Package ‘sSDR’

March 26, 2016

Type Package

Title Tools Developed for Structured Sufficient Dimension Reduction (sSDR)

Version 1.2.0

Date 2016-03-26

Author Yang Liu <zjubioly@gmail.com>, Francesca Chiaromonte, Bing Li

Maintainer Yang Liu <zjubioly@gmail.com>

Description Performs structured OLS (sOLS) and structured SIR (sSIR).

License GPL (>= 2)

LazyData TRUE

Depends R (>= 3.0.0), MASS, Matrix

NeedsCompilation no

Repository CRAN

Date/Publication 2016-03-26 22:02:24

R topics documented:

  center ................................................................. 2
  cov.x ................................................................. 2
  disvm ................................................................. 3
  gen.data ............................................................. 4
  gOLS ................................................................. 5
  gOLS.comp.d ....................................................... 6
  gSIR ................................................................. 7
  gSIR.comp.d ....................................................... 8
  matpower ......................................................... 9
  norm1 ............................................................. 9
  orthnormal ..................................................... 10
  sOLS.comp.d ................................................... 11
  standmat ...................................................... 12

Index 13
**center**

*Center a vector*

**Description**

Center a vector

**Usage**

`center(v)`

**Arguments**

- `v` A vector.

**Details**

This function centers any vector and returns a vector with mean zero.

**Value**

A vector with mean zero.

**Examples**

```r
data <- gen.data(n=100)
y.centered <- center(data$y)
```

---

**cov.x**

*Covariance matrix*

**Description**

Covariance matrix

**Usage**

`cov.x(X)`

**Arguments**

- `X` a n x p matrix of n observations and p predictors.

**Details**

This function returns a p x p covariance matrix for any n x p matrix.
disvm

Value
A p x p covariance matrix.

Examples
```r
data <- gen.data(n=100)
x.cov <- cov.x(data$X)
```

---

**disvm**

Subspace distance

**Description**
Subspace distance

**Usage**
```r
disvm(v1, v2)
```

**Arguments**
- `v1` A matrix, each column consists of a p-dimensional vector.
- `v2` A matrix, each column consists of a p-dimensional vector.

**Details**
This function computes the distances between two spaces using the formulation in Li, Zha, Chiaromonte (2005), which is the Frobenius norm of the difference between the two orthogonal projection matrices defined by v1 and v2.

**Value**
A scaler represents the distance between the two spaces spanned by v1 and v2 respectively.

**References**

**Examples**
```r
v1 <- c(1, 0, 0)
v2 <- c(0, 1, 0)
disvm(v1, v1)
disvm(v1, v2)
```
**gen.data**

Simulate data

**Usage**

```r
gen.data(n, rho = 0.5, theta = 1, binary = FALSE)
```

**Arguments**

- `n` Sample size.
- `rho` Pairwise correlation between covariates.
- `theta` Standard deviation of the random error.
- `binary` If TRUE, generate binary responses; otherwise, by default, create continuous responses.

**Details**

This function simulates data as presented in Liu (2015).

**Value**

`gen.data` returns a list containing at least the following components: "X", a covariate matrix of `n` observations and `p` predictors; "y", a univariate response; "b.true", the actual coefficients for each predictor group.

**References**


**Examples**

```r
data <- gen.data(n=100)
names(data)
```
**gOLS**  

*Groupwise OLS (gOLS)*

**Description**

Groupwise OLS (gOLS)

**Usage**

```r
gOLS(X, Y, groups, dims)
```

**Arguments**

- **X**: A covariate matrix of n observations and p predictors.
- **Y**: A univariate response.
- **groups**: A vector with the number of predictors in each group.
- **dims**: A vector with the dimension (at most 1) for each predictor group.

**Details**

This function estimates directions for each predictor group using gOLS. Predictors need to be organized in groups within the "X" matrix, as the same order saved in "groups". We only allow continuous covariates in the "X" matrix; while categorical covariates can be handled outside of gOLS, e.g. structured OLS.

**Value**

gOLS returns a list containing at least the following components: "b_est", the estimated directions for each group with its own dimension using gOLS AFTER normalization; "B", the estimated directions for each group using gOLS BEFORE normalization.

**References**


**Examples**

```r
data <- gen.data(n=1000, binary=FALSE) # generate data
dim(data$X) # covariate matrix of 1000 observations and 15 predictors
dim(data$y) # univariate response
groups <- c(5, 10) # two predictor groups and their numbers of predictors
dims <- c(1, 1) # dimension of each predictor group
est_gOLS <- gOLS(data$X, data$y, groups, dims)
names(est_gOLS)
```
gOLS.comp.d  

Groupwise OLS (gOLS) BIC criterion to estimate dimensions with eigen-decomposition

Description

Groupwise OLS (gOLS) BIC criterion to estimate dimensions with eigen-decomposition

Usage

gOLS.comp.d(x, y, groups)

Arguments

- **x**: A covariate matrix of n observations and p predictors.
- **y**: A univariate response.
- **groups**: A vector with the number of predictors in each group.

Details

This function estimates dimension for each predictor group using eigen-decomposition. Predictors need to be organized in groups within the "X" matrix, as the same order saved in "groups". We only allow continuous covariates in the "X" matrix; while categorical covariates can be handled outside of gOLS, e.g. structured OLS.

Value

gOLS.comp.d returns a list containing at least the following components: "d", the estimated dimension (at most 1) for each predictor group; "crit", the BIC criterion from each iteration.

References


Examples

data <- gen.data(n=1000, binary=FALSE) # generate data
dim(data$x) # covariate matrix of 1000 observations and 15 predictors
dim(data$y) # univariate response
groups <- c(5, 10) # two predictor groups and their numbers of predictors
dim_gOLS <- gOLS.comp.d(data$x, data$y, groups)
names(dim_gOLS)
gSIR

**Groupwise SIR (gSIR) for binary response**

**Description**

Groupwise SIR (gSIR) for binary response

**Usage**

```r
gSIR(X, Y, groups, dims)
```

**Arguments**

- `X` A covariate matrix of `n` observations and `p` predictors.
- `Y` A binary response.
- `groups` A vector with the number of predictors in each group.
- `dims` A vector with the dimension (at most 1) for each predictor group.

**Details**

This function estimates directions for each predictor group using gSIR. Predictors need to be organized in groups within the "X" matrix, as the same order saved in "groups". We only allow continuous covariates in the "X" matrix; while categorical covariates can be handled outside of gSIR, e.g. structured SIR.

**Value**

gSIR returns a list containing at least the following components: "b_est", the estimated directions for each group with its own dimension using gSIR AFTER normalization; "B", the estimated directions for each group using gSIR BEFORE normalization.

**References**


**Examples**

```r
data <- gen.data(n=1000, binary=TRUE) # generate data
dim(data$X) # covariate matrix of 1000 observations and 15 predictors
length(data$y) # binary response
groups <- c(5, 10) # two predictor groups and their numbers of predictors
dims <- c(1, 1) # dimension of each predictor group
est_gSIR <- gSIR(data$X, data$y, groups, dims)
names(est_gSIR)
```
gSIR.comp.d

Groupwise SIR (gSIR) BIC criterion to estimate dimensions with eigen-decomposition (binary response)

Description
Groupwise SIR (gSIR) BIC criterion to estimate dimensions with eigen-decomposition (binary response)

Usage

```r
gSIR.comp.d(x, y, groups)
```

Arguments

- `x` A covariate matrix of n observations and p predictors.
- `y` A binary response.
- `groups` A vector with the number of predictors in each group.

Details
This function estimates dimension for each predictor group using eigen-decomposition. Predictors need to be organized in groups within the "X" matrix, as the same order saved in "groups". We only allow continuous covariates in the "X" matrix; while categorical covariates can be handled outside of gSIR, e.g. structured SIR.

Value

`gSIR.comp.d` returns a list containing at least the following components: "d", the estimated dimension (at most 1) for each predictor group; "crit", the BIC criterion from each iteration.

References


Examples

```r
data <- gen.data(n=1000, binary=TRUE) # generate data
dim(data$x) # covariate matrix of 1000 observations and 15 predictors
length(data$y) # univariate response
groups <- c(5, 10) # two predictor groups and their numbers of predictors
dim_gSIR <- gSIR.comp.d(data$x, data$y, groups)
names(dim_gSIR)
```
**matpower**

_Power of a matrix_

**Description**

Power of a matrix

**Usage**

`matpower(X, alpha)`

**Arguments**

- `alpha` : A scaler determining the order of the power.

**Details**

This function calculates the power of a square matrix.

**Value**

A p x p square matrix.

**Examples**

```r
data <- gen.data(n=100)
cov.squared <- matpower(cov.x(data$x), 2)
```

---

**norm1**

_Notify a vector_

**Description**

Normalize a vector

**Usage**

`norm1(v)`

**Arguments**

- `v` : A vector.

**Details**

This function normalizes any non-zero vector and returns a vector with the norm equal to 1.
Value

A vector with norm 1.

Examples

data <- gen.data(n=100)
y.norm1 <- norm1(data$y)

---

**orthnormal**  
*Gram-Schmidt orthonormalization*

Description

Gram-Schmidt orthonormalization

Usage

orthnormal(X)

Arguments

X  
a n x p matrix of n observations and p predictors.

Details

This function orthonormalizes any n x p matrix.

Value

A n x p matrix of n observations and p predictors.

Examples

data <- gen.data(n=100)
x.orth <- orthnormal(data$X)
Description
Structured OLS (sOLS) outer level BIC criterion to estimate dimension with eigen-decomposition

Usage
sOLS.comp.d(X, sizes)

Arguments
X A matrix containing directions estimated from all subpopulations.
sizes A vector with the sample sizes of all subpopulation.

Details
This function estimates dimension across the subpopulations using eigen-decomposition. The order of the subpopulations in the "sizes" vector should match the one in the "X" matrix. Also, this function returns the linearly independent directions among all subpopulations.

Value
sOLS.comp.d returns a list containing at least the following components: "d", the dimension estimated across subpopulations; "u", the "d" linearly independent directions among the matrix X.

References

Examples
v1 <- c(1, 1, 0, 0)
v2 <- c(0, 1, 1, 0)
v3 <- c(0, 0, 1, 1)
v4 <- c(1, 1, 1, 1)
m1 <- cbind(v1, v2)
sizes1 <- c(100, 200)
sOLS.comp.d(m1, sizes1)
m2 <- cbind(v1, v2, v3)
sizes2 <- c(100, 200, 500)
sOLS.comp.d(m2, sizes2)
m3 <- cbind(v1, v3, v4)
sizes3 <- c(100, 500, 1000)
sOLS.comp.d(m3, sizes3)
standmat  Matrix standardization

Description

Matrix standardization

Usage

standmat(x)

Arguments

x  A n x p matrix of n observations and p predictors.

Details

This function standardizes a matrix treating each row as a random vector in an iid sample. It returns a n x p matrix with column-mean zero and identity-covariance matrix.

Value

A n x p matrix of n observations and p predictors.

Examples

data <- gen.data(n=100)
x.std <- standmat(data$X)
Index

center, 2
cov.x, 2
dismv, 3
gen.data, 4
gOLS, 5
gOLS.comp.d, 6
gSIR, 7
gSIR.comp.d, 8
matpower, 9

norm1, 9
orthnormal, 10

sOLS.comp.d, 11
standmat, 12