Package: osir (via r-universe)

November 14, 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 (OSIR) 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 https://guangbaog.r-universe.dev

Date/Publication 2026-11-13 22:20:02 UTC

RemoteUrl https://github.com/cran/osir

RemoteRef HEAD

RemoteSha 23dc9f6498e5266b6e31286d40e0307b347e563d

Contents

generate_elliptical 2
offline_sir 3
osir 3
sim_ablation 4
subspace_mse 5

Index **6**

`generate_elliptical` *Generate Elliptical Streaming Data Generate elliptical normal / student-t heavy-tailed covariate stream*

Description

Generate Elliptical Streaming Data Generate elliptical normal / student-t heavy-tailed covariate stream

Usage

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

Arguments

`n`	total sample size
`p`	covariate dimension
`nu`	t-distribution freedom
`sigma`	scale coefficient
`heavy`	TRUE = elliptical-t, FALSE = elliptical normal

Value

A list with components:

X An n by p numeric matrix of generated elliptically distributed covariates.

Y A numeric response vector of length n.

B.true A p by 2 orthonormal matrix; its first column spans the true central direction.

Sigma A p by p positive definite covariance matrix used to scale `X`.

Examples

```
set.seed(1)
dat <- generate_elliptical(n = 100, p = 6)
dim(dat$X)
length(dat$Y)
```

`offline_sir` *Offline Batch SIR (Baseline Comparison Algorithm) Classical full-data sliced inverse regression*

Description

Offline Batch SIR (Baseline Comparison Algorithm) Classical full-data sliced inverse regression

Usage

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

Arguments

`X`	n*p covariate matrix
`Y`	response vector
`H`	slice number
`K`	target central subspace dimension

Value

A p by `K` numeric matrix whose columns are the estimated SIR directions, i.e. the leading `K` eigenvectors of the sliced covariance matrix.

Examples

```
set.seed(1)
dat <- generate_elliptical(n = 100, p = 6)
offline_sir(dat$X, dat$Y)
```

`osir` *Online SIR (Two Online Solvers) Streaming recursive OSIR, gradient / perturbation two update methods*

Description

Online SIR (Two Online Solvers) Streaming recursive OSIR, gradient / perturbation two update methods

Usage

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

Arguments

X	full streaming matrix
Y	response vector
H	slice count
K	target subspace dimension
method	"grad" gradient descent / "pert" perturbation update
C	gradient step constant

Value

A list with component B_hat: a p by K numeric matrix of the recursively updated estimated directions after scanning the whole stream X.

Examples

```
set.seed(1)
dat <- generate_elliptical(n = 80, p = 5)
osir(dat$X, dat$Y, method = "grad")$B_hat
```

---

sim_ablation *Batch Ablation Simulation Tool Generate experiment table for paper drawing*

Description

Batch Ablation Simulation Tool Generate experiment table for paper drawing

Usage

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

Arguments

heavy	TRUE = heavy-tailed elliptical-t
n_vec	sample size grid
p_vec	dimension grid
rep	repeat experiment times

Value

A data.frame with one row per (n,p) setting and columns n, p, OSIR_MSE and Offline_MSE, giving the averaged subspace projection MSE over rep replications for the online and offline solvers.

subspace_mse *Subspace Projection MSE Loss Function*

Description

Calculate estimation error between true & estimated subspace

Usage

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

Arguments

est	estimated subspace matrix
true	real true subspace basis

Value

A non-negative numeric scalar giving the mean squared error between the projection matrices of the estimated and true subspaces.

Examples

```
set.seed(123)
# Generate demo subspace
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

generate_elliptical, 2

offline_sir, 3
osir, 3

sim_ablation, 4
subspace_mse, 5