Package 'oesir'

September 10, 2026

Type Package

Title Online Sliced Inverse Regression for Elliptical Model with Streaming Data

Version 0.3.2

Author Sirui Yan [aut],
Guangbao Guo [aut, cre]

Maintainer Guangbao Guo <ggb11111111@163.com>

Description For high-dimensional streaming heavy-tailed elliptical data, traditional sliced inverse regression methods require full offline data and cannot adapt to incremental data arrival. This package implements Online Sliced Inverse Regression for Elliptical Model with Streaming Data (OESIR) algorithm with two recursive updating strategies, including offline batch SIR as benchmark, elliptical heavy-tailed data simulator, subspace evaluation metric and batch simulation tools for numerical experiments. Cai, Z., Li, R., & Zhu, L. (2020) <doi:10.48550/arXiv.2002.02795>.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 3.5.0)

Imports stats, Matrix, MASS

NeedsCompilation no

Language en-US

Config/roxygen2/version 8.0.0

Repository CRAN

Date/Publication 2026-09-10 09:20:08 UTC

Contents

`generate_elliptical` *Generate Elliptical Streaming Data*

Description

Generate elliptical normal / student-t heavy-tailed covariate stream. Simulate streaming high-dimensional elliptical predictors and response, with true subspace and covariance matrix returned for subspace evaluation.

Usage

```
generate_elliptical(n, p, nu = 3, sigma = 0.3, heavy = TRUE)
```

Arguments

`n`	integer, total sample size.
`p`	integer, covariate dimension.
`nu`	numeric, t-distribution degrees of freedom.
`sigma`	numeric, scale coefficient for predictor matrix.
`heavy`	logical; `TRUE` for elliptical-t heavy-tail, `FALSE` for elliptical normal distribution.

Value

A list containing:

`X` numeric matrix of dimension `n * p`, simulated predictor matrix.

`Y` numeric vector of length `n`, simulated response variable.

`B.true` numeric matrix, true effective dimension reduction subspace.

`Sigma` numeric matrix, true population covariance matrix.

Examples

```
# tiny toy example for CRAN automatic checking (run <5s)
set.seed(123)
dat <- generate_elliptical(n = 80, p = 6, nu = 3, sigma = 0.3, heavy = TRUE)
dim(dat$X)
length(dat$Y)
dim(dat$B.true)

# larger simulation example, skipped in auto-check

set.seed(123)
dat_large <- generate_elliptical(n = 2000, p = 30)
```

oe_sir *Online Elliptical SIR (Two Online Solvers)*

Description

Streaming recursive OE-SIR algorithm, supporting gradient-based and perturbation-based two recursive updating strategies for streaming elliptical high-dimensional data.

Usage

```
oe_sir(X, Y, H = 5, K = 2, method = c("grad", "pert"), C = 1)
```

Arguments

X	numeric matrix, full streaming predictor matrix.
Y	numeric vector, response variable.
H	integer, number of slices for sliced inverse regression.
K	integer, target dimension of effective subspace.
method	character; updating method, either "grad" for gradient descent or "pert" for perturbation update.
C	numeric, step-size constant for gradient descent update.

Value

A list containing:

B_hat numeric matrix of dimension p * K, estimated effective dimension reduction subspace directions.

Examples

```
# tiny toy example for CRAN automatic checking (run <5s)
set.seed(123)
dat <- generate_elliptical(n = 80, p = 6)
res <- oe_sir(X = dat$X, Y = dat$Y, H = 5, K = 2, method = "grad")
dim(res$B_hat)

# larger streaming example, skipped in auto-check

set.seed(123)
dat_large <- generate_elliptical(n = 2000, p = 30)
res_pert <- oe_sir(X = dat_large$X, Y = dat_large$Y, method = "pert")
```

offline_sir *Offline Batch SIR (Baseline Comparison Algorithm)*

Description

Classical full-data sliced inverse regression, offline batch-mode baseline algorithm for estimating the central subspace.

Usage

```
offline_sir(X, Y, H = 5, K = 2)
```

Arguments

X	numeric matrix of dimension n * p, covariate matrix.
Y	numeric vector, response variable.
H	integer, number of slices for sliced inverse regression.
K	integer, target dimension of central subspace.

Value

Numeric matrix p * K, the estimated central subspace basis directions.

Examples

```
# tiny toy example for CRAN automatic checking (run <5s)
set.seed(123)
dat <- generate_elliptical(n = 80, p = 6)
res_off <- offline_sir(X = dat$X, Y = dat$Y, H = 5, K = 2)
dim(res_off)

# larger example, skipped in auto-check

set.seed(123)
dat_large <- generate_elliptical(n = 2000, p = 30)
res_off_large <- offline_sir(X = dat_large$X, Y = dat_large$Y)
```

`sim_ablation` *Batch Ablation Simulation Tool*

Description

Generate simulation result table for numerical experiment and paper visualization. Run repeated experiments over grids of sample size and dimension, compare OE-SIR and offline SIR subspace estimation error.

Usage

```
sim_ablation(
  heavy = TRUE,
  n_vec = c(2000, 4000, 6000),
  p_vec = c(8, 10, 12),
  rep = 5
)
```

Arguments

`heavy`	logical; if `TRUE`, use heavy-tailed elliptical-t distribution; otherwise use regular elliptical data.
`n_vec`	numeric vector, grid of sample sizes for simulation.
`p_vec`	numeric vector, grid of predictor dimensions for simulation.
`rep`	integer, number of replication times per simulation setting.

Value

A `data.frame` containing simulation summary results: sample size `n`, dimension `p`, averaged MSE for OESIR and offline SIR.

Examples

```
# tiny toy example for CRAN automatic checking (small rep to save time)
set.seed(123)
sim_res <- sim_ablation(heavy = TRUE, n_vec = c(200), p_vec = c(6), rep = 2)
print(sim_res)

# full simulation, skipped in auto-check due to computation cost

sim_full <- sim_ablation(heavy = TRUE, n_vec = c(2000,4000), p_vec = c(8,10), rep = 5)
```

subspace_mse *Subspace Projection MSE Loss Function*

Description

Calculate mean-squared estimation error between estimated subspace projection matrix and true subspace projection matrix.

Usage

```
subspace_mse(est, true)
```

Arguments

est	Numeric matrix, estimated subspace basis matrix.
true	Numeric matrix, ground-truth subspace basis matrix.

Value

Numeric scalar, mean squared error between two projection matrices.

Examples

```
# tiny runnable example for CRAN check
set.seed(123)
true_sub <- matrix(rnorm(10*2), 10, 2)
est_sub  <- true_sub + matrix(rnorm(10*2, sd = 0.01), 10, 2)
subspace_mse(est = est_sub, true = true_sub)
```

Index