

Package ‘shock’

July 23, 2025

Type Package

Title Slope Heuristic for Block-Diagonal Covariance Selection in High Dimensional Gaussian Graphical Models

Version 1.0

Date 2015-12-24

Author Emilie Devijver, Melina Gallopin

Maintainer Melina Gallopin <melina.gallopin@gmail.com>

Description Block-diagonal covariance selection for high dimensional Gaussian graphical models. The selection procedure is based on the slope heuristics.

License GPL (>= 3)

Imports glasso, mvtnorm, capushe, GGMselect, igraph, stats

NeedsCompilation no

Repository CRAN

Date/Publication 2015-12-24 14:45:29

Contents

shock-package	2
computeLoglikeFromPartition	3
dataTest	4
networkInferenceGlassoBIC	4
shockSelect	5
simulateBlockDiagNetwork	6
thresholdAbsSPath	7
Index	8

shock-package

Slope Heuristic for Block-Diagonal Covariance Selection in High Dimensional Gaussian Graphical Models

Description

Block-diagonal covariance selection for high dimensional Gaussian graphical models. The selection procedure is based on the slope heuristics.

Details

Package: shock
Type: Package
Version: 1.0
Date: 2015-11-07
License: GPL (>= 3)

The function main function of the package (performShock) performs block-diagonal covariance selection for high-dimensional Gaussian graphical models.

Author(s)

Emilie Devijver, Melina Gallopin

Maintainer: Melina Gallopin <melina.gallopin@gmail.com>

References

Devijver, E., Gallopin, M. (2015). Block-diagonal covariance selection for high dimensional Gaussian graphical models. Inria Research Report . Available at <http://arxiv.org/abs/1511.04033>.

Examples

```
## load data to test
data(dataTest)

## dimension of the dataset expdata
n <- dim(dataTest)[1]
p <- dim(dataTest)[2]

## perform partition of variables selection
## based on the slope heuristic
resShock <- shockSelect(dataTest)

## verify that the two slope heuristic
## calibrations give the same result
```

```

table(resShock$SHDJlabels == resShock$SHRRlabels)

## collect the labels of variables
SHlabels <- resShock$SHDJlabels

## SHadjaMat: adjacency matrix of the inferred network
## Shock network inference
SHadjaMat<- diag(p)
for(itt in 1:length(unique(SHlabels))){
  stepdata <- as.matrix(dataTest[,SHlabels==itt],nrow=dim(dataTest)[1])
  if(dim(stepdata)[2]>1){
    resNet <- networkInferenceGlassoBIC(stepdata)
    SHadjaMat[SHlabels==itt,SHlabels==itt] <- resNet$A
  }
}

```

```
computeLoglikeFromPartition
```

Compute the log-likelihood of the model

Description

This function computes the log-likelihood of a multivariate Gaussian model with a block-diagonal covariance matrix.

Usage

```
computeLoglikeFromPartition(labels, expdata)
```

Arguments

labels	vector of block labels for each variable
expdata	matrix of data

Details

This function computes the log-likelihood of a multivariate Gaussian model with a block-diagonal covariance matrix described in the labels vector.

Value

loglike	loglikelihood of the model
df	degree of freedom of the model
labels	labels provided as input

Examples

```
## load data to test
data(dataTest)

## threshold of absS matrix
myLABELS <- thresholdAbsSPath(dataTest)$partitionList

## compute loglikelihood
logLikePath <- lapply(myLABELS, function(x) computeLoglikeFromPartition(x,dataTest))
```

dataTest	<i>Simulated data to test the R package</i>
----------	---

Description

This toy dataset as been simulated under a multivariate normal distribution with a block-diagonal covariance matrix and is used to test the method.

Usage

```
dataTest
```

Format

The dataset dataTest is a matrix.

networkInferenceGlassoBIC	<i>Network inference using the glasso algorithm</i>
---------------------------	---

Description

This function performs network inference using the glasso algorithm for several regularization parameters and selects a network based on the BIC of the model.

Usage

```
networkInferenceGlassoBIC(dataNet, nb.rho = 100)
```

Arguments

dataNet	matrix of data
nb.rho	number of regularization parameters to test in the glasso algorithm

Value

A	selected adjacency matrix based on BIC
Theta	selected precision matrix based on BIC
Sigma	selected covariance matrix based on BIC
penaltieslist	list of regularization parameters
pathA	list of adjacency matrices for each regularization parameter
pathTheta	list of precision matrices for each regularization parameter
pathSigma	list of covariance matrices for each regularization parameter
pathBIC	list of BIC values for each regularization parameter

References

<https://cran.r-project.org/web/packages/glasso/glasso.pdf>

Examples

```
## load data to test
data(dataTest)

## perform network inference
resNet <- networkInferenceGlassoBIC(dataTest)
```

shockSelect

Shock selection

Description

This function performs block-diagonal covariance selection for high-dimensional Gaussian graphical models.

Usage

```
shockSelect(expdata)
```

Arguments

expdata matrix of data

Value

SHDJlabels	Vector of partition labels based on the slope heuristic dimension jump
SHRRlabels	Vector of partition labels based on the slope heuristic robust regression
capusheOutput	output of the kappa coefficient calibration capushe function

Examples

```
## load data to test
data(dataTest)

## dimension of the dataset expdata
n <- dim(dataTest)[1]
p <- dim(dataTest)[2]

## perform partition of variables selection
## based on the slope heuristic
resShock <- shockSelect(dataTest)
```

```
simulateBlockDiagNetwork
```

Simulate a modular network

Description

This function simulates a modular network with p variables based on the partition of variables into blocks labels.

Usage

```
simulateBlockDiagNetwork(p, labels)
```

Arguments

<code>p</code>	number of variables in the network
<code>labels</code>	vector indicating the partition of variables into blocks

Details

To simulate covariance matrices, we use the methodology detailed in Giraud, S. Huet, and N. Verze-len. Graph selection with GGMselect. 2009

<http://fr.arxiv.org/abs/0907.0619> <https://cran.r-project.org/package=GGMselect>

Value

<code>A</code>	simulated adjacency matrix
<code>C</code>	simulated correlation matrix
<code>Pcor</code>	simulated partial correlation matrix
<code>labels</code>	vector indicating the partition of variables into blocks provided as input of the function

Examples

```
## number of variables
p <- 100
## number of blocks
K <- 15
## vector of partition into blocks
labels <- factor(rep(1:K, length.out=p))
## simulate network
g <- simulateBlockDiagNetwork(p,labels)
```

thresholdAbsSPath	<i>Detect partitions of variables into blocks.</i>
-------------------	--

Description

This function returns a list of partitions of variables based on the sample covariance matrix for several levels of threshold.

Usage

```
thresholdAbsSPath(expdata)
```

Arguments

expdata	matrix of data
---------	----------------

Value

partitionList	list of partitions of variables (vectors) deduced by thresholding the sample covariance matrix
lambdaPath	list of threshold parameters

Examples

```
## load data to test
data(dataTest)

## detect partitions of variables into blocks based on the sample covariance matrix
partitions <- thresholdAbsSPath(dataTest)$partitionList
```

Index

`computeLoglikeFromPartition`, [3](#)

`dataTest`, [4](#)

`networkInferenceGlassoBIC`, [4](#)

`shock (shock-package)`, [2](#)

`shock-package`, [2](#)

`shockSelect`, [5](#)

`simulateBlockDiagNetwork`, [6](#)

`thresholdAbsSPath`, [7](#)