

# Package ‘psvmSDR’

October 27, 2025

**Type** Package

**Title** Unified Principal Sufficient Dimension Reduction Package

**Version** 2.0.1

**Maintainer** Jungmin Shin <c16267@gmail.com>

**Description** A unified and user-friendly framework for applying the principal sufficient dimension reduction methods for both linear and nonlinear cases. The package has an extendable power by varying loss functions for the support vector machine, even for an user-defined arbitrary function, unless those are convex and differentiable everywhere over the support (Li et al. (2011) <[doi:10.1214/11-AOS932](https://doi.org/10.1214/11-AOS932)>). Also, it provides a real-time sufficient dimension reduction update procedure using the principal least squares support vector machine (Artemiou et al. (2021) <[doi:10.1016/j.patcog.2020.107768](https://doi.org/10.1016/j.patcog.2020.107768)>).

**License** GPL-2

**Encoding** UTF-8

**Imports** stats, graphics

**Suggests** testthat (>= 3.0.0)

**RoxygenNote** 7.3.3

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Jungmin Shin [aut, cre],  
Seung Jun Shin [aut],  
Andreas Artemiou [aut]

**Repository** CRAN

**Date/Publication** 2025-10-27 20:00:14 UTC

## Contents

|                      |    |
|----------------------|----|
| npsdr . . . . .      | 2  |
| npsdr_x . . . . .    | 4  |
| plot.npsdr . . . . . | 5  |
| plot.psdr . . . . .  | 6  |
| psdr . . . . .       | 7  |
| psdr_bic . . . . .   | 9  |
| rtpsdr . . . . .     | 10 |

**Index****13**


---

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

---

**Description**

Principal Sufficient Dimension Reduction method

**Usage**

```
npsdr(
  x,
  y,
  loss = "svm",
  h = 10,
  lambda = 1,
  b = floor(length(y)/3),
  eps = 1e-05,
  max.iter = 100,
  eta = 0.1,
  mtype,
  plot = TRUE
)
```

**Arguments**

|          |                                                                                                                                                                                                                                                            |
|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| x        | data matrix                                                                                                                                                                                                                                                |
| y        | either continuous or (+1,-1) typed binary response vector                                                                                                                                                                                                  |
| loss     | pre-specified loss functions belongs to "svm", "logit", "l2svm", "wsvm", "qr", "asls", "wlogit", "wl2svm", "lssvm", "wlssvm", and user-defined loss function object also can be used formed by inside double (or single) quotation mark. Default is 'svm'. |
| h        | the number of slices. default value is 10                                                                                                                                                                                                                  |
| lambda   | hyperparameter for the loss function. default value is 1                                                                                                                                                                                                   |
| b        | 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                                                                                                                                                                                 |
| eta      | learning rate for gradient descent method. default value is 0.1                                                                                                                                                                                            |
| mtype    | type of margin, either "m" or "r" refer margin and residual, respectively (See, Table 1 in the package manuscript). When one use user-defined loss function this argument should be specified. Default is "m".                                             |
| plot     | If TRUE then it produces scatter plots of Y versus the first sufficient predictor. The default is FALSE.                                                                                                                                                   |

**Value**

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

|         |                                                                                                                                   |
|---------|-----------------------------------------------------------------------------------------------------------------------------------|
| evalues | Eigenvalues of the estimated working matrix $M$ .                                                                                 |
| evecors | Eigenvectors of the estimated working matrix $M$ , the first $d$ leading eigenvectors consists the basis of the central subspace. |

**Author(s)**

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

**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**

[npsdr\\_x](#), [psdr](#), [rtpsdr](#)

**Examples**

```
set.seed(1)
n <- 200;
p <- 5;
x <- matrix(rnorm(n*p, 0, 2), n, p)
y <- 0.5*sqrt((x[,1]^2+x[,2]^2))*(log(x[,1]^2+x[,2]^2))+ 0.2*rnorm(n)
obj_kernel <- npsdr(x, y, plot=FALSE)
print(obj_kernel)
plot(obj_kernel)
```

npsdr\_x

*Reconstruct the estimated sufficient predictors for a given data matrix***Description**

Returning the estimated sufficient predictors  $\hat{\phi}(\mathbf{x})$  for a given  $\mathbf{x}$

**Usage**

```
npsdr_x(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  $\mathbf{X}$  with dimension d is returned.

**Author(s)**

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

**See Also**

[npsdr](#)

**Examples**

```
set.seed(1)
n <- 200; n.new <- 300
p <- 5;
h <- 20;
x <- matrix(rnorm(n*p, 0, 2), n, p)
y <- 0.5*sqrt((x[,1]^2+x[,2]^2))*(log(x[,1]^2+x[,2]^2))+ 0.2*rnorm(n)
new.x <- matrix(rnorm(n.new*p, 0, 2), n.new, p)
obj_kernel <- npsdr(x, y)
npsdr_x(object=obj_kernel, newdata=new.x)
```

---

plot.npsdr*Scatter plot with sufficient predictors from npsdr() function*

---

**Description**

Scatter plot with sufficient predictors from npsdr() function

**Usage**

```
## S3 method for class 'npsdr'  
plot(x, ..., d = 1, lowess = TRUE)
```

**Arguments**

|        |                                                             |
|--------|-------------------------------------------------------------|
| x      | object from the function npsdr()                            |
| ...    | Additional arguments to be passed to generic plot function. |
| d      | number of sufficient predictors. Default is 1.              |
| lowess | draw a lowess curve. Default is TRUE.                       |

**Value**

A scatter plot with sufficient predictors.

**Author(s)**

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

**See Also**

[npsdr\\_x](#), [npsdr](#)

**Examples**

```
set.seed(1)  
n <- 200;  
p <- 5;  
x <- matrix(rnorm(n*p, 0, 2), n, p)  
y <- x[,1]/(0.5 + (x[,2] + 1)^2) + 0.2*rnorm(n)  
obj_kernel <- npsdr(x, y, plot=FALSE)  
plot(obj_kernel)
```

---

`plot.psd`*Scatter plot with sufficient predictors from psdr() function*

---

**Description**

Scatter plot with sufficient predictors from psdr() function

**Usage**

```
## S3 method for class 'psdr'  
plot(x, ..., d = 1, lowess = TRUE)
```

**Arguments**

|                     |                                                                       |
|---------------------|-----------------------------------------------------------------------|
| <code>x</code>      | object from the function psdr()                                       |
| <code>...</code>    | Additional arguments to be passed to generic plot function.           |
| <code>d</code>      | number of sufficient predictors. Default is 1.                        |
| <code>lowess</code> | draw a locally weighted scatterplot smoothing curve. Default is TRUE. |

**Value**

A scatter plot with sufficient predictors.

**Author(s)**

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

**See Also**

[psdr\\_bic](#), [psdr](#)

**Examples**

```
set.seed(1)  
n <- 200; p <- 5;  
x <- matrix(rnorm(n*p, 0, 2), n, p)  
y <- x[,1]/(0.5 + (x[,2] + 1)^2) + 0.2*rnorm(n)  
obj <- psdr(x, y)  
plot(obj, d=2, lowess=TRUE)
```

## Description

A function for a linear principal sufficient dimension reduction.

## Usage

```
psdr(
  x,
  y,
  loss = "svm",
  h = 10,
  lambda = 1,
  eps = 1e-05,
  max.iter = 100,
  eta = 0.1,
  mtype = "m",
  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.                                                                                                                                                              |
| loss     | pre-specified loss functions belongs to "svm", "logit", "l2svm", "wsvm", "qr", "asls", "wlogit", "wl2svm", "lssvm", "wlssvm", and user-defined loss function object also can be used formed by inside double (or single) quotation mark. Default is 'svm'. |
| 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 1.                                                                                                                                                                                      |
| 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 100.                                                                                                                                                                               |
| eta      | learning rate for the gradient descent algorithm. The default value is 0.1.                                                                                                                                                                                |
| mtype    | a margin type, which is either margin ("m") or residual ("r") (See, Table 1 in the manuscript). Only need when user-defined loss is used. Default is "m".                                                                                                  |
| plot     | If TRUE then it produces scatter plots of $Y$ versus $\hat{B}_j^\top \mathbf{X}$ . $j$ can be specified by the user with $j = 1$ as a default. The default is FALSE.                                                                                       |

## Details

Two examples of the usage of user-defined losses are presented below ( $u$  represents a margin):

```
mylogit <- function(u, ...) log(1+exp(-u)),
myls <- function(u ...) u^2.
```

Argument  $u$  is a function variable (any character is possible) and the argument `mtype` for `psdr()` determines a type of a margin, either (`type="m"`) or (`type="r"`) method. `type="m"` is a default. Users have to change `type="r"`, when applying residual type loss. Any additional parameters of the loss can be specified via `...` argument.

## Value

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

|                       |                                                                                                                                                                                   |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>Mn</code>       | The estimated working matrix, which is obtained by the cumulative outer product of the estimated parameters over the slices. It will not print out, unless it is called manually. |
| <code>evalues</code>  | Eigenvalues of the working matrix $Mn$                                                                                                                                            |
| <code>evectors</code> | Eigenvectors of the $Mn$ , the first leading $d$ eigenvectors consists the basis of the central subspace                                                                          |

## Author(s)

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

## 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\\_bic](#), [rtpsdr](#)

**Examples**

```
## -----
## Linear PM
## -----
set.seed(1)
n <- 200; p <- 5;
x <- matrix(rnorm(n*p, 0, 2), n, p)
y <- x[,1]/(0.5 + (x[,2] + 1)^2) + 0.2*rnorm(n)
y.tilde <- sign(y)
obj <- psdr(x, y)
print(obj)
plot(obj, d=2)

## -----
## Linear PM (Binary classification)
## -----
obj_wsvm <- psdr(x, y.tilde, loss="wsvm")
plot(obj_wsvm)

## -----
## User-defined loss function
## -----
mylogistic <- function(u) log(1+exp(-u))
psdr(x, y, loss="mylogistic")
```

psdr\_bic

*Order estimation via BIC-type criterion***Description**

Estimation of a structural dimensionality. Choose the  $k$  which maximizes a BIC (Bayesian information criterion) value.

**Usage**

```
psdr_bic(obj, rho = 0.01, plot = TRUE, ...)
```

**Arguments**

|      |                                                             |
|------|-------------------------------------------------------------|
| obj  | The psdr object                                             |
| rho  | Parameter for BIC criterion. Default is 0.01.               |
| plot | Boolean. If TRUE, the plot of BIC values are depicted.      |
| ...  | Additional arguments to be passed to generic plot function. |

**Value**

Estimated BIC scores for determining the optimal structural dimension will be returned with plot.

**Author(s)**

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

**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](#)

**Examples**

```
set.seed(1234)
n <- 200; p <- 10;
x <- matrix(rnorm(n*p, 0, 1), n, p)
y <- x[,1]/(0.5 + (x[,2] + 1)^2) + rnorm(n, 0, .2)
obj <- rtpsdr(x, y, loss="svm")
d.hat <- rtpsdr_bic(obj)
print(d.hat)
```

---

rtpsdr

*Real time sufficient dimension reduction through principal least squares SVM*

---

**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(x, y, obj = NULL, h = 10, lambda = 1)
```

**Arguments**

|        |                                                            |
|--------|------------------------------------------------------------|
| x      | x in new data                                              |
| y      | y in new data, y is continuous                             |
| obj    | the latest output object from the rtpsdr                   |
| h      | a number of slices. default is set to 10.                  |
| lambda | hyperparameter for the loss function. default is set to 1. |

**Value**

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

|          |                                                                                                                    |
|----------|--------------------------------------------------------------------------------------------------------------------|
| x        | input data matrix                                                                                                  |
| y        | input 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 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>, Seung Jun Shin, <sjshin@korea.ac.kr>, Andreas Artemiou <artemiou@uol.ac.cy>

**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**

```
p <- 5
m <- 500 # batch size
N <- 10 # number of batches
obj <- NULL
for (iter in 1:N){
  set.seed(iter)
  x <- matrix(rnorm(m*p), m, p)
  y <- x[,1]/(0.5 + (x[,2] + 1)^2) + 0.2 * rnorm(m)
  obj <- rtpsdr(x = x, y = y, obj=obj)
}
print(obj)
```

# Index

npsdr, [2](#), [4](#), [5](#), [12](#)

npsdr\_x, [3](#), [4](#), [5](#)

plot.npsdr, [5](#)

plot.psd, [6](#)

psdr, [3](#), [6](#), [7](#), [10](#), [12](#)

psdr\_bic, [6](#), [8](#), [9](#)

rtpsdr, [3](#), [8](#), [10](#)