Package ‘ppmSDR’

February 16, 2026

Type Package

Title Penalized Principal Machine for Sufficient Dimension Reduction

Version 1.0.0

Description Provides a unified interface for computation-friendly penalized principal machine (P2PM) estimators for sufficient dimension reduction (SDR) in regression and classification. Includes state-of-the-art sparse SDR methods such as P2(W)LSM, P2(W)L2M, P2(W)LR, P2(W)SVM, P2QR, and P2AR.

Depends R (>= 4.1.0)

License GPL-3

Encoding UTF-8

Imports Matrix, grpreg, blockmatrix, Rfast, methods

Suggests knitr, MASS, rmarkdown, testthat (>= 3.0.0)

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

NeedsCompilation no

Author Jungmin Shin [aut, cre] (Department of Biomedical Informatics, The Ohio State University),
Seung Jun Shin [aut] (Department of Statistics, Korea University)

Maintainer Jungmin Shin <c16267@gmail.com>

Contents

ppasls *Penalized Principal Asymmetric Least Square Regression (P^2AR)*

Description

This function implements sparse sufficient dimension reduction using penalized principal asymmetric least squares regression (P^2AR), supporting group SCAD, MCP, and Lasso penalties.

Usage

```
ppasls(
  x,
  y,
  H = 10,
  C = 1,
  lambda = 0.001,
  gamma = 3.7,
  penalty = "grSCAD",
  max.iter = 100
)
```

Arguments

`x`	A numeric matrix of predictors (n x p), where n is the number of samples and p is the number of predictors.
`y`	A numeric response vector of length n.
`H`	Number of quantile levels (slices) used for asymmetric loss.
`C`	Penalty parameter for the loss function.
`lambda`	Regularization parameter controlling sparsity (default: SCAD).
`gamma`	Hyperparameter for SCAD or MCP penalty (default: 3.7 for group SCAD).
`penalty`	Type of group penalty. One of `"grSCAD"`, `"grLasso"`, or `"grMCP"`.
`max.iter`	Maximum number of iterations for convergence.

Details

The function estimates a sparse basis for the central subspace by solving the penalized principal asymmetric least squares problem. The penalty can be chosen from group SCAD, group Lasso, or group MCP.

For further details on the underlying method, see Soale et al. (2022) and Shin et al. (2024).

Value

A list with the following components:

evalues Eigenvalues of the estimated Mn matrix.

evectors Eigenvectors (columns) corresponding to the estimated central subspace directions.

x Original input predictor matrix.

References

Soale, B. B., Artemiou, A., & Li, B. (2022). Sufficient dimension reduction via principal asymmetric regression. *Statistica Sinica.*

Examples

```
## Not run:
  set.seed(1)
  n <- 100; p   <- 10
  B <- matrix(0, p, 2)
  B[1,1] <- B[2,2] <- 1
  x <- MASS::mvrnorm(n, rep(0, p), diag(1,p))
  y <- (x %*% B[,1]/(0.5 + (x %*% B[,2] + 1)^2)) + 0.2*rnorm(n, 0, 1)
  fit <- ppasls(x, y, H = 10, C = 0.5, lambda = 0.1, gamma = 3.7,
              penalty = "grSCAD", max.iter = 100)
  fit$evectors[,1:2]

## End(Not run)
```

`ppl2svm` *Penalized Principal L2-Hinge Support Vector Machine for Sparse Sufficient Dimension Reduction*

Description

This function implements the penalized principal L2-hinge support vector machine (P^2L2M) approach for sparse sufficient dimension reduction (SDR)

Usage

```
ppl2svm(
  x,
  y,
  H = 10,
  C = 1,
  lambda = 0.001,
  gamma = 3.7,
  penalty = "grSCAD",
  max.iter = 100
)
```

Arguments

`x`	A numeric matrix of predictors (n x p), where n is the number of samples and p is the number of predictors.
`y`	A numeric response vector of length n.
`H`	The number of slicing or quantiles (default: 10).
`C`	Regularization parameter (default: 1).
`lambda`	Penalty parameter controlling sparsity (default: 0.001).
`gamma`	Regularization parameter for SCAD/MCP penalty (default: 3.7).
`penalty`	Type of penalty: `"grSCAD"` (default), `"grLasso"`, or `"grMCP"`.
`max.iter`	Maximum number of iterations for the algorithm (default: 100).

Details

The method estimates the sparse basis of the central subspace by minimizing a penalized principal L2-hinge SVM objective. The algorithm supports group penalties to encourage row-wise sparsity, and efficiently computes the solution even in high-dimensional settings.

Value

A list with the following components:

`evalues`	Eigenvalues of the estimated matrix spanning the central subspace.
`evectors`	Corresponding eigenvectors (columns) for the estimated sufficient directions.
`x`	The original predictor matrix.

Examples

```
## Not run:
  set.seed(1)
  n <- 100; p <- 10
  B <- matrix(0, p, 2)
  B[1,1] <- B[2,2] <- 1
  x <- MASS::mvrnorm(n, rep(0, p), diag(1, p))
  y <- (x %*% B[,1] / (0.5 + (x %*% B[,2] + 1)^2)) + 0.2 * rnorm(n)
  fit <- ppl2svm(x, y, H = 10, C = 1, lambda = 0.2, gamma = 3.7,
                  penalty = "grSCAD", max.iter = 100)
  fit$evectors[, 1:2]

## End(Not run)
```

`pplr` *Penalized Principal Logistic Regression for Sparse Sufficient Dimension Reduction*

Description

This function implements the penalized principal logistic regression (P^2LR) approach for sparse sufficient dimension reduction (SDR), as proposed in Shin et al. (2024).

Usage

```
pplr(
  x,
  y,
  H = 10,
  C = 1,
  lambda,
  gamma = 3.7,
  penalty = "grSCAD",
  max.iter = 100
)
```

Arguments

`x`	Numeric predictor matrix (n x p), where n is the sample size and p is the number of predictors.
`y`	Numeric response vector of length n.
`H`	The number of slicing quantiles (default: 10).
`C`	Regularization parameter (default: 1).
`lambda`	Penalty parameter for sparsity (default: 0.001).
`gamma`	Regularization parameter for SCAD/MCP penalty (default: 3.7).
`penalty`	Penalty type: `"grSCAD"` (default), `"grLasso"`, or `"grMCP"`.
`max.iter`	Maximum number of iterations for the algorithm (default: 100).

Details

This method estimates a sparse basis for the central subspace by minimizing a penalized principal logistic regression objective. Supports group penalties to induce row-wise sparsity and efficiently computes solutions even for high-dimensional data.

Value

A list containing:

`Mn`	The estimated kernel matrix for the central subspace.
`evalues`	Eigenvalues of the estimated central subspace matrix.
`evectors`	Eigenvectors (columns) corresponding to the estimated sufficient directions.
`x`	The original predictor matrix.

Examples

```
## Not run:
  set.seed(1)
  n <- 100; p <- 10
  B <- matrix(0, p, 2)
  B[1,1] <- B[2,2] <- 1
  x <- MASS::mvrnorm(n, rep(0, p), diag(1, p))
  y <- (x %*% B[,1] / (0.5 + (x %*% B[,2] + 1)^2)) + 0.2 * rnorm(n)
  fit <- pplr(x, y, H = 10, C = 1, lambda = 0.00005, gamma = 3.7,
              penalty = "grSCAD", max.iter = 100)
  fit$evectors[, 1:2]

## End(Not run)
```

`pplssvm` *Penalized Principal Least Squares Support Vector Machine (P^2LSSVM) for Sparse Sufficient Dimension Reduction*

Description

This function implements the penalized principal least squares support vector machine (P^2LSSVM) approach for sparse sufficient dimension reduction (SDR).

Usage

```
pplssvm(
  x,
  y,
  H = 10,
  C = 1,
  lambda,
  gamma = 3.7,
  penalty = "grSCAD",
  max.iter = 100
)
```

Arguments

`x`	Numeric predictor matrix (n x p), where n is the sample size and p is the number of predictors.
`y`	Numeric response vector of length n.
`H`	The number of slicing point (default: 10).
`C`	Regularization parameter (default: 1).
`lambda`	Penalty parameter for sparsity.
`gamma`	Regularization parameter for SCAD/MCP penalty (default: 3.7).
`penalty`	Penalty type: `"grSCAD"` (default), `"grLasso"`, or `"grMCP"`.
`max.iter`	Maximum number of iterations for the algorithm (default: 100).

Details

This method estimates a sparse basis for the central subspace by minimizing a penalized principal least squares SVM objective. It supports group penalties to induce row-wise sparsity and efficiently computes solutions even for high-dimensional data.

Value

A list containing:

`evalues`	Eigenvalues of the estimated central subspace matrix.
`evectors`	Eigenvectors (columns) corresponding to the estimated sufficient directions.
`x`	The original predictor matrix.

Examples

```
## Not run:
  set.seed(1)
  n <- 100; p <- 10
  B <- matrix(0, p, 2)
  B[1,1] <- B[2,2] <- 1
  x <- MASS::mvrnorm(n, rep(0, p), diag(1, p))
  y <- (x %*% B[,1] / (0.5 + (x %*% B[,2] + 1)^2)) + 0.2 * rnorm(n)
  fit <- pplssvm(x, y, H = 10, C = 1, lambda = 0.03, gamma = 3.7,
              penalty = "grSCAD", max.iter = 100)
  fit$evectors[, 1:2]

## End(Not run)
```

---

ppm *Penalized Principal Machine Wrapper Function*

Description

Provides a unified interface for fitting various penalized principal machine (P^{2}PM) estimators for sufficient dimension reduction (SDR) in regression and classification. The `ppm()` function calls the appropriate penalized SDR estimator based on the specified loss function, allowing users to select among squared loss, logistic, hinge, quantile, and their weighted versions for classification and regression.

Usage

```
ppm(
  x,
  y,
  H = 10,
  C = 1,
  loss = "lssvm",
  penalty = "grSCAD",
  lambda = 0.001,
  gamma = 3.7,
  max.iter = 100,
  ...
)
```

Arguments

x	Numeric matrix. The input data matrix (observations in rows, variables in columns).
y	Numeric or factor vector. The response variable.
H	Integer. Number of quantile slices or thresholds to use (default: 10).
C	Numeric. Regularization parameter for the loss (default: 1).
loss	Character. Specifies the loss type. One of "lssvm", "wlssvm", "svm", "wsvm", "l2svm", "wl2svm", "logit", "wlogit", "asls", or "qr".

`penalty`	Character. Penalty type for group selection. One of `"grSCAD"`, `"grLasso"`, or `"grMCP"` (default: "grSCAD").
`lambda`	Numeric. Regularization parameter for the penalty (default: 0.001).
`gamma`	Numeric. Penalty shape parameter for SCAD/MCP (default: 3.7).
`max.iter`	Integer. Maximum number of iterations (default: 100).
`...`	Additional arguments passed to the underlying method.

Details

This wrapper selects among ten state-of-the-art penalized SDR estimators, including penalized principal least squares SVM (P^{2}LSM), penalized principal SVM (P^{2}SVM), penalized principal L2-hinge SVM (P^{2}L2M), penalized principal quantile regression (P^{2}QR), penalized principal asymmetric least squares (P^{2}AR), penalized principal logistic regression (P^{2}LR), and their weighted counterparts for classification.

Each method is implemented in a dedicated function and uses an efficient optimization algorithm such as group coordinate descent (GCD) or majorization-minimization (MM) for computation-friendly and scalable estimation. The argument `loss` selects the corresponding method, and all relevant parameters are passed automatically.

Value

A list containing:

`evectors`	Estimated basis directions (principal SDR vectors) as columns.
`evalues`	Associated eigenvalues.
`Mn`	Estimated working matrix (if returned by the underlying method).
`x`	Original input matrix.

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.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties, *Journal of the American Statistical Association*, 96: 1348–1360.

Hunter, D. R. and Lange, K. (2004). A tutorial on MM algorithms, *The American Statistician*, 58(1): 30–37.

Jang, H. J., Shin, S. J. and Artemiou, A. (2023). Principal weighted least square support vector machine: An online dimension-reduction tool for binary classification, *Computational Statistics & Data Analysis*, 187: 107818.

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.

Shin, J. and Shin, S. J. (2024). A concise overview of principal support vector machines and its generalization, *Communications for Statistical Applications and Methods*, 31(2): 235–246.

Shin, J., Shin, S. J. and Artemiou, A. (2024). The R package psvmsdr: A unified algorithm for sufficient dimension reduction via principal machines, *arXiv preprint arXiv:2409.01547*.

Shin, S. J. and Artemiou, A. (2017). Penalized principal logistic regression for sparse sufficient dimension reduction, *Computational Statistics & Data Analysis*, 111: 48–58.

Examples

```
library(MASS)
set.seed(1)
n <- 1000
p   <- 10
B <- matrix(0, p, 2)
B[1,1] <- B[2,2] <- 1
x <- mvrnorm(n, rep(0, p), diag(1,p))
y <- (x %*% B[,1]/(0.5 + (x %*% B[,2] + 1)^2)) + 0.2*rnorm(n, 0, 1)
y.binary <- sign(y)

# SDR in Regression: lssvm
result1 <- ppm(x, y, H=10, C=1, loss="lssvm", penalty = 'grSCAD', lambda=0.003)
round(result1$evectors[,1:2], 5)

# SDR in Regression: svm
result2 <- ppm(x, y, H=10, C=1, loss="svm", penalty = 'grSCAD', lambda=0.0001)
round(result2$evectors[,1:2], 5)
# SDR in Classification: wlssvm
result3 <- ppm(x, y.binary, H=10, C=1, loss="wlssvm", penalty = 'grSCAD', lambda=0.0005)
round(result3$evectors[,1:2], 5)

# SDR in Classification: wlogit
result4 <- ppm(x, y.binary, H=10, C=20, loss="wlogit", penalty = 'grSCAD', lambda=0.07)
round(result4$evectors[,1:2], 5)
```

---

ppqr — *Penalized Principal Quantile Regression (P^2QR, MM-GCD-based) for Sparse Sufficient Dimension Reduction*

Description

This function implements the penalized principal quantile regression (P^2QR) approach for sparse sufficient dimension reduction (SDR).

Usage

```
ppqr(
  x,
  y,
  H = 10,
  C = 1,
  lambda,
  gamma = 3.7,
  penalty = "grSCAD",
  max.iter = 100
)
```

Arguments

`x`	Numeric predictor matrix (n x p), where n is the sample size and p is the number of predictors.
`y`	Numeric response vector of length n.
`H`	The number of quantile slices (default: 10).
`C`	Regularization parameter (default: 1).
`lambda`	Penalty parameter for sparsity.
`gamma`	Regularization parameter for SCAD/MCP penalty (default: 3.7).
`penalty`	Penalty type: `"grSCAD"` (default), `"grLasso"`, or `"grMCP"`.
`max.iter`	Maximum number of iterations for the algorithm (default: 100).

Details

This function estimates a sparse basis for the central subspace using a penalized principal quantile regression objective. The Majorization-Minimization-Group Coordinate Descent algorithm is used for optimization, providing both computational efficiency and reliable convergence.

Value

A list containing:

`Mn`	Estimated central subspace matrix.
`evalues`	Eigenvalues of the estimated matrix.
`evectors`	Eigenvectors (columns) corresponding to the estimated sufficient directions.
`x`	The original predictor matrix.

Examples

```
## Not run:
  set.seed(1)
  n <- 300; p <- 10
  B <- matrix(0, p, 2)
  B[1,1] <- B[2,2] <- 1
  x <- MASS::mvrnorm(n, rep(0, p), diag(1, p))
  y <- (x %*% B[,1] / (0.5 + (x %*% B[,2] + 1)^2)) + 0.2 * rnorm(n)
  fit <- ppqr(x, y, H = 10, C = 1, lambda = 0.0001, gamma = 3.7,
              penalty = "grSCAD", max.iter = 100)
  fit$evectors[, 1:2]

## End(Not run)
```

ppsvm *Penalized Principal Support Vector Machine (P^2SVM, MM-GCD-based) for Sparse Sufficient Dimension Reduction*

Description

This function implements the penalized principal support vector machine (P^2SVM) approach for sparse sufficient dimension reduction (SDR), using the MM-GCD algorithm for efficient computation.

Usage

```
ppsvm(
  x,
  y,
  H = 10,
  C = 1,
  lambda,
  gamma = 3.7,
  penalty = "grSCAD",
  max.iter = 100
)
```

Arguments

`x`	Numeric predictor matrix (n x p), where n is the sample size and p is the number of predictors.
`y`	Numeric response vector of length n.
`H`	The number of quantile slices (default: 10).
`C`	Regularization parameter (default: 1).
`lambda`	Penalty parameter for sparsity.
`gamma`	Regularization parameter for SCAD/MCP penalty (default: 3.7).
`penalty`	Penalty type: `"grSCAD"` (default), `"grLasso"`, or `"grMCP"`.
`max.iter`	Maximum number of iterations for the algorithm (default: 100).

Details

This function estimates a sparse basis for the central subspace by solving a penalized principal support vector machine objective with an MM-GCD optimization algorithm, enabling scalable and accurate sparse SDR.

Value

A list containing:

`Mn`	Estimated central subspace matrix.
`evalues`	Eigenvalues of the estimated matrix.
`evectors`	Eigenvectors (columns) corresponding to the estimated sufficient directions.
`x`	The original predictor matrix.

Examples

```
## Not run:
  library(MASS)
  set.seed(1)
  n <- 300; p <- 10
  B <- matrix(0, p, 2)
  B[1,1] <- B[2,2] <- 1
  x <- mvrnorm(n, rep(0, p), diag(1, p))
  y <- (x %*% B[,1] / (0.5 + (x %*% B[,2] + 1)^2)) + 0.2 * rnorm(n)
  fit <- ppsvm(x, y, H = 10, C = 100, lambda = 0.0001, gamma = 3.7,
         penalty = "grSCAD", max.iter = 100)
  fit$evectors[, 1:2]

## End(Not run)
```

`ppwl2svm` *Penalized Principal Weighted L2-Hinge SVM (P^2WL2M) for Sparse Sufficient Dimension Reduction*

Description

This function implements the penalized principal weighted L2-hinge support vector machine (P^2WL2M) method for sparse sufficient dimension reduction (SDR), typically for binary responses. Group penalties (SCAD, Lasso, MCP) are supported for variable selection.

Usage

```
ppwl2svm(
  x,
  y,
  H = 10,
  C = 1,
  lambda,
  gamma = 3.7,
  penalty = "grSCAD",
  max.iter = 100
)
```

Arguments

`x`	Numeric predictor matrix (n x p), where n is the sample size and p is the number of predictors.
`y`	Binary response vector of length n, coded as `-1, 1`.
`H`	Number of quantile slices for principal machine construction (default: 10).
`C`	Regularization parameter (default: 1).
`lambda`	Penalty parameter for sparsity.
`gamma`	Regularization parameter for SCAD/MCP penalty (default: 3.7).
`penalty`	Penalty type: `"grSCAD"` (default), `"grLasso"`, or `"grMCP"`.
`max.iter`	Maximum number of iterations (default: 100).

Details

This function solves a group-penalized principal machine with weighted L2 hinge loss, suitable for classification tasks. Efficient group coordinate descent (GCD) is used for parameter estimation. Use only with binary responses (-1, 1).

Value

A list with components:

evalues	Eigenvalues of the estimated subspace matrix.
evectors	Eigenvectors spanning the sufficient dimension reduction subspace.
x	The original predictor matrix.

Examples

```
## Not run:
  set.seed(1)
  n <- 1000; p <- 10
  x <- MASS::mvrnorm(n, rep(0, p), diag(1, p))
  y <- (x %*% B[,1] / (0.5 + (x %*% B[,2] + 1)^2)) + 0.2 * rnorm(n)
  y.binary <- sign(y)
  fit <- ppwl2svm(x, y.binary, H = 10, C = 5, lambda = 0.08, gamma = 3.7,
            penalty = "grSCAD", max.iter = 100)
  fit$evectors[, 1:2]

## End(Not run)
```

---

ppwlr — *Penalized Principal Weighted Logistic Regression (P^2WLR) for Sparse Sufficient Dimension Reduction*

Description

Implements the penalized principal weighted logistic regression (P^2WLR) method for sparse sufficient dimension reduction (SDR), typically for binary responses.

Usage

```
ppwlr(
  x,
  y,
  H = 10,
  C = 1,
  lambda = 0.001,
  gamma = 3.7,
  penalty = "grSCAD",
  max.iter = 100
)
```

Arguments

x	Numeric predictor matrix (n x p), where n is the sample size and p is the number of predictors.
y	Binary response vector of length n, coded as `-1, 1`.
H	Number of quantile slices for principal machine construction.
C	Regularization parameter.
lambda	Penalty parameter for sparsity.
gamma	Regularization parameter for SCAD/MCP penalty.
penalty	Penalty type: `"grSCAD"`, `"grLasso"`, or `"grMCP"`.
max.iter	Maximum number of iterations (default: 100).

Details

This function fits a penalized principal machine using weighted logistic regression loss, which is well-suited for a binary classification in sufficient dimension reduction. Group penalties (SCAD, Lasso, MCP) are available for structured sparsity and variable selection. Only use with binary responses (`-1, 1`).

Value

A list with components:

Mn	Estimated sufficient dimension reduction matrix.
evalues	Eigenvalues of `Mn`.
evectors	Eigenvectors spanning the SDR subspace.
x	Original predictor matrix.

Examples

```
## Not run:
  set.seed(1)
  n <- 1000; p <- 10
  x <- MASS::mvrnorm(n, rep(0, p), diag(1, p))
  y <- (x %*% B[,1] / (0.5 + (x %*% B[,2] + 1)^2)) + 0.2 * rnorm(n)
  y.binary <- sign(y)
  fit <- ppwlr(x, y.binary, H = 10, C = 10, lambda = 0.02, gamma = 3.7,
               penalty = "grSCAD", max.iter = 100)
  fit$evectors[, 1:2]

## End(Not run)
```

`ppwlssvm` *Penalized Principal Weighted Least Squares SVM (P^2WLSM) for Sparse Sufficient Dimension Reduction*

Description

Implements the penalized principal weighted least squares support vector machine (P^2WLSM) for sparse sufficient dimension reduction (SDR)

Usage

```
ppwlssvm(
  x,
  y,
  H = 10,
  C = 1,
  lambda,
  gamma = 3.7,
  penalty = "grSCAD",
  max.iter = 100
)
```

Arguments

`x`	Numeric predictor matrix (n x p), where n is the sample size and p is the number of predictors.
`y`	Binary response vector of length n, coded as `-1, 1`.
`H`	Number of quantile slices for principal machine construction (default: 10).
`C`	Regularization parameter.
`lambda`	Penalty parameter for sparsity. If NULL, cross-validation or an information criterion will be used for selection.
`gamma`	Regularization parameter for SCAD/MCP penalty (default: 3.7).
`penalty`	Penalty type: `"grSCAD"`, `"grLasso"`, or `"grMCP"`.
`max.iter`	Maximum number of iterations (default: 100).

Details

This function fits a penalized principal machine using a weighted least squares SVM loss, suitable for binary responsess. It supports group penalties for structured sparsity and variable selection. Only use with binary responses (`-1, 1`).

Value

A list with components:

`evalues`	Eigenvalues of the estimated sufficient dimension reduction matrix.
`evectors`	Eigenvectors spanning the SDR subspace.
`x`	(Optional) Original predictor matrix.

Examples

```
## Not run:
  set.seed(1)
  n <- 1000; p <- 10
  x <- MASS::mvrnorm(n, rep(0, p), diag(1, p))
  y <- (x %*% B[,1] / (0.5 + (x %*% B[,2] + 1)^2)) + 0.2 * rnorm(n)
  y.binary <- sign(y)
  fit <- ppwlssvm(x, y.binary, H = 10, C = 1, lambda = 0.0003, gamma = 3.7,
                  penalty = "grSCAD", max.iter = 100)
  fit$evectors[, 1:2]

## End(Not run)
```

`ppwsvm` *Penalized Principal Weighted SVM (MM-GCD based) for Sparse Sufficient Dimension Reduction*

Description

Implements the penalized principal weighted support vector machine (P^2WSVM) for sparse sufficient dimension reduction (SDR), using an MM-GCD (majorization-minimization group coordinate descent) algorithm. Observation-level weights are incorporated to account for class imbalance or sampling design, and group penalties (SCAD, Lasso, MCP) are supported for structured sparsity.

Usage

```
ppwsvm(
  x,
  y,
  H = 10,
  C = 1,
  lambda,
  gamma = 3.7,
  penalty = "grSCAD",
  max.iter = 100
)
```

Arguments

`x`	Numeric predictor matrix (n x p), where n is the sample size and p is the number of predictors.
`y`	Binary response vector of length n, coded as `-1, 1`.
`H`	Number of quantile slices for principal machine construction (default: 10).
`C`	Regularization parameter.
`lambda`	Penalty parameter for sparsity.
`gamma`	Regularization parameter for SCAD/MCP penalty (default: 3.7).
`penalty`	Penalty type: `"grSCAD"`, `"grLasso"`, or `"grMCP"`.
`max.iter`	Maximum number of iterations (default: 100).

Details

This function fits a penalized principal machine using a weighted SVM loss, suitable for binary response problems. It applies group penalties to facilitate variable selection and structured sparsity. The majorization-minimization group coordinate descent (MM-GCD) algorithm is used for efficient computation. Only use with binary responses (`-1, 1`).

Value

A list with components:

`Mn`	Estimated SDR matrix.
`evalues`	Eigenvalues of the SDR matrix.
`evectors`	Eigenvectors spanning the SDR subspace.
`x`	(Optional) The original predictor matrix.

Examples

```
## Not run:
  set.seed(1)
  n <- 2000; p <- 10
  x <- MASS::mvrnorm(n, rep(0, p), diag(1, p))
  y <- (x %*% B[,1] / (0.5 + (x %*% B[,2] + 1)^2)) + 0.2 * rnorm(n)
  y.binary <- sign(y)
  fit <- ppwsvm(x, y.binary, H = 10, C = 2, lambda = 0.00003,
               gamma = 3.7, penalty = "grSCAD", max.iter = 100)
  fit$evectors[, 1:2]

## End(Not run)
```

Index