Package: AdaptiveGlasso (via r-universe)

October 25, 2026

Title Adaptive Graphical Lasso for Sparse Network Estimation

Version 1.0.0

Description Implements an adaptive graphical lasso method for sparse precision matrix and network estimation. The method combines ridge stabilization, cross-validated regularization, an initial graphical lasso estimate, and adaptive edge-specific penalties to estimate sparse conditional dependence structures. The method supports settings where the sample size is smaller than, equal to, or larger than the number of variables.

URL https://github.com/elizabethamona/AdaptiveGlasso

BugReports https://github.com/elizabethamona/AdaptiveGlasso/issues

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Imports CVglasso, glasso

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Repository https://elizabethamona.r-universe.dev

Date/Publication 2026-10-13 04:30:44 UTC

RemoteUrl https://github.com/elizabethamona/adaptiveglasso

RemoteRef HEAD

RemoteSha 3a27aef313a16e575e61b605604a5367b6012f5d

Contents

adaptive_glasso 2
coef.adaptive_glasso_fit 4
print.adaptive_glasso_fit 5

Index **6**

`adaptive_glasso` *Fit an adaptive graphical lasso model*

Description

Fits an adaptive graphical lasso model for sparse precision-matrix and network estimation. The procedure combines ridge stabilization, an initial graphical lasso estimate, adaptive edge-specific penalties, and a final weighted graphical lasso estimate.

Usage

```
adaptive_glasso(
  x,
  ridge = 0.1,
  d = 0.32,
  lambda = NULL,
  lam.min.ratio = 0.01,
  nfolds = 5L,
  standardize = TRUE,
  seed = 123L,
  penalize_diagonal = TRUE,
  tol = 1e-04,
  maxit = 10000L
)
```

Arguments

`x`	Numeric matrix or data frame with observations in rows and variables in columns.
`ridge`	Non-negative ridge stabilization constant added to the diagonal of the sample covariance matrix. Default is `0.10`.
`d`	Positive stabilization constant used in the adaptive weights $w_{ij} = 1/(\lvert\hat{\omega}_{ij}^{init}\rvert + d)$. Default is `0.32`.
`lambda`	Optional positive regularization parameter. If `NULL`, `lambda` is selected automatically by cross-validation using `CVglasso`. If a positive numeric value is supplied, that fixed value is used and cross-validation is skipped.
`lam.min.ratio`	Minimum-to-maximum lambda ratio used by `CVglasso` when `lambda = NULL`. Default is `0.01`.
`nfolds`	Number of cross-validation folds used when `lambda = NULL`. Default is `5`.
`standardize`	Logical; if `TRUE`, variables are standardized before fitting. Default is `TRUE`.
`seed`	Optional integer seed used for reproducibility of cross-validation when `lambda = NULL`. Default is `123`.
`penalize_diagonal`	Logical; whether diagonal precision-matrix elements are penalized. Default is `TRUE`, matching the fitting procedure used in the motivating analysis.

`tol`	Positive convergence tolerance passed to the graphical lasso fitting routines. Default is `1e-4`.
`maxit`	Maximum number of graphical lasso iterations. Default is `10000`.

Details

The sample covariance matrix is first ridge-stabilized as

$$S_\delta = S + \delta I,$$

where `ridge` corresponds to δ.

An initial graphical lasso estimate is then used to construct the adaptive weights

$$w_{ij} = \frac{1}{|\hat{\omega}_{ij}^{init}| + d},$$

followed by the edge-specific penalties

$$\rho_{ij} = \lambda w_{ij}.$$

The value of `lambda` can be handled in two ways:

- `lambda = NULL`: `lambda` is selected automatically by cross-validation using `CVglasso`.
- `lambda =` a positive numeric value: the supplied value is used directly and cross-validation is not performed.

The method can be applied when the number of observations is smaller than, equal to, or larger than the number of variables.

Value

An object of class `"adaptive_glasso_fit"` containing the fitted precision and covariance matrices, adjacency matrix, initial precision estimate, adaptive weights, adaptive penalty matrix, selected or supplied `lambda`, ridge and adaptive stabilization constants, cross-validation results when applicable, and fitting diagnostics.

Examples

```
set.seed(123)

x <- matrix(
  rnorm(60 * 6),
  nrow = 60,
  ncol = 6
)

# Use a fixed lambda: cross-validation is not performed
fit <- adaptive_glasso(
  x,
  lambda = 0.195237
)
```

```
fit
coef(fit, type = "precision")

# Select lambda automatically by cross-validation
fit_cv <- adaptive_glasso(
  x,
  lambda = NULL,
  nfolds = 3
)

fit_cv$lambda
```

`coef.adaptive_glasso_fit`

Extract matrices from an adaptive graphical lasso fit

Description

Extract matrices from an adaptive graphical lasso fit

Usage

```
## S3 method for class 'adaptive_glasso_fit'
coef(
  object,
  type = c("precision", "adjacency", "weights", "penalty", "covariance"),
  ...
)
```

Arguments

`object`	An object returned by `adaptive_glasso()`.
`type`	Matrix to return.
`...`	Additional arguments; currently unused.

Value

A matrix.

`print.adaptive_glasso_fit` *Print an adaptive graphical lasso fit*

Description

Print an adaptive graphical lasso fit

Usage

```
## S3 method for class 'adaptive_glasso_fit'
print(x, ...)
```

Arguments

`x`	An object returned by `adaptive_glasso()`.
`...`	Additional arguments; currently unused.

Value

The fitted object, invisibly.

Index

adaptive_glasso, 2

coef.adaptive_glasso_fit, 4

print.adaptive_glasso_fit, 5