Package 'Glarmadillo'

May 7, 2026

Title Solve the Graphical Lasso Problem with 'Armadillo'

Version 1.1.1

Description Efficiently implements the Graphical Lasso algorithm, utilizing the 'Armadillo' 'C++' library for rapid computation. This algorithm introduces an L1 penalty to derive sparse inverse covariance matrices from observations of multivariate normal distributions. Features include the generation of random and structured sparse covariance matrices, beneficial for simulations, statistical method testing, and educational purposes in graphical modeling. A unique function for regularization parameter selection based on predefined sparsity levels is also offered, catering to users with specific sparsity requirements in their models. The methodology for sparse inverse covariance estimation implemented in this package is based on the work of Friedman, Hastie, and Tibshirani (2008) <doi:10.1093/biostatistics/kxm045>.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.2.3

Imports stats, Rcpp (>= 0.12), RcppArmadillo

Depends R (>= 3.3)

LinkingTo Rcpp (>= 0.12), RcppArmadillo

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation yes

Author Alessandro Meng [aut, cre]

Maintainer Alessandro Meng <mengfangeng@ruc.edu.cn>

Repository CRAN

Date/Publication 2023-12-15 12:40:09 UTC

Contents

`find_lambda_by_sparsity` *Find Optimal Lambda by Sparsity Level*

Description

This function performs a grid search over a range of lambda values to identify the lambda that achieves a desired level of sparsity in the precision matrix estimated by Graphical Lasso. Sparsity is defined as the proportion of zero elements (excluding the diagonal) in the precision matrix.

Usage

```
find_lambda_by_sparsity(
  s,
  lambda_grid,
  desired_sparsity,
  mtol = 1e-04,
  maxIterations = 10000,
  ltol = 1e-06
)
```

Arguments

`s`	The sample covariance matrix of the data.
`lambda_grid`	A numeric vector of lambda values to be tested in the grid search.
`desired_sparsity`	The target sparsity level as a proportion of zero elements in the precision matrix. This should be a value between 0 and 1.
`mtol`	The convergence threshold for Graphical Lasso optimization.
`maxIterations`	The maximum number of iterations for Graphical Lasso optimization.
`ltol`	The tolerance for determining whether elements are considered zero when calculating sparsity.

Value

A list containing the following components: - `best_lambda`: the lambda value that results in sparsity closest to the desired level. - `best_sparsity_difference`: the smallest difference between achieved and desired sparsity. - `actual_sparsity`: a numeric vector of actual sparsity levels for each lambda tested. - `lambda_grid`: the vector of lambda values tested.

Examples

```
# Generate a sparse covariance matrix
values <- c(160, 50)
n <- values[1]
p <- values[2]
s <- generate_sparse_cov_matrix(n, p, standardize = TRUE, sparse_rho = 0, scale_power = 0)

# Define a sequence of lambda values for the grid search
lambda_find <- c(0.1, 0.2, 0.3, 0.4)

# Perform a grid search to find the lambda value
# that results in a precision matrix with approximately 80% sparsity
lambda_results <- find_lambda_by_sparsity(s, lambda_find, desired_sparsity = 0.8)

# Inspect the optimal lambda value
optimal_lambda <- lambda_results$best_lambda

# Inspect the sparsity levels for each lambda tested
sparsity_levels <- lambda_results$actual_sparsity
```

---

generate_sparse_cov_matrix

Generate Sparse Covariance Matrix

Description

Generates a sparse covariance matrix with specified dimension and rank. The generated matrix can be scaled or standardized, and further sparsified based on a given threshold.

Usage

```
generate_sparse_cov_matrix(
  n,
  p,
  standardize = TRUE,
  sparse_rho = 0,
  scale_power = 0
)
```

Arguments

n	The dimension of the covariance matrix (number of rows and columns).
p	The rank of the covariance matrix (number of non-zero eigenvalues). Must be less than or equal to n.
standardize	Logical indicating whether to standardize the matrix, setting this to TRUE overrides scale_power and sparse_rho.

sparse_rho	Numeric threshold for enforcing sparsity. Elements with absolute values below sparse_rho are set to zero.
scale_power	The exponent used to scale the matrix elements. Only used if standardize is FALSE.

Value

A n by n covariance matrix with rank p. If sparse_rho is greater than zero and standardize is FALSE, elements with absolute values below sparse_rho are set to zero to increase sparsity, while ensuring that the matrix is at least semi-definite.

Examples

```
# Generate a 10x10 sparse covariance matrix with rank 5
sparse_cov_matrix <- generate_sparse_cov_matrix(n = 10, p = 5)

# Generate a sparser matrix with elements below 0.3 set to zero
sparser_cov_matrix <- generate_sparse_cov_matrix(n = 100, p = 50,
                                                 sparse_rho = 0.3,
                                                standardize = FALSE)

# Generate a standardized matrix
standardized_cov_matrix <- generate_sparse_cov_matrix(n = 100, p = 50, standardize = TRUE)
```

---

generate_specific_shape_sparse_cov_matrix

Generate Specific Shape Sparse Covariance Matrix

Description

Generates a covariance matrix and corresponding data matrix (Y) with a specific shape defined by a given shape matrix (M). This function is particularly useful for simulating data with predefined covariance structures, facilitating the testing of statistical methods such as sparse covariance estimation.

Usage

```
generate_specific_shape_sparse_cov_matrix(n, p, M)
```

Arguments

n	The number of variables (rows of Y and dimensions of M).
p	The number of samples (columns of Y).
M	The shape matrix used to define the structure of the covariance matrix. Must be a positive definite square matrix of size n x n.

Value

A list containing two elements: - Y: A n by p data matrix, where each column represents a sample, and each row represents a variable. - cov_Y: The n by n covariance matrix of the transposed data matrix Y. This covariance matrix reflects the structure imposed by the shape matrix M.

Examples

```
# Generate a 10x10 specific shape sparse covariance matrix
shape_matrix <- matrix(rnorm(100), 10, 10)
shape_matrix <- shape_matrix %*% t(shape_matrix) # Making it positive definite
result <- generate_specific_shape_sparse_cov_matrix(n = 10, p = 5, M = shape_matrix)
Y <- result$Y
cov_Y <- result$cov_Y
```

---

glarma *Solve Graphical Lasso with Armadillo*

Description

This function solves the Graphical Lasso (GLasso) problem using the Armadillo library. GLasso is a technique used in statistical learning and network analysis to estimate sparse inverse covariance matrices from observed data.

Usage

```
glarma(s, rho, mtol = 1e-04, maxIterations = 10000, ltol = 1e-06)
```

Arguments

s	A symmetric, positive-definite sample covariance matrix. It should be a square matrix representing the covariance matrix of the variables.
rho	A positive scalar representing the regularization parameter. It controls the sparsity level of the inverse covariance matrix.
mtol	A numeric value representing the convergence threshold for the main algorithm. It determines the condition under which the iterative process will stop. Default is 1e-4.
maxIterations	An integer value specifying the maximum number of iterations allowed for the algorithm. Default is 10000.
ltol	A numeric value representing the convergence threshold for the Lasso solver. It is used to control the Lasso solving process within the algorithm. Default is 1e-6.

Value

Returns a covariance matrix W and a estimated sparse inverse covariance matrix Theta estimated by solving the Graphical Lasso problem. The sparsity is controlled by the 'rho' parameter.

Examples

```
# Generate a sample covariance matrix
s <- matrix(runif(100), nrow = 10)
s <- t(s) %*% s
# Solve the Graphical Lasso problem with default parameters
inv_cov_matrix <- glarma(s, rho = 0.1)
# Solve with custom convergence thresholds and maximum iterations
inv_cov_matrix <- glarma(s, rho = 0.1, mtol = 1e-5, maxIterations = 5000, ltol = 1e-6)
```

Index