Package 'GGMselect'

May 1, 2022

Type Package

Title Gaussian Graphs Models Selection

Version 0.1-12.5

Date 2022-05-01

Author Annie Bouvier, Christophe Giraud, Sylvie Huet, Verzelen N.

Maintainer Benjamin Auder <benjamin.auder@universite-paris-saclay.fr>

Description Graph estimation in Gaussian Graphical Models, following the method developed by C. Giraud, S. Huet and N. Verzelen (2012) <doi:10.1515/1544-6115.1625>. The main functions return the adjacency matrix of an undirected graph estimated from a data matrix.

Imports lars, gtools, stats

Depends R (>= 3.5.0), mvtnorm

Suggests network, glasso

License GPL (>= 3)

URL https://CRAN.R-project.org/package=GGMselect

NeedsCompilation yes

Repository CRAN

Date/Publication 2022-05-01 15:10:05 UTC

R topics documented:

GGMselect-package 2
convertGraph 3
penalty 4
selectFast 5
selectMyFam 7
selectQE 8
simulateGraph 10

Index **12**

GGMselect-package *Gaussian Graphs Models selection*

Description

GGMselect is a package dedicated to graph estimation in Gaussian Graphical Models. The main functions return the adjacency matrix of an undirected graph estimated from a data matrix.

This package is developed in the Applied Mathematics and Informatics (https://maiage.inrae.fr/) Lab of INRA - Jouy-en-Josas, France.

To cite GGMselect, please use `citation("GGMselect")`.

Details

Package:	GGMselect
URL:	https://CRAN.R-project.org/package=GGMselect

Author(s)

Annie Bouvier, Christophe Giraud, Sylvie Huet, Verzelen Nicolas

Maintainer: Benjamin Auder <benjamin.auder@universite-paris-saclay.fr>

More details are available on ../doc/Notice.pdf

References

Please use `citation("GGMselect")`.

See Also

selectFast, selectQE, selectMyFam,convertGraph, simulateGraph, penalty

Examples

```
p=30
n=30
# simulate graph
eta=0.11
Gr <- simulateGraph(p,eta)
# simulate data
X <- rmvnorm(n, mean=rep(0,p), sigma=Gr$C)
# estimate graph
## Not run: GRest <- selectFast(X)
```

```
# plot result
## Not run: library(network)
## Not run: par(mfrow=c(1,2))
## Not run: gV <- network(Gr$G)
## Not run: plot(gV,jitter=TRUE, usearrows = FALSE, label=1:p,displaylabels=TRUE)
## Not run: g <- network(GRest$EW$G)
## Not run: plot(g, jitter=TRUE, usearrows = FALSE,  label=1:p,displaylabels=TRUE)
```

---

convertGraph *Convert graphs into adjacency matrices*

Description

Convert into adjacency matrices NG graphs (expressed as lists of connected nodes)

Usage

```
convertGraph(Graph)
```

Arguments

Graph	array of dimension p x Dmax x NG, where Dmax is the degree of the graph and NG the number of graphs. If NG is equal to 1, Graph can be a matrix of dimension p x Dmax. Graph[a,,iG] should be the indices of the nodes connected to the node a, for the graph iG; Graph[a,1,iG] should be equal to 0 if there is no node connected to the node a.

Value

An array of dimension p x p x NG, or, when NG is equal to 1, a matrix of dimension p x p.

The entry [,,iG] is a symmetric matrix, with diagonal equal to zero. The entry [a,b,iG] is equal to 1 if a is connected to b, 0 otherwise.

Note

This function is useful to generate the entry MyFamily of the function selectMyFam. Actually, the list of adjacency matrices MyFamily can be generated from lists of connected nodes with convertGraph.

Author(s)

Bouvier A, Giraud C, Huet S, Verzelen N

References

Please use citation("GGMselect")

See Also

selectQE, selectMyFam, selectFast, simulateGraph, penalty

Examples

```
p=30
n=30
# simulate graph
eta=0.11
Gr <- simulateGraph(p,eta)
X <- rmvnorm(n, mean=rep(0,p), sigma=Gr$C)
# estimate graph
GRest <- selectFast(X, family="C01")
# Neighb and G are 2 forms of the same result
a <- convertGraph(GRest$C01$Neighb)
print(all.equal(a, GRest$C01$G)) # TRUE
# recalculate the graph with selectMyFam
GMF <- selectMyFam(X, list(a))
print(all.equal(a,GMF$G)) # TRUE
```

---

penalty *Penalty function*

Description

Compute the penalty function of GGMselect.

Usage

```
penalty(p,n, dmax=min(3,n-3,p-1), K=2.5)
```

Arguments

p	the number of variables. p should be greater than 1.
n	the sample size. n should be greater than 3.
dmax	integer or p-dimensional vector of integers smaller or equal to min(n-3, p-1). When dmax is a scalar, it gives the maximum degree of the estimated graph. When dmax is a vector, dmax[a] gives the maximum degree of the node a. Default value: min(3,n-3,p-1).
K	scalar or vector of real numbers larger than 1. Tuning parameter of the penalty function.

Details

More details are available on ../doc/Notice.pdf

Value

A matrix of dimension `(max(Dmax)+1) x length(K)`. The entry `[d+1,k]` gives the value of the penalty for the dimension `d` and the parameter `K[k]`.

Author(s)

Bouvier A, Giraud C, Huet S, Verzelen N

References

Please use `citation("GGMselect")`

See Also

`selectQE`, `selectMyFam`, `selectFast`, `simulateGraph`, `convertGraph`

Examples

```
p=30
n=30
pen <- penalty(p,n, 3)
```

`selectFast` *Estimate a graph in a Gaussian Graphical Model: Fast procedure*

Description

Select a graph within the (data-driven) families of graphs `EW`, `C01`, and `LA`.

Usage

```
selectFast(X, dmax=min(floor(nrow(X)/3),nrow(X)-3,ncol(X)-1),
           K=2.5, family="EW",
           min.ev=10**(-8), max.iter=200, eps=0.01,
        beta=nrow(X)*nrow(X)/2, tau=1/sqrt(nrow(X)*(ncol(X)-1)), h=0.001, T0=10,
           verbose=FALSE )
```

Arguments

`X`	`n` x `p` matrix where `n` is the sample size and `p` the number of variables. `n` should be greater than `3` and `p` greater than `1`.
`dmax`	integer or `p`-dimensional vector of integers smaller or equal to `min(n-3, p-1)`. When `dmax` is a scalar, it gives the maximum degree of the estimated graph. When `dmax` is a vector, `dmax[a]` gives the maximum degree of the node `a`.
`K`	scalar or vector with values greater than 1. Tuning parameter of the penalty function.

family	character string or vector of character strings, among "EW", "LA", "C01", c("C01","LA") or c("C01","LA","EW").
min.ev	minimum eigenvalue for matrix inversion.
max.iter, eps, beta, tau, h, T0	tuning parameters for the Langevin Monte Carlo algorithm. Only used when family is "EW" or c("C01","LA","EW").
verbose	logical. If TRUE a trace of the current process is displayed in real time.

Details

More details are available on ../doc/Notice.pdf

Value

A list with components "EW", "LA", "C01", "C01.LA" and "C01.LA.EW", according to the family argument, each one with components:

Neighb	array of dimension p x max(dmax) x length(K) or, when length(K) equals 1, matrix of dimension p x max(dmax). Neighb[a, , k] contains the indices of the nodes connected to node a for K[k].
crit.min	vector of dimension length(K). It gives the minimal values of the selection criterion for each value of K
G	array of dimension p x p x length(K) or, when length(K) equals 1, matrix of dimension p x p. G[,,k] gives the adjacency matrix for K[k].

Author(s)

Bouvier A, Giraud C, Huet S, Verzelen N.

References

Please use citation("GGMselect").

See Also

selectQE, selectMyFam, simulateGraph, penalty, convertGraph

Examples

```
p=30
n=30
# simulate graph
eta=0.11
Gr <- simulateGraph(p,eta)
# simulate data
X <- rmvnorm(n, mean=rep(0,p), sigma=Gr$C)
# estimate graph
GRest <- selectFast(X, family="C01")

# plot result
```

```
library(network)
par(mfrow=c(1,2))
gV <- network(Gr$G)
plot(gV,jitter=TRUE, usearrows = FALSE, label=1:p,displaylabels=TRUE)
g <- network(GRest$C01$G)
plot(g, jitter=TRUE, usearrows = FALSE,  label=1:p,displaylabels=TRUE)
```

`selectMyFam` *Select a graph within a given family of graphs in Gaussian Graphical Modeling.*

Description

Select a graph within a given family of graphs.

Usage

```
selectMyFam(X, MyFamily, K=2.5, min.ev=10**(-8))
```

Arguments

`X`	n x p matrix where n is the sample size and p the number of variables. n should be greater than 3 and p greater than 1.
`MyFamily`	list of pxp adjacency matrices corresponding to graphs with degree less or equal to n-3.
`K`	scalar or vector with values larger than 1. Tuning parameter of the penalty function.
`min.ev`	minimum eigenvalue for matrix inversion.

Details

More details are available on `../doc/Notice.pdf`

Value

`Neighb`	array of dimension p x dmax x length(K) where dmax is the maximum degree of the graphs in MyFamily. When K is of length 1, matrix of dimension p x dmax. Neighb[a, , k] contains the indices of the nodes connected to node a for K[k].
`crit.min`	vector of dimension length(K). The minimal values of the selection criterion for each value of K.
`ind.min`	vector of dimension length(K). Indices of the families for which the criterion is minimum.
`G`	array of dimension p x p x length(K) or, when length(K) equals 1, matrix of dimension p x p. G[,,k] gives the adjacency matrix for K[k].

Note

Adjacency matrices can be generated from lists of connected nodes by using the function convertGraph

Author(s)

Bouvier A, Giraud C, Huet S, Verzelen N.

References

Please use citation("GGMselect").

See Also

selectFast, selectQE, simulateGraph, penalty, convertGraph

Examples

```
p=30
n=30
# generate graph
eta=0.11
Gr <- simulateGraph(p,eta)
# generate data
X <- rmvnorm(n, mean=rep(0,p), sigma=Gr$C)
# generate a family of candidate graphs with glasso
library("glasso")
MyFamily <- NULL
for (j in 1:3){
  MyFamily[[j]] <- abs(sign(glasso(cov(X),rho=j/5)$wi))
  diag(MyFamily[[j]]) <- 0
}
# select a graph within MyFamily
GMF <- selectMyFam(X,MyFamily)
# plot the result
library(network)
par(mfrow=c(1,2))
gV <- network(Gr$G)
plot(gV,jitter=TRUE, usearrows = FALSE, label=1:p,displaylabels=TRUE)
gMyFam <- network(GMF$G)
plot(gMyFam, jitter=TRUE, usearrows = FALSE,  label=1:p,displaylabels=TRUE)
```

---

selectQE *Estimate a graph in a Gaussian Graphical Model: Quasi Exhaustive search*

Description

Select a graph within the family of graphs QE

Usage

```
selectQE(X, dmax=min(3,nrow(X)-3,ncol(X)-1), K=2.5,
   min.ev=10**(-8), max.iter=10**6, max.nG=10**8, max.size=10**8,
   verbose=FALSE)
```

Arguments

X	n x p matrix where n is the sample size and p the number of variables. n should be greater than 3 and p greater than 1.
dmax	integer or p-dimensional vector of integers smaller or equal to min(n-3, p-1). When dmax is a scalar, it gives the maximum degree of the estimated graph. When dmax is a vector, dmax[a] gives the maximum degree of the node a.
K	scalar or vector with values greater than 1. Tuning parameter in the penalty function.
min.ev	minimum eigenvalue for matrix inversion.
max.iter	integer. Maximum number of stepwise iterations.
max.nG	integer. Maximum number of graphs considered in the exhaustive search. Stepwise procedure beyond.
max.size	integer. Maximum number of calculations of the residuals sums of squares. Execution stopped beyond.
verbose	logical. If TRUE a trace of the current process is displayed in real time.

Details

More details are available on ../doc/Notice.pdf

Value

Neighb	array of dimension p x max(dmax) x length(K) or, when length(K) equals 1, matrix of dimension p x max(dmax). Neighb[a, , k] contains the indices of the nodes connected to node a for K[k].
crit.min	vector of dimension length(K). The minimal values of the selection criterion for each value of K.
G	array of dimension p x p x length(K) or, when length(K) equals 1, matrix of dimension p x p. G[, ,k] gives the adjacency matrix for K[k].

Author(s)

Bouvier A, Giraud C, Huet S, Verzelen N.

References

Please use citation("GGMselect").

See Also

selectFast, selectMyFam, simulateGraph, penalty, convertGraph

Examples

```
p=30
n=30
# simulate graph
eta=0.11
Gr <- simulateGraph(p,eta)
# simulate data
X <- rmvnorm(n, mean=rep(0,p), sigma=Gr$C)
# estimate graph
## Not run: GQE <- selectQE(X)

# plot the result
## Not run: library(network)
## Not run: par(mfrow=c(1,2))
## Not run: gV <- network(Gr$G)
## Not run: plot(gV,jitter=TRUE, usearrows = FALSE, label=1:p,displaylabels=TRUE)
## Not run: gQE <- network(GQE$G)
## Not run: plot(gQE, jitter=TRUE, usearrows = FALSE,  label=1:p,displaylabels=TRUE)
```

`simulateGraph` *Generate sparse Gaussian Graphical Models*

Description

Generate random covariance matrices `C` with sparse inverse. The Gaussian law `N(0,C)` is then a sparse (non-uniform) Gaussian Graphical Model.

Usage

```
simulateGraph(p, eta, extraeta = eta/5)
```

Arguments

`p`	integer. Number of rows and columns of `C`. Should be greater than `1`.
`eta`	real number in (0,1). Proportion of edges in subgroups. Small values of `eta` give sparse graphs.
`extraeta`	real number in (0,1). Proportion of edges inter groups.

Details

More details are available on `../doc/Notice.pdf`

Value

`G`	p x p matrix. Adjacency matrix of the graph.
`Dmax`	integer. Maximum degree of the graph.

Neighb	array of dimension p x Dmax. Neighb[a,] contains the indices of the nodes connected to node a.
Nnodes	integer. Number of nodes.
C	p x p matrix. Covariance matrix.
PCor	p x p matrix. Partial correlation matrix.

Author(s)

Bouvier A, Giraud C, Huet S, Verzelen N

References

Please use `citation("GGMselect")`.

See Also

selectQE, selectMyFam, selectFast, penalty, convertGraph

Examples

```
# simulate a graph
p=30
eta=0.13
Gr <- simulateGraph(p,eta)

# plot the graph
library(network)
par(mfrow=c(1,1))
gV <- network(Gr$G)
plot(gV,jitter=TRUE, usearrows = FALSE, label=1:p,displaylabels=TRUE)
```

Index

∗ **datagen**
simulateGraph, 10
∗ **graphs**
convertGraph, 3
GGMselect-package, 2
penalty, 4
selectFast, 5
selectMyFam, 7
selectQE, 8
simulateGraph, 10
∗ **models**
GGMselect-package, 2
penalty, 4
selectFast, 5
selectMyFam, 7
selectQE, 8
simulateGraph, 10
∗ **multivariate**
convertGraph, 3
GGMselect-package, 2
penalty, 4
selectFast, 5
selectMyFam, 7
selectQE, 8
simulateGraph, 10
∗ **package**
GGMselect-package, 2

convertGraph, *2*, 3, *5, 6*, *8, 9*, *11*

GGMselect *(*GGMselect-package*)*, 2
GGMselect-package, 2

penalty, *2*, *4*, 4, *6*, *8, 9*, *11*

selectFast, *2*, *4, 5*, 5, *8, 9*, *11*
selectMyFam, *2–6*, 7, *9*, *11*
selectQE, *2*, *4–6*, *8*, 8, *11*
simulateGraph, *2*, *4–6*, *8, 9*, 10