obj <- psdr(x.wisc, y.wisc, init.theta, H=20,lambda=0.1, h=1.0e-6,
                 delta=0.5,eps=10^-4, max.iter=30, loss="wlogit")
value.lsvm <- wisc.obj$values
lsvm <- round(wisc.obj$vectors,3)
x.lsvm <- x.wisc %*% lsvm
par(mar=c(5,5,5,5), oma=c(1,1,1,1))
plot(x.lsvm[,1], x.lsvm[,2], type = "n", xlab = expression(hat(b)[1]^T*X), ylab = expression(hat(b)[2]^T*X))
points(x.lsvm[y.wisc == 1,1], x.lsvm[y.wisc == 1,2], col = 4, pch = "+")
points(x.lsvm[y.wisc != 1,1], x.lsvm[y.wisc != 1,2], col = 2)
```

rtpsdr *Real time sufficient dimension reduction through principal least squares regression*

Description

In stream data, where we need to constantly update the estimation as new data are collected, the use of all available data can create computational challenges even for computationally efficient algorithms. Therefore it is important to develop real time SDR algorithms that work efficiently in the case that there are data streams. After getting an initial estimator with the currently available data, the basic idea of real-time method is to update the estimator efficiently as new data are collected. This function realizes real time least squares SVM SDR method for a both regression and classification problem It is efficient algorithms for either adding new data or removing old data are provided.

Usage

```
rtpsdr(A, r, n, Xbar, x, y, direction = "forward", H = NULL, lambda = NULL)
```

Arguments

`A` a temporal matrix produced by the previous estimation with original data

`r` an estimated parameters from the previous estimation with original data

`n` a number of sample size of the original data

`Xbar` a mean vector from the original data

`x` x in new data

`y` y in new data, y is continuous

`direction` A direction of change of the new data which is either "forward" or "backward". default is "forward"

`H` a number of slices. default is set to 10.

`lambda` hyperparameter for the loss function. default is set to 0.1.

Value

An object with S3 class "rtpsdr". Details are listed below.

`x` input data matrix

`y` iniput response vector

`Mn` The estimated working matrix, which is obtained by the cumulative outer product of the estimated parameters over H

`evalues` Eigenvalues of the Mn

`evectors` Eigenvectors of the Mn, the first d leading eigenvectors consists the basis of the central subspace

`N` total number of observation $n_1 + n_2$

`Xbar` mean of total $\mathbf{x}$

`r` updated estimated coefficients matrix

`A` new A part for update. See Artemiou et. al., (2021)

Author(s)

Jungmin Shin, <`jungminshin@korea.ac.kr`>, Andreas Artemiou <`artemiou@uol.ac.cy`>, Seung Jun Shin, <`sjshin@korea.ac.kr`>

References

Artemiou, A. and Dong, Y. (2016) *Sufficient dimension reduction via principal lq support vector machine, Electronic Journal of Statistics 10: 783–805.*

Artemiou, A., Dong, Y. and Shin, S. J. (2021) *Real-time sufficient dimension reduction through principal least squares support vector machines, Pattern Recognition 112: 107768.*

Kim, B. and Shin, S. J. (2019) *Principal weighted logistic regression for sufficient dimension reduction in binary classification, Journal of the Korean Statistical Society 48(2): 194–206.*

Li, B., Artemiou, A. and Li, L. (2011) *Principal support vector machines for linear and nonlinear sufficient dimension reduction, Annals of Statistics 39(6): 3182–3210.*

Soale, A.-N. and Dong, Y. (2022) *On sufficient dimension reduction via principal asymmetric least squares, Journal of Nonparametric Statistics 34(1): 77–94.*

Wang, C., Shin, S. J. and Wu, Y. (2018) *Principal quantile regression for sufficient dimension reduction with heteroscedasticity, Electronic Journal of Statistics 12(2): 2114–2140.*

Shin, S. J., Wu, Y., Zhang, H. H. and Liu, Y. (2017) *Principal weighted support vector machines for sufficient dimension reduction in binary classification, Biometrika 104(1): 67–81.*

Li, L. (2007) *Sparse sufficient dimension reduction, Biometrika 94(3): 603–613.*

See Also

psdr npsdr

Examples

```
set.seed(1234)
n <- 500; n2 <- 300; p <- 5; H <- 10; lambda <- 0.1;
x <- matrix(rnorm(n*p, 0, 1), n, p)
x.new <- matrix(rnorm(n2*p, 0, 1), n2, p)
err <- rnorm(n, 0, .2); err.new <- rnorm(n2, 0, .2)
B <- matrix(0, p, 2)
B[1,1] <- 1; B[2,2] <- 1
x1 <- x %*% B[,1]
x2 <- x %*% B[,2]
x1.new <- x.new %*% B[,1]
x2.new <- x.new %*% B[,2]
fx <-  x1/(0.5 + (x2 + 1)^2)
fx.new <-  x1.new/(0.5 + (x2.new + 1)^2)
y <- fx + err;                     y.new <- fx.new + err.new
y.binary <- c(sign(fx + err));     y.binary.new <- c(sign(fx.new + err.new))
#real time least squares forward
ls <- psdr(x, y, H=10, lambda=0.1, loss="lssvm")
ls$vectors
rt_ls <- rtpsdr(A=ls$A, r=ls$r, n=ls$N, Xbar=ls$Xbar, x=x.new, y=y.new,
        direction="forward", H, lambda)
rt_ls$vectors
#real time weighted least squares backward
wls <- psdr(x, y.binary, H=10, lambda=0.1, loss="wlssvm")
wls$vectors
rt_wls <- rtpsdr(A=wls$A, r=wls$r, n=wls$N, Xbar=wls$Xbar, x=x.new, y=y.binary.new,
               direction="backward", H, lambda)
rt_wls$vectors
```

wisc *Breast Cancer Wisconsin (Diagnostic) Data Set*

Description

Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image. Ten real-valued features are computed for each cell nucleus:

Usage

```
data(wisc)
```

Format

A data frame with 569 rows and 32 variables

Details

- ID number
- Diagnosis (M = malignant, B = benign)
- radius (mean of distances from center to points on the perimeter)
- texture (standard deviation of gray-scale values)
- perimeter
- area
- smoothness (local variation in radius lengths)
- compactness (perimeter^2 / area - 1.0)
- concavity (severity of concave portions of the contour)
- concave points (number of concave portions of the contour)
- symmetry
- fractal dimension ("coastline approximation" - 1)

Index