

Package ‘psdr’

April 11, 2024

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 version of principal sufficient dimension reduction methods for both linear and nonlinear cases.

License GPL-2

Encoding UTF-8

Roxygen list(markdown = TRUE)

Imports stats, graphics, svmpath(>= 0.970)

Suggests testthat

RoxygenNote 7.2.3

R topics documented:

BostonHousing	1
dimension	2
new.y	3
npsdr	4
psdr	6
rtpsdr	7
wisc	9
Index	10

BostonHousing	<i>Boston house price data The Boston house-price data of Harrison, D. and Rubinfeld, D.L. 'Hedonic prices and the demand for clean air', J. Environ. Economics & Management, vol.5, 81-102, 1978. Used in Belsley, Kuh & Welsch, 'Regression diagnostics, Wiley, 1980. N.B. Various transformations are used in the table on pages 244-261 of the latter.</i>
---------------	--

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 $1000(B_k - 0.63)^2$ where B_k 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

dimension	<i>Determination of structural dimension</i>
-----------	--

Description

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

Usage

```
dimension(obj, rho)
```

Arguments

obj	The psdr object
rho	hyperparameter of CVBIC

Value

Estimated G_n scores for a optimal structural dimension will be returned

Author(s)

Jungmin Shin, <jungminshin@korea.ac.kr>, Seung Jun Shin, <sjshin@korea.ac.kr>

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, h, delta, eps, max.iter, loss="svm")
d.hat <- dimension(obj, rho=0.005); d.hat
plot(d.hat)

```

new.y

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

Predict a response after reducing a input data dimension when the response and the input variable have a nonlinear functional relationship.

Usage

```
new.y(x, new.x, d = 2, ...)
```

Arguments

x	The object from function npsdr
new.x	new data X
d	true dimension for data x. d=2 is default
...	additional arguments affecting the predictions produced.

Value

Estimated response variable y return

Author(s)

Jungmin Shin, <jungminshin@korea.ac.kr>, SeungJun 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")
new.y(x=obj, new.x, d = 2)

```

npsdr	<i>#'A unified Principal sufficient dimension reduction method via kernel trick</i>
-------	---

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-05,
  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

<code>h</code>	very small interval for calculating numerical derivatives for a given arbitrary loss function
<code>lambda</code>	hyperparameter for the loss function. default value is 0.1
<code>delta</code>	learning rate for gradient descent method. default value is 0.1
<code>k</code>	number of basis functions for a kernel trick, $\text{floor}(\text{length}(y)/3)$ is default
<code>eps</code>	threshold for stopping iteration with respect to the magnitude of derivative, default value is $1.0\text{e-}5$
<code>max.iter</code>	maximum iteration number for the optimization process. default value is 30
<code>loss</code>	pre-specified loss functions are "logistic", "svm", "l2svm", "lwpsvm", and user-defined loss function object also can be used formed by inside double quotation mark
<code>a</code>	the first hyperparameter for the LUM loss function
<code>c</code>	second hyperparameter for the LUM loss function

Value

An estimated kernel matrix and its eigenvalues and eigenvectors for a sufficient dimension reduction will be returned

Author(s)

Jungmin Shin, <jungminshin@korea.ac.kr>, Seung Jun Shin, <sjshin@korea.ac.kr>

See Also

[new.y](#)

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)
npsdr(x, y, H, h, lambda, delta, k=floor(length(y)/3), eps,
      max.iter, loss="svm")
```

psdr	<i>#Unified Principal sufficient dimension reduction methods</i>
------	--

Description

Principal Sufficient Dimension Reduction methods

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	data matrix
y	either continuous or (+1,-1) typed binary response vector
init	initial coefficient vector. If not specified, random vector from standard normal distribution is applied by default
H	the number of slices. default value is 10
lambda	hyperparameter for the loss function. default value is 0.1
delta	learning rate for gradient descent method. default value is 0.1
h	very small interval for calculating numerical derivatives for a given arbitrary loss function, default is 1.0e-5
eps	threshold for stopping iteration with respect to the magnitude of derivative, 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 "logistic", "svm", "l2svm", "lwpsvm", and user-defined loss function object also can be used formed by inside double quotation mark
a	the first hyperparameter for the LUM loss function
c	second hyperparameter for the LUM loss function
stochastic	specify whether the user want to use the stochastic gradient descent algorithm. default is FALSE
plot	visualize user defined loss function. default is FALSE

Value

An estimated basis for central subspace and its eigenvalues and eigenvectors for SDR is returned

Author(s)

Jungmin Shin, <jungminshin@korea.ac.kr>, Seung Jun Shin, <sjshin@korea.ac.kr>

See Also

plot.psdr

Examples

```
set.seed(1)
n <- 200;
p <- 5;
H <- 10;
lambda <- 0.1
eps <- 1.0e-5
max.iter <- 10
init.theta <- rnorm(p,0,1)
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
my.hinge <- function(m,...){
  rslt <- (1-m)*(as.numeric((1-m) > 0))
  return(rslt)
}
obj <- psdr(x, y, init.theta, H,lambda, h, delta, eps, max.iter, loss="svm")
psdr(x, y, init.theta, H,lambda, h, delta, eps, max.iter, loss="my.hinge")
print(obj)
plot(obj)
```

rtpsdr

Real time sufficient dimension reduction through principal least squares SVM

Description

Real time least squares SVM SDR method for a both regression and classification problem

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 updated psdr result will be returned in a real time fashion

Author(s)

Jungmin Shin, <jungminshin@korea.ac.kr>, Seung Jun Shin, <sjshin@korea.ac.kr>

See Also

[psdr](#)

Examples

```
set.seed(1234)
n <- 300; n2 <- 500; 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))
ls <- psdr(x, y, H=10, lambda=0.1, loss="ls")
wls <- psdr(x, y.binary, H=10, lambda=0.1, loss="wls")
rtpsdr(A=ls$A, r=ls$r, n=ls$N, Xbar=ls$Xbar, X=x.new, y=y.new, direction="forward", H, lambda)
rtpsdr(A=wls$A, r=wls$r, n=wls$N, Xbar=wls$Xbar, X=x.new, y=y.binary.new,
direction="forward", H, lambda)
```

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 ($\text{perimeter}^2 / \text{area} - 1.0$)
- concavity (severity of concave portions of the contour)
- concave points (number of concave portions of the contour)
- symmetry
- fractal dimension ("coastline approximation" - 1)

Index

* **datasets**

BostonHousing, [1](#)
wisc, [9](#)

BostonHousing, [1](#)

dimension, [2](#)

new.y, [3](#), [5](#)

npsdr, [3](#), [4](#)

psdr, [3](#), [6](#), [8](#)

rtpsdr, [7](#)

wisc, [9](#)