Package: gif (via r-universe)

September 9, 2024

Type Package

Version 0.1.1

Title Graphical Independence Filtering

Description Provides a method of recovering the precision matrix for Gaussian graphical models efficiently. Our approach could be divided into three categories. First of all, we use Hard Graphical Thresholding for best subset selection problem of Gaussian graphical model, and the core concept of this method was proposed by Luo et al. (2014) <arXiv:1407.7819>. Secondly, a closed form solution for graphical lasso under acyclic graph structure is implemented in our package (Fattahi and Sojoudi (2019) <`https://jmlr.org/papers/v20/17-501.html`>). Furthermore, we implement block coordinate descent algorithm to efficiently solve the covariance selection problem (Dempster (1972) <doi:10.2307/2528966>). Our package is computationally efficient and can solve ultra-high-dimensional problems, e.g. p > 10,000, in a few minutes.

License GPL (>= 2)

Encoding UTF-8

LazyData true

Depends R (>= 3.2)

Imports Rcpp (>= 0.12.15), MASS, Matrix

Suggests testthat, knitr, rmarkdown

VignetteBuilder knitr

LinkingTo Rcpp, RcppEigen

NeedsCompilation yes

Author Shiyun Lin [aut, cre], Jin Zhu [aut], Junxian Zhu [aut], Xueqin Wang [aut], SC2S2 [cph]

Maintainer Shiyun Lin <`linshy27@mail2.sysu.edu.cn`>

RoxygenNote 7.1.0

Repository CRAN

Date/Publication 2024-01-12 16:10:02 UTC

Contents

`ar1` *Synthetic multivariate Gaussian data*

Description

A synthetic dataset includes 200 samples under multivariate Gaussian distribution with 100 variables.

Format

ar1$x A numeric matrix with 200 rows and 100 variables where each row represents a sample.

ar1$Omega The corresponding inverse covariance matrix of the Gaussian graphical model.

Details

This synthetic dataset contains 200 samples, each of them is a vector following multivariate Gaussian distribution with 100 variables. The inverse covariance matrix of the distribution is as follows,

`Omega[i, i] = 1`.

`Omega[i, i + 1] = Omega[i, i - 1] = 0.5`.

Otherwise: `Omega[i, j] = 0`.

The corresponding graph structure is the so-called AR(1).

`ggm.generator` *Simulate Data from Gaussian Graphical Model*

Description

Produce one or more samples from the specified Gaussian graphical model.

Usage

```
ggm.generator(n, Omega)
```

Arguments

`n`	The number of samples required.
`Omega`	The inverse covariance matrix of the specified Gaussian graphical model.

Value

A numeric matrix with `n` rows and `p` variables where `p` corresponds to the dimension of `Omega`.

Note

`Omega` should be positive definite.

Examples

```
library(gif)

set.seed(1)
n <- 200
p <- 100
Omega <- diag(1, p, p)
for(i in 1:(p - 1)) {
  Omega[i, i + 1] <- 0.5
  Omega[i + 1, i] <- 0.5
}
x <- ggm.generator(n, Omega)
```

`hgt` *Hard Graphical Thresholding Algorithm*

Description

Estimates sparse inverse covariance matrix.

Usage

```
hgt(x, size, active.entry = NULL, bcd.opt = list(max.iter = 10, eps = 0.001))
```

Arguments

`x` There are 2 options: (1) `x` is an n by p data matrix; (2) a p by p sample covariance matrix. The program automatically identifies the input matrix by checking the symmetry. (n is the sample size and p is the dimension.)

`size` A non-negative integer for determining the model size, i.e., the number of non-zero off-diagonal entries in the upper-triangular precision matrix, which is also the number of edges in the graph. `size` must range from 0 to $(p^2 - p)/2$.

`active.entry` Pre-determined non-zero off-diagonal entries positions of the precision matrix. Default: `active.entry = NULL`.

`bcd.opt` A list of options that control details of the block coordinate descent algorithm.

Details

Hard Graphical Thresholding (HGT) algorithm proceeds by thresholding the sample correlation matrix and estimating the inverse covariance matrix with block coordinate descent algorithm. HGT algorithm could recover the inverse covariance matrix given model size or given active entries. When active entries are given directly, model fitting is the so-called covariance selection.

Value

A list with following components:

`Omega`	Estimated inverse covariance matrix.
`active.entry`	The position of the non-zero off-diagonal entries of `Omega` in the upper-triangular part.

Note

Either `size` or `active.entry` should be specified when function `hgt` is called. If both arguments are given, `size` would be omitted and the inverse covariance matrix would be estimated based on the given `active.entry`.

If arguments `active.entry` is specified, only one of the entries in symmetric positions should be given.

References

Luo, Shikai, Rui Song, and Daniela Witten (2014). Sure Screening for Gaussian Graphical Models. arXiv preprint arXiv:1407.7819. URL https://arxiv.org/abs/1407.7819.

Dempster, A.P. (1972). Covariance Selection. Biometrics, 28(1), 157-175. doi:10.2307/2528966

Examples

```
library(gif)

data("ar1")
p <- 100
non_zero_num <- sum(ar1[["Omega"]] != 0) - p
res <- hgt(ar1[["x"]], size = non_zero_num / 2)
```

`sgt` *Soft Graphical Thresholding Algorithm*

Description

Estimates a sparse inverse covariance matrix using the closed form solution of graphical lasso under acyclic graph structure.

Usage

```
sgt(x, lambda, size = NULL)
```

Arguments

`x` There are 2 options: (1) `x` is an n by p data matrix; (2) a p by p sample covariance matrix. The program automatically identifies the input matrix by checking the symmetry. (n is the sample size and p is the dimension.)

`lambda` The regularization parameter for graphical lasso.

`size` A non-negative integer for determining the model size, i.e., the number of non-zero off-diagonal entries in the upper-triangular precision matrix, which is also the number of edges in the graph. `size` must range from 0 to $(p^2 - p)/2$.

Details

Soft Graphical Thresholding (SGT) algorithm proceeds by thresholding the sample covariance matrix and estimating the inverse covariance matrix with a closed-form formula. If the graph structure detected by the thresholding procedure is acyclic, then the estimation is equivalent to the solution of graphical lasso.

Value

A list with following components:

`Omega` Estimated inverse covariance matrix.

`active.entry` The position of the non-zero entries of `Omega`.

`is.acyclic` The boolean flag of whether the detected graph structure is acyclic or not.

Note

Either `lambda` or `size` should specified when function `sgt` is called. If both arguments are given, only `lambda` would be considered.

References

Fattahi, Salar, and Somayeh Sojoudi. Graphical Lasso and Thresholding: Equivalence and Closed-form Solutions. Journal of Machine Learning Research 20.10 (2019): 1-44. doi: 10.5555/3322706.3322716

Examples

```
library(gif)

data("ar1")
res <- sgt(ar1[["x"]], lambda = 0.01)
```

Index