Package ‘cglasso’

September 15, 2018

Version 1.0.0
Date 2018-09-09
Type Package
Title L1-Penalized Censored Gaussian Graphical Models
Author Luigi Augugliaro
Maintainer Luigi Augugliaro <luigi.augugliaro@unipa.it>
Depends R (>= 3.5), igraph
Description The l1-penalized censored Gaussian graphical model (cglasso) is an extension of the graphical lasso estimator developed to handle datasets with censored observations. An EM-like algorithm is implemented to estimate the parameters of the censored Gaussian graphical models.
Imports methods, MASS
License GPL (>= 2)
LazyLoad yes
NeedsCompilation yes
Repository CRAN
Date/Publication 2018-09-14 23:22:15 UTC

R topics documented:
cglasso-package .................................................. 2
aic ................................................................. 3
cglasso .......................................................... 5
coeff ............................................................. 9
datacggm ......................................................... 11
ebic ............................................................. 13
event ............................................................ 16
glasso .......................................................... 17
loglik ........................................................... 21
mle ............................................................... 23
plot ............................................................ 27
The $\ell_1$-penalized censored Gaussian graphical model (Augugliaro and other, 2018) is an extension of the graphical lasso estimator (Yuan and other, 2007) developed to handle datasets from a censored Gaussian graphical model. An EM-like algorithm is implemented to fit the model. The graphical lasso algorithm (Friedman and other, 2008) is used to solve the maximization problem in the M-step.

**Details**

Package: cglasso
Type: Package
Version: 1.0.0
Date: 2018-09-09
License: GPL (>=2)

**Author(s)**

Luigi Augugliaro
Maintainer: Luigi Augugliaro <luigi.augugliaro@unipa.it>

**References**


Description

‘aic’ computes the ‘Akaike Information Criterion’ whereas ‘bic’ computes the ‘Bayesian Information Criterion’.

Usage

```r
aic(object, k = 2)
bic(object)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>object</td>
<td>an object with class ‘glasso’, ‘ggm’, ‘cglasso’ or ‘cggm’.</td>
</tr>
<tr>
<td>k</td>
<td>the penalty per parameter to be used; the default k = 2 is the classical AIC.</td>
</tr>
</tbody>
</table>

Details

The measure of goodness-of-fit (gof) returned by the functions ‘aic’ and ‘bic’ depends on the class of the fitted model.

If ‘object’ has class ‘glasso’ or ‘ggm’, then ‘aic’ computes the following measure of goodness-of-fit:

\[-2 \log\text{-likelihood} + k \text{df},\]

where \(k\) is the penalty per parameter and \(\text{df}\) represents the number of parameters in the fitted model. The values of the log-likelihood function are computed using the function `loglik`. The usual Akaike Information Criterion (AIC) is computed letting \(k = 2\) (default value of the function ‘aic’) whereas the ‘Bayesian Information Criterion’ (BIC) is computed letting \(k = \log(n)\), where \(n\) is the sample size.

If ‘object’ has class ‘cglasso’ or ‘cggm’, then ‘aic’ computes the following measure of goodness-of-fit:

\[-2 Q\text{-function} + k \text{df},\]

in other words the log-likelihood is replaced with the \(Q\)-function maximized in the M-step of the EM-like algorithm described in `cglasso` and `mle`. This measure of goodness-of-fit was proposed in Ibrahim and others (2008) for statistical model with missing-data.

‘aic’ and ‘bic’ return an object with S3 class ‘gof’ for which are available the method functions ‘print.gof’ and ‘plot.gof’. These method functions are developed with the aim of helping the user in finding the optimal value of the tuning parameter, defined as the \(\rho\)-value minimizing the chosen measure of goodness-of-fit. For this reason, ‘print.gof’ shows also the ranking of the fitted models (the best model is pointed out with an arrow) whereas ‘plot.gof’ pointed out the optimal \(\rho\)-value by a vertical dashed line (see below for some examples).
Value

`'aic'` and `'bic'` return an object with S3 class “gof”, i.e. a list containing the following components:

- `value_gof`: the values of the measure of goodness-of-fit used to evaluate the fitted models.
- `rho`: the values of the tuning parameter used to fit the model.
- `value`: the values of the log-likelihood function or the Q-function.
- `df`: the number of the estimated non-zero parameters, i.e. the number of non-zero partial correlations plus 2p.
- `n`: the sample size.
- `p`: the number of variables.
- `model`: the name of the fitted models.
- `type`: the measure of goodness-of-fit used to evaluate the fitted models.

Author(s)

Luigi Augugliaro (<luigi.augugliaro@unipa.it>)

References


See Also

`loglik`, `cglasso`, `glasso`, `mle`, `ebic` and the method functions `plot` and `summary`.

Examples

```r
library("cglasso")
set.seed(123)

# cglasso model #
#
# n <- 100L
# p <- 5L
# mu <- rep.int(0L, times = p)
# X <- rdatacgmm(n = n, mu = mu, prob = 0.05)
# out <- cglasso(X = X)
# out_aic <- aic(out)
# out_aic
# plot(out_aic)
#
# out_bic <- bic(out)
# out_bic
# plot(out_bic)
```
cglasso

Censored Graphical Lasso Estimator

Description

‘cglasso’ function is used to fit an l1-penalized censored Gaussian graphical model.

Usage

cglasso(X, lo, up, weights, nrho = 50, rho.min.ratio, rho, maxR2, 
maxit_em = 1.0E+3, thr_em = 1.0e-4, maxit_bcd = 10000, 
thr_bcd = 1.0e-4, trace = 0L)
Arguments

**X**
an object with S3 class `datacggm`, usually the output of the function `datacggm`. Optionally, this argument can be a matrix of dimension \( n \times p \); in this case, the matrix `X` and the arguments `lo` and `up` are passed to `datacggm` to create the object with class `datacggm`.

**lo**
optional argument. If the argument `X` is a matrix then `lo` is used to create an object with class `datacggm`.

**up**
optional argument. If the argument `X` is a matrix then `up` is used to create an object with class `datacggm`.

**weights**
an optional symmetric matrix of non-negative weights. This matrix can be used to specify the unpenalized partial correlation coefficients (`weights[i, j] = 0`) or the structural zeros in the precision matrix (`weights[i, j] = +Inf`). See below for an example. By default, cglasso model is fitted without weights.

**nrho**
the integer specifying the number of tuning parameter used to fit the cglasso model. Default is `nrho = 50`.

**rho.min.ratio**
the smallest value for the tuning parameter \( \rho \), as a fraction of the smallest tuning parameter for which all the estimated partial correlation coefficients are zero. The default depends on the sample size `n` relative to the number of variables `p`. If `p < n`, the default is `1.0E-4` otherwise the value `1.0E-2` is used as default. A very small value of `rho.min.ratio` will lead to a saturated fitted model in the `p<n` case.

**rho**
optional argument. A user supplied rho sequence. WARNING: avoid supplying a single value for the tuning parameter; supply instead a decreasing sequence of \( \rho \)-values.

**maxR2**
a value belonging to the interval \([0, 1]\) specifying the largest value of the pseudo R-squared measure (see Section Details). The regularization path is stopped when \( R^2 \) exceeds `maxR2`. Default depends on the sample size `n` relative to the number of variables `p`. If `p < n`, the default is `1` otherwise the value `0.9` is used as default.

**maxit_em**
maximum number of iterations of the EM algorithm. Default is `1.0E+3`.

**thr_em**
threshold for the convergence of the EM algorithm. Default value is `1.0E-4`.

**maxit_bcd**
maximum number of iterations of the glasso algorithm. Default is `1.0E+4`.

**thr_bcd**
threshold for the convergence of the glasso algorithm. Default is `1.0E-4`.

**trace**
integer for printing out information as iterations proceed: `trace = 0` no information is printed out on video; `trace = 1` basic information is printed out on video; `trace = 2` detailed information is printed out on video.

Details

The censored graphical lasso (cglasso) estimator (Augugliaro and other, 2018) is an extension of the classical graphical lasso (glasso) estimator (Yuan and other, 2007) developed to fit a sparse censored Gaussian graphical model (see Section 2 in Augugliaro and other (2018) for a formal definition).

cglasso function fits the model using the following EM-like algorithm:
Step Description
1. Let \( \{ \hat{\mu}_\rho^{ini}; \hat{\Theta}_\rho^{ini} \} \) be initial estimates;
2. **E-step**
   use the moments of the truncated normal distribution to compute the current estimates of the marginal means, denoted by \( \bar{x}_\rho \), and to complete the empirical covariance matrix \( S_\rho \);
3. **M-step**
   let \( \hat{\mu}_\rho = \bar{x}_\rho \);
   compute \( \hat{\Theta}_\rho \) using \( S_\rho \) and the glasso algorithm (Friedman and other);
4. repeat steps 2. and 3. until a convergence criterion is met. See Section 3.2 in Augugliaro and other (2018) for more details.

In order to reduce the computational burden of the algorithm, in Step 2. the matrix \( S_\rho \) is approximated using the method proposed in Guo and others (2015).

In order to avoid the overfitting of the model, we use the following pseudo R-squared measure:

\[
R^2 = 1 - \frac{||S_\rho - \hat{\Sigma}_\rho||_F}{||S_{\rho_{\text{max}}} - \hat{\Sigma}_{\rho_{\text{max}}}||_F},
\]

where \( ||\cdot||_F \) denotes the Frobenius norm and \( \rho_{\text{max}} \) denotes the smallest value of the tuning parameter for which all the estimated partial correlation coefficients are zero. By straightforward algebra, it is easy to show that the proposed pseudo R-squared belongs to the closed interval \([0, 1]\): \( R^2 = 0 \) when the tuning parameter is equal to \( \rho_{\text{max}} \) and \( R^2 = 1 \) when \( \rho = 0 \). The regularization path is stopped when \( R^2 \) exceeds the threshold specified by ‘maxR2’.

**Value**

cglasso returns an object with S3 class “cglasso”, i.e., a list containing the following components:

- **call**
  the call that produced this object.
- **x**
  the object with S3 class ‘dataCGGM’ used to fit the cglasso model.
- **weights**
  the weights used to fit the cglasso model.
- **xm**
  the \( p \)-dimensional vector reporting the estimates of the marginal expected values under the assumption that the precision matrix is diagonal.
- **vm**
  the \( p \)-dimensional vector reporting the estimates of the marginal variances under the assumption that the precision matrix is diagonal.
- **nrho**
  the number of fitted cglasso model.
- **rho.min.ratio**
  the scale factor used to compute the smallest value of the tuning parameter.
- **rho**
  the \( p \)-dimensional vector reporting the values of the tuning parameter used to fit the cglasso model.
- **maxR2**
  the threshold value used to stop the regularization path.
- **maxit_em**
  the maximum number of iterations of the EM algorithm.
- **thr_em**
  the threshold for the convergence of the EM algorithm.
- **maxit_bcd**
  the maximum number of iterations of the glasso algorithm.
- **thr_bcd**
  the threshold for the convergence of the glasso algorithm.
Xipt is an array of dimension $n \times p \times nrho$. $Xipt[, , k]$ is the matrix where the censored values are replaced using the conditional expected values computed in the E-step of the algorithm described in section Details.

S is an array of dimension $p \times p \times nrho$. $S[, , k]$ is the matrix $S^\rho$ used to fit the cglasso model in the M-step of the algorithm described in section Details.

mu is a matrix of dimension $p \times nrho$. The $k$th column is the estimate of the expected values of the cglasso model fitted using $\rho[k]$.

Sgm is an array of dimension $p \times p \times nrho$. $Sgm[, , k]$ is the estimate of the covariance matrix of the cglasso model fitted using $\rho[k]$.

Tht is an array of dimension $p \times p \times nrho$. $Tht[, , k]$ is the estimate of the precision matrix of the cglasso model fitted using $\rho[k]$.

Adj is an array of dimension $p \times p \times nrho$. $Adj[, , k]$ is the adjacency matrix associated to $Tht[, , k]$, i.e. $Adj[i, j, k] = 1$ if $Tht[i, j, k] \neq 0$ and 0 otherwise.

df is the $p$-dimensional vector reporting the number of non-zero partial correlation coefficients.

R2 is the $p$-dimensional vector reporting the values of the measure $R^2$ described in section Details.

ncomp is the $p$-dimensional vector reporting the number of connected components (for internal purposes only).

Ck is the $(p \times nrho)$-dimensional matrix encoding the connected components (for internal purposes only).

Pk is the $(p \times nrho)$-dimensional matrix reporting the number of vertices per connected component (for internal purposes only).

nit is the $(nrho \times 2)$-dimensional matrix reporting the number of iterations.

conv is a description of the error that has occurred.

subrout is the name of the Fortran subroutine where the error has occurred (for internal debug only).

trace is the integer used for printing out information.

Author(s)

Luigi Augugliaro (<luigi.augugliaro@unipa.it>)

References


See Also
datacgmm, glasso, to_graph, mle and the method functions summary, coef, plot, aic, bic, ebic.

Examples

library("cglasso")
set.seed(123)

p <- 5L
n <- 100L
mu <- rep(0L, p)
Tht <- diag(p)
diag(Tht[-1L, -p]) <- diag(Tht[-p, -1L]) <- 0.3
Sgm <- solve(Tht)
X <- rdatacgmm(n = n, mu = mu, Sigma = Sgm, prbr = 0.05)
out <- cglasso(X = X)
out

# in this example we use the argument 'weights' to specify
# the unpenalized partial correlation coefficients and the
# structural zeros in the precision matrix

w <- rep(1L, p * p)
dim(w) <- c(p, p)

# specifying the unpenalized partial correlation coefficients
diag(w) <- diag(w[-1L, -p]) <- diag(w[-p, -1L]) <- 0L

# specifying the structural zeros
w[1L, 4L:5L] <- w[4L:5L, 1L] <- +Inf
w[2L, 5L] <- w[5L, 2L] <- +Inf
w

out <- cglasso(X = X, weights = w)

# checking structural zeros
out$Tht[, , out$nrho][w == +Inf]

# checking stationarity conditions of the MLE estimators
# (the unpenalized partial correlation coefficients)
(out$Sgm[, , out$nrho] - out$S[, , out$nrho])[w == 0]

---

**coef** 

*Extract Model Coefficients*

**Description**

‘coef’ extracts model coefficients from a fitted model.
Usage

```r
## S3 method for class 'glasso'
coef(object, nrho = 1L, type = c("theta", "sigma"), print.info = FALSE,
digits = 3L, ...)
## S3 method for class 'cglasso'
coef(object, nrho = 1L, type = c("theta", "sigma", "mu"), print.info = FALSE,
digits = 3L, ...)
```

Arguments

- **object**: an object of class 'glasso', 'gmm', 'cglasso' or 'cggm'.
- **nrho**: integer used to specify the model from which to extract the coefficients. Default is nrho = 1.
- **type**: a string specifying the returned parameters. If 'object' has class 'glasso', the user can choice between the precision matrix ('type = "theta"') and the covariance matrix ('type = "sigma"'). For an 'object' with class 'cglasso', the user can also extract the estimates of the expected values ('type = "mu"'). Default is "theta".
- **print.info**: flag specifying if information about the model is printed out. Default is FALSE.
- **digits**: the minimum number of significant digits to be used. Default is 3L.
- **...**: additional argument added for backward compatibility with the generic function `coef`.

Details

By default, the method functions `coef.glasso` and `coef.cglasso` return the parameters specified by the argument 'type'.

If `print.info = TRUE` then the estimated parameters are silently returned and information about the chosen model is printed out, i.e. the value of the tuning parameter, the value of the pseudo R-squared, the number of connected components and the number of vertices per connected component. Furthermore, to improve the readability of the results the estimates are printed out taken into account the connected components (see the examples below).

Value

Coefficients extracted from 'object' are returned.

Author(s)

Luigi Augugliaro (<luigi.augugliaro@unipa.it>)

See Also

glasso, cglasso and mle.
Examples

library("cglasso")

# cglasso model
set.seed(123)
p <- 5L
n <- 100L
mu <- rep(0L, p)
Tht <- diag(p)
diag(Tht[-1L, -pl]) <- diag(Tht[-p, -1L]) <- 0.3
Sgm <- solve(Tht)
X <- rdatacggm(n = n, mu = mu, Sigma = Sgm, prob = 0.05)
out <- cglasso(X = X)

coef(out, nrho = 3L, type = "theta", print.info = TRUE)
Tht_hat <- coef(out, nrho = 3L, type = "theta")
Tht_hat

coef(out, nrho = 3L, type = "sigma", print.info = TRUE)
Sgm_hat <- coef(out, nrho = 3L, type = "sigma")
Sgm_hat

coef(out, nrho = 3L, type = "mu", print.info = TRUE)
mu_hat <- coef(out, nrho = 3L, type = "mu")
mu_hat

# glasso model
X <- MASS::mvrnorm(n = n, mu = mu, Sigma = Sgm)
out <- glasso(X = X)

coef(out, nrho = 3L, type = "theta", print.info = TRUE)
Tht_hat <- coef(out, nrho = 3L, type = "theta")
Tht_hat

coef(out, nrho = 3L, type = "sigma", print.info = TRUE)
Sgm_hat <- coef(out, nrho = 3L, type = "sigma")
Sgm_hat

datacggm

Create a Dataset from a Censored Gaussian Graphical Model

Description

'datacggm' function is used to create a dataset from a censored Gaussian graphical model.

Usage

datacggm(X, lo, up)
Arguments

X  a \((n \times p)\)-dimensional matrix; each row is an observation from a censored Gaussian graphical model with censoring vectors \(\text{lo}\) and \(\text{up}\).

\(\text{lo}\)  the lower censoring vector; \(\text{lo}[j]\) is used to specify the lower censoring value for the random variable \(X_j\).

\(\text{up}\)  the upper censoring vector; \(\text{up}[j]\) is used to specify the upper censoring value for the random variable \(X_j\).

Details

The function ‘\(\text{datacggm}\)’ returns a named list with class ‘\(\text{datacggm}\)’ containing the elements needed to fit a censored graphical lasso (cglasso) model. In output, the matrix \(X\) is ordered according to the pattern of censoring values.

There are specific method functions developed to help the user to deal with the censored values. The ‘\(\text{print.datacggm}\)’ method function print out the left and right-censored values using the following rules: a right-censored value is labeled adding the symbol ‘+’ at the end of the value, whereas the symbol ‘−’ is used for the left-censored values (see examples below). The summary statistics about the censored values can be obtained using the method function ‘\(\text{summary.datacggm}\)’.

Finally, the status indicator matrix, denoted by \(R\), can be obtained by the function \(\text{event}\). The elements of this matrix specify the status of an observation as follows:

- ‘\(R[i, j] = 0\)’ means that the \(i\)th observation of the \(j\)th random variable is observed;
- ‘\(R[i, j] = -1\)’ means that the \(i\)th observation of the \(j\)th random variable is left-censored;
- ‘\(R[i, j] = +1\)’ means that the \(i\)th observation of the \(j\)th random variable is right-censored.

Value

‘\(\text{datacggm}\)’ returns an object with S3 class ‘\(\text{datacggm}\)’, i.e. a list containing the following components:

\(X\)  the \((n \times p)\)-dimensional matrix \(X\) ordered according to the patterns of censored values.

\(\text{lo}\)  the lower censoring vector.

\(\text{up}\)  the upper censoring vector.

\(R\)  the augmented status indicator matrix encoding the patterns of censored values (for internal purposes only); the status indicator matrix is returned by function \(\text{event}\).

\(\text{startmis}\)  the row of the matrix \(X\) where are starting the patterns of censored values (for internal purposes only).

Author(s)

Luigi Augugliaro (<luigi.augugliaro@unipa.it>)
References


See Also

event, rdatacggm, cglasso and the method function summary.datacggm.

Examples

```r
set.seed(123)
library("cglasso")

# a dataset from a left-censored Gaussian graphical model
n <- 100L
p <- 5L
X <- matrix(rnorm(n * p), n, p)
lo <- -1
X[X <= lo] <- lo
X <- datacggm(X, lo = lo)
X

# a dataset from a right-censored Gaussian graphical model
n <- 100L
p <- 5L
X <- matrix(rnorm(n * p), n, p)
up <- 1
X[X >= up] <- up
X <- datacggm(X, up = up)
X

# a dataset from a censored Gaussian graphical model
n <- 100L
p <- 5L
X <- matrix(rnorm(n * p), n, p)
up <- 1
lo <- -1
X[X >= up] <- up
X[X <= lo] <- lo
X <- datacggm(X, lo = lo, up = up)
X
```

---

**ebic**

Extended Bayesian Information Criterion

Description

`ebic` function computes the extended Bayesian Information Criterion.
Usage

```r
ebic(object, g)
```

## S3 method for class 'glasso'
```r
ebic(object, g = 0.5)
```

## S3 method for class 'cglasso'
```r
ebic(object, g = 0.5)
```

Arguments

- `object` an object of class ‘glasso’, ‘ggm’, ‘cglasso’ or ‘cggm’.
- `g` the parameter indexing the extended BIC: a value belonging to the interval \([0, 1]\). Default is 0.5

Details

The measure of goodness-of-fit (gof) returned by the function ‘ebic’ depends on the class of the fitted model.

If ‘object’ has class ‘glasso’ or ‘ggm’, then ‘ebic’ computes the extended Bayesian Information Criterion (eBIC) proposed in Foygel and others (2010):

\[
eBIC = -2 \log \text{-likelihood} + a(\rho)(\log n + 4\gamma \log p),
\]

where \(a(\rho)\) denotes the number of non-zero off-diagonal elements in \(\hat{\Theta}_\rho\) and \(\gamma\) is a value belonging to the interval \([0, 1]\) indexing the measure of goodness-of-fit. As explained in Foygel and others (2010), the log-likelihood function is evaluated using the maximum likelihood estimates of the model select by glasso. For this reason, ‘ebic’ calls the generic function `mle` to fit the Gaussian graphical model (GGM) selected by `glasso`.

If ‘object’ has class ‘cglasso’ or ‘cggm’, eBIC is defined as:

\[
eBIC = -2 Q\text{-function} + a(\rho)(\log n + 4\gamma \log p),
\]

where the \(Q\)-function is evaluated at the M-step of the EM-like algorithm described in `mle`.

‘ebic’ returns an object with S3 class ‘gof’ for which are available the method functions ‘print.gof’ and ‘plot.gof’. These method functions are developed with the aim of helping the user in finding the optimal value of the tuning parameter, defined as the \(\rho\)-value minimizing the eBIC measures. For this reason, ‘print.gof’ shows also the ranking of the fitted models (the best model is pointed out with an arrow) whereas ‘plot.gof’ pointed out the optimal \(\rho\)-value by a vertical dashed line (see below for some examples).

Value

‘ebic’ returns an object with S3 class “gof”, i.e. a list containing the following components:

- `value_gof` the values of the measure of goodness-of-fit used to evaluate the fitted models.
- `rho` the values of the tuning parameter used to fit the glasso or cglasso model.
- `value` the values of the log-likelihood function or the values of the Q-function.
ebic

df the number of the estimated non-zero parameters, i.e. the number of non-zero partial correlations plus $2p$.

n the sample size.

p the number of variables.

model the name of the fitted model.

type the measure of goodness-of-fit used to evaluate the fitted models.

Author(s)

Luigi Augugliaro (<luigi.augugliaro@unipa.it>)

References


See Also

loglik, cglasso, glasso, mle, aic, bic and the method functions plot and summary.

Examples

library("cglasso")
set.seed(123)

###############
# cglasso model #
###############
n <- 100L
p <- 5L
mu <- rep.int(0, times = p)
X <- rdatacgmm(n = n, mu = mu, probr = 0.05)
out <- cglasso(X = X)
out_ebic <- ebic(out)
plot(out_ebic)

###############
# cggm model #
###############
out_mle <- mle(out)
out_ebic <- ebic(out_mle)
plot(out_ebic)

###############
# glasso model #
###############
library(MASS)
X <- mvrnorm(n = n, mu = mu, Sigma = diag(p))
out <- glasso(X)
out_ebic <- ebic(out)
out_ebic
plot(out_ebic)

# ggm model

out_mle <- mle(out)
out_ebic <- ebic(out_mle)
out_ebic
plot(out_ebic)

---

**event**

_Return the Indicator Matrix from an Object with class `datacggm`_

**Description**

The `event` function is used to create a status indicator matrix from an object with class `datacggm`. The elements of the matrix, denoted by \( R \), are used to specify the status of an observation:

- \( R[i, j] = 0 \) means that the \( i \)th observation of the \( j \)th random variable is observed;
- \( R[i, j] = -1 \) means that the \( i \)th observation of the \( j \)th random variable is left-censored;
- \( R[i, j] = +1 \) means that the \( i \)th observation of the \( j \)th random variable is right-censored.

See examples below.

**Usage**

`event(x)`

**Arguments**

- `x` an object with class `datacggm`.

**Value**

`event` returns a \((n \times p)\)-dimensional matrix.

**Author(s)**

Luigi Augugliaro (<luigi.augugliaro@unipa.it>)

**References**

See Also
datacggm, rdatacggm and the method function summary.datacggm.

Examples

```r
set.seed(123)
library("cglasso")

# dataset from a left-censored Gaussian graphical model
n <- 100L
p <- 5L
X <- matrix(rnorm(n * p), n, p)
lo <- -1
X[X <= lo] <- lo
X <- datacggm(X, lo = lo)
event(X)

# dataset from a right-censored Gaussian graphical model
n <- 100L
p <- 5L
X <- matrix(rnorm(n * p), n, p)
up <- 1
X[X >= up] <- up
X <- datacggm(X, up = up)
event(X)

# dataset from a censored Gaussian graphical model
n <- 100L
p <- 5L
X <- matrix(rnorm(n * p), n, p)
up <- 1
lo <- -1
X[X >= up] <- up
X[X <= lo] <- lo
X <- datacggm(X, lo = lo, up = up)
event(X)
```

glasso

Lasso Estimator for Gaussian Graphical Models

Description

‘glasso’ fits the l1-penalized Gaussian graphical model.

Usage

glasso(X, weights, pendiag = FALSE, nrho = 50L, rho.min.ratio, rho, maxR2,
maxit = 1.0E+4, thr = 1.0e-04, trace = 0L)
Arguments

- **X**: the \((n \times p)\)-dimensional matrix used to compute the covariance matrix.
- **weights**: an optional symmetric matrix of non-negative weights. This matrix can be used to specify the unpenalized partial correlation coefficients ('weights[i, j] = 0') or the structural zeros in the precision matrix ('weights[i, j] = +Inf'). See below for an example. By default, the glasso model is fitted without weights.
- **pendiag**: flag used to specify if the diagonal elements of the concentration matrix are penalized ('pendiag = FALSE') or unpenalized ('pendiag = TRUE').
- **nrho**: the integer specifying the number of tuning parameter used to fit the glasso model. Default is 'nrho = 50L'.
- **rho.min.ratio**: the smallest value for the tuning parameter \(\rho\), as a fraction of the smallest tuning parameter for which all the estimated partial correlation coefficients are zero. The default depends on the sample size 'n' relative to the number of variables 'p'. If 'p < n', the default is '1.0E-4' otherwise the value '1.0E-2' is used as default. A very small value of 'rho.min.ratio' will lead to a saturated fitted model in the 'p < n' case.
- **rho**: optional argument. A user supplied \(\rho\) sequence. WARNING: avoid supplying a single value for the tuning parameter; supply instead a decreasing sequence of \(\rho\)-values.
- **maxR2**: a value belonging to the interval [0, 1] specifying the largest value of the pseudo R-squared measure (see Section Details). The regularization path is stopped when \(R^2\) exceeds 'maxR2'. The default depends on the sample size 'n' relative to the number of variables 'p'. If 'p < n', the default is '1' otherwise the value '0.9' is used as default.
- **maxit**: maximum number of iterations of the glasso algorithm. Default is 1.0E+4.
- **thr**: threshold for the convergence of the glasso algorithm. Default is 1.0E-4.
- **trace**: integer for printing out information as iterations proceed: trace = 0 no information is printed out on video; trace = 1 basic information is printed out on video; trace = 2 detailed information is printed out on video.

Details

For a fixed value of the tuning parameter, glasso solves the following maximization problem:

\[
\max_\Theta \log \det \Theta - tr\{S\Theta\} - \rho \|\Theta\|_1,
\]

where \(\|\Theta\|_1 = \sum_{h,k} |\theta_{hk}|\). The previous maximization problem is solved efficiently combining the block-coordinate descent algorithm (Friedman and others, 2008) with the screening rule proposed in Witten and others (2011).

In order to avoid the overfitting of the model, we use the following pseudo R-squared measure:

\[
R^2 = 1 - \frac{\|S - \hat{\Sigma}_\rho\|_F}{\|S - \Sigma_{\rho_{\text{max}}}\|_F},
\]

where \(\|\cdot\|_F\) denotes the Frobenius norm and \(\rho_{\text{max}}\) denotes the smallest value of the tuning parameter for which all the estimated partial correlation coefficients are zero. By straightforward algebra, it
is easy to show that the proposed pseudo R-squared belongs to the closed interval $[0, 1]$: $R^2 = 0$ when the tuning parameter is equal to $\rho_{\text{max}}$ and $R^2 = 1$ when $\rho = 0$. The regularization path is stopped when $R^2$ exceeds ‘maxR2’.

Value

'glasso' returns an object with S3 class “glasso”, i.e. a list containing the following components:

call the call that produced this object.
X the matrix used to compute the covariance matrix.
S the covariance matrix used to fit the glasso model.
weights the used weights.
pendiag the flag specifying if the diagonal elements of the precision matrix are penalized.
nrho the number of fitted glasso model.
rho.min.ratio the scale factor used to compute the smallest value of the tuning parameter.
rho the $p$-dimensional vector reporting the values of the tuning parameter used to fit the glasso model.
maxR2 the threshold value used for the pseudo R-squared measure.
maxit the maximum number of iterations of the glasso algorithm.
thr the threshold for the convergence of the glasso algorithm.
Sgm an array of dimension $(p \times p \times nrho)$. Sgm[, , k] is the estimate of the covariance matrix of the glasso model fitted using rho[k].
Tht an array of dimension $(p \times p \times nrho)$. Tht[, , k] is the estimate of the precision matrix of the glasso model fitted using rho[k].
Adj an array of dimension $(p \times p \times nrho)$. Adj[, , k] is the adjacency matrix associated to Tht[, , k]. i.e. Adj[i, j, k] = 1 iff Tht[i, j, k] ≠ 0 and 0 otherwise.
df the $p$-dimensional vector reporting the number of non-zero partial correlation coefficients.
R2 the $p$-dimensional vector reporting the values of the measure $R^2$ described in the section Details.
ncomp the $p$-dimensional vector reporting the number of connected components (for internal purposes only).
Ck the $(p \times nrho)$-dimensional matrix encoding the connected components (for internal purposes only).
pk the $(p \times nrho)$-dimensional matrix reporting the number of vertices per connected component (for internal purposes only).
nit the $p$-dimensional vector reporting the number of iterations.
conv a description of the error that has occurred.
trace the integer used for printing out information.
Author(s)
Luigi Augugliaro (<luigi.augugliaro@unipa.it>)

References

See Also
cglasso, mle, to_graph and the method functions summary, coef, plot, aic, bic and ebic.

Examples
library("cglasso")
set.seed(123)

p <- 5L
n <- 100L
mu <- rep(0L, p)
Tht <- diag(p)
diag(Tht[-1L, -p]) <- diag(Tht[-p, -1L]) <- 0.3
Sgm <- solve(Tht)
X <- MASS::mvrnorm(n = n, mu = mu, Sigma = Sgm)
out <- glasso(X)
out

# in this example we use the argument 'weights' to specify
# the unpenalized partial correlation coefficients and the
# structural zeros in the precision matrix
w <- rep(1L, p * p)
dim(w) <- c(p, p)

diag(w) <- diag(w[-1L, -p]) <- diag(w[-p, -1L]) <- 0

diag(w[1L, 4L:5L]) <- w[4L:5L, 1L] <- +Inf
w[2L, 5L] <- w[5L, 2L] <- +Inf

w
out <- glasso(X = X, weights = w)

# checking structural zeros
out$Tht[, , out$nrho][w == +Inf]

# checking stationarity conditions of the MLE estimators
# (the unpenalized partial correlation coefficients)
\( \text{loglik} \)  

\[(\text{out}\$\text{Sgm}[\cdot, \text{out}\$\text{nrho}] - \text{out}\$\text{S}[w == 0L])\]

---

**loglik**  

*Extract Log-Likelihood or Q-Function*

---

**Description**

‘\text{loglik}’ function extracts the values of the log-likelihood function from an object with class ‘\text{glasso}’ or ‘\text{ggm}’, whereas the values of the Q-function are returned if the object has class ‘\text{cglasso}’ or ‘\text{cggm}’.

**Usage**

\texttt{loglik(object)}

**Arguments**

- **object**  
an object of class ‘\text{glasso}’, ‘\text{ggm}’, ‘\text{cglasso}’ or ‘\text{cggm}’.

**Details**

If ‘\text{object}’ has class ‘\text{glasso}’ or ‘\text{ggm}’, the function ‘\text{loglik()}’ returns the value of the log-likelihood function:  
\[ \frac{n}{2} \left\{ \log \det \Theta - \text{tr}(S\Theta) - p \log(2\pi) \right\}, \]

where \( \Theta \) is estimated using the function \text{glasso} or \text{mle.glasso}.

If ‘\text{object}’ has class ‘\text{cglasso}’ or ‘\text{cggm}’, the function ‘\text{loglik()}’ returns the value of the Q-function, i.e. the function maximized in the M-step of the EM-like algorithm (see details in \text{cglasso} or \text{mle.cglasso}). The Q-function is defined as follows:

\[ \frac{n}{2} \left\{ \log \det \Theta - \text{tr}(S'\Theta) - p \log(2\pi) \right\}, \]

where \( S' \) is computed in the E-step.

The method function ‘\text{print.loglik}’ is used to improve the readability of the results.

**Value**

‘\text{loglik}’ returns an object with S3 class “\text{loglik}”, i.e. a list containing the following components:

- **value**  
  the values of the log-likelihood function or the values of the Q-function.

- **df**  
  the number of the estimated non-zero parameters, i.e. the number of non-zero partial correlations plus \( 2p \).

- **n**  
  the sample size.

- **p**  
  the number of variables.

- **rho**  
  the values of the tuning parameter used to fit the model.

- **model**  
  the name of the fitted model.

- **fun**  
  the name of the used function, i.e. the log-likelihood or the Q-function.
**Author(s)**

Luigi Augugliaro (<luigi.augugliaro@unipa.it>)

**See Also**

cglasso, glasso, mle and the method functions, plot, aic, bic and ebic.

**Examples**

```r
library("cglasso")
set.seed(123)

#############################
# cglasso model #
#############################
p <- 5L
n <- 100L
mu <- rep(0L, p)
Tht <- diag(p)
diag(Tht[-1L, -p]) <- diag(Tht[-p, -1L]) <- 0.3
Sgm <- solve(Tht)
X <- rdatacggm(n = n, mu = mu, Sigma = Sgm, probr = 0.05)
out <- cglasso(X = X)
out_loglik <- loglik(out)
out_loglik

#############################
# cggm model #
#############################
out_mle <- mle(out)
out_loglik <- loglik(out_mle)
out_loglik

#############################
# glasso model #
#############################
library(MASS)
X <- mvrnorm(n = n, mu = mu, Sigma = Sgm)
out <- glasso(X)
out_loglik <- loglik(out)
out_loglik

#############################
# ggm model #
#############################
out_mle <- mle(out)
out_loglik <- loglik(out_mle)
out_loglik
```
Maximum Likelihood Estimation

Description

The generic function `mle` fits the graphical model selected by `glasso` and `cglasso`.

Usage

```r
mle(object, ...) # S3 method for class 'glasso'
mle(object, maxit = 1.0e+04, thr = 1.0e-04, trace = 0L, ...)
```

```r
mle(object, maxit_em = 1.0e+04, thr_em = 1.0e-4, maxit_bcd = 1.0e+4, thr_bcd = 1.0e-4, trace = 0L, ...)
```

Arguments

- `object`: an object of class 'glasso' or 'cglasso'.
- `maxit`: maximum number of iterations of the glasso algorithm. Default is 1.0E+4.
- `thr`: threshold for the convergence of the glasso algorithm. Default is 1.0E-4.
- `maxit_em`: maximum number of iterations of the EM algorithm. Default is 1.0E+03.
- `thr_em`: threshold for the convergence of the EM algorithm. Default value is 1.0E-4.
- `maxit_bcd`: maximum number of iterations of the glasso algorithm. Default is 1.0E+4.
- `thr_bcd`: threshold for the convergence of the glasso algorithm. Default is 1.0E-4.
- `trace`: integer for printing out information as iterations proceed: trace = 0 no information is printed out on video; trace = 1 basic information is printed out on video; trace = 2 detailed information is printed out on video.

Details

The generic function `mle` computes the maximum likelihood estimates of the graphical model selected by the function `glasso` or `cglasso`.

If `object` has class 'glasso', the method function `mle.glasso` computes the maximum likelihood estimates of the parameters of the Gaussian graphical models (ggm) associated to the sequence of glasso estimates. Formally, for a given value of the tuning parameter let $\hat{\Theta}_\rho$ be the glasso estimate of the precision matrix, then `mle.glasso` solves the following maximization problem:

$$\max_{\Theta} \log \det \Theta - tr\{S\Theta\},$$

where $\bar{\theta}_{hk} = 0$ if $\hat{\theta}_{hk}^{\rho} = 0$ otherwise it is estimated.
If `object` has class `cglasso`, the method function `mle.cglasso` computes the maximum likelihood estimates of the parameters of the censored Gaussian graphical models (cggm) associated to the sequence of cglasso estimates. Formally, for a given value of the tuning parameter let $\hat{\Theta}^\rho$ be the cglasso estimate of the precision matrix, then `mle.cglasso` computes the maximum likelihood estimate by the following EM-like algorithm:

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>let $\hat{\Theta}^\rho$ be the cglasso estimate of the precision matrix;</td>
</tr>
<tr>
<td>2.</td>
<td><strong>E-step</strong> use the moments of the truncated normal distribution to compute the current estimates of the marginal means, denoted by $\bar{x}'$, and to complete the empirical covariance matrix $S'$;</td>
</tr>
<tr>
<td>3.</td>
<td><strong>M-step</strong> let $\mu = \bar{x}'$; estimate $\hat{\Theta}$ maximixing the $Q$-function: $\log det \hat{\Theta} - tr{S'\hat{\Theta}}$, where $\bar{\theta}<em>{hk} = 0$ if $\hat{\theta}</em>{hk}^\rho = 0$ otherwise it is estimated;</td>
</tr>
<tr>
<td>4.</td>
<td>repeat steps 2. and 3. until a convergence criterion is met.</td>
</tr>
</tbody>
</table>

**Value**

If `object` has class `glasso`, then `mle` returns and object with S3 class `ggm`, which inherits the class `glasso`. An object with class `ggm` is a list containing the following components:

- **call** the call that produced this object.
- **X** the matrix used to compute the covariance matrix.
- **S** the covariance matrix used to fit the ggm model.
- **nrho** the number of fitted ggm model.
- **rho** the $p$-dimensional vector reporting the values of the tuning parameter used to fit the glasso model.
- **maxit** the maximum number of iterations of the glasso algorithm.
- **thr** the threshold for the convergence of the glasso algorithm.
- **Sgm** an array of dimension $(p \times p \times nrho)$. $\text{Sgm}[, , k]$ is the estimate of the covariance matrix of the $k$th ggm model.
- **Tht** an array of dimension $(p \times p \times nrho)$. $\text{Tht}[, , k]$ is the estimate of the precision matrix of the $k$th ggm model.
- **Adj** an array of dimension $(p \times p \times nrho)$. $\text{Adj}[, , k]$ is the adjacency matrix associated to $\text{Tht}[, , k]$, i.e. $\text{Adj}[i, j, k] = 1$ iff $\text{Tht}[i, j, k] \neq 0$ and 0 otherwise.
- **df** the $p$-dimensional vector reporting the number of non-zero partial correlation coefficients.
- **R2** the $p$-dimensional vector reporting the values of the measure $R^2$. See section Details, in glasso.
- **ncomp** the $p$-dimensional vector reporting the number of connected components (for internal purposes only).
- **Ck** the $(p \times nrho)$-dimensional matrix encoding the connected components (for internal purposes only).
pk the \((p \times \text{nrho})\)-dimensional matrix reporting the number of vertices per connected component (for internal purposes only).

\text{nit} the \(p\)-dimensional vector reporting the number of iterations.

\text{conv} a description of the error that has occurred.

\text{trace} the integer used for printing out information.

If ‘object’ has class ‘\text{cglasso}’, then ‘mle’ returns and object with S3 class ‘\text{cggm}’, which inherits the class ‘\text{cglasso}’. An object with class ‘\text{cggm}’ is a list containing the following components:

\text{call} the call that produced this object.

\text{X} the object with S3 class ‘\text{datacggm}’ used to fit the cggm model.

\text{nrho} the number of fitted cggm model.

\text{rho} the \(p\)-dimensional vector reporting the values of the tuning parameter used to fit the cglasso model.

\text{maxit_em} maximum number of iterations of the EM algorithm.

\text{thr_em} threshold for the convergence of the EM algorithm.

\text{maxit_bcd} maximum number of iterations of the glasso algorithm.

\text{thr_bcd} threshold for the convergence of the glasso algorithm.

\text{Xipt} an array of dimension \(n \times p \times \text{nrho}\). \text{Xipt}[, , k] is the matrix where the censored vaules are replaced using the conditional expected vaules computed in the E-step of the algorithm described in section Details.

\text{S} an array of dimension \(p \times p \times \text{nrho}\). \text{S}[, , k] is the matrix \(S'\) used to fit the cggm model (see the section Details).

\text{mu} a matrix of dimension \(p \times \text{nrho}\). The \(k\)th column is the estimate of the expected values of the \(k\)th cggm model.

\text{Sgm} an array of dimension \(p \times p \times \text{nrho}\). \text{Sgm}[, , k] is the estimate of the covariance matrix of the \(k\)th cggm model.

\text{Tht} an array of dimension \(p \times p \times \text{nrho}\). \text{Tht}[, , k] is the estimate of the precision matrix of the \(k\)th cggm model.

\text{Adj} an array of dimension \(p \times p \times \text{nrho}\). \text{Adj}[, , k] is the adjacency matrix associated to \text{Tht}[, , k], i.e. \text{Adj}[i, j, k] = 1 \text{iff Tht}[i, j, k] \neq 0 and 0 otherwise.

\text{df} the \(p\)-dimensional vector reporting the number of non-zero partial correlation coefficients.

\text{R2} the \(p\)-dimensional vector reporting the values of the measure \(R^2\). See section Details, in cglasso.

\text{ncomp} the \(p\)-dimensional vector reporting the number of connected components (for internal purposes only).

\text{Ck} the \((p \times \text{nrho})\)-dimensional matrix encoding the connected components (for internal purposes only).

\text{pk} the \((p \times \text{nrho})\)-dimensional matrix reporting the number of vertices per connected component (for internal purposes only).
mle

 nit the (nrho \times 2)-dimensional matrix reporting the number of iterations.
 conv a description of the error that has occurred.
 subrout the name of the Fortran subroutine where the error has occurred (for internal
debug only).
 trace the integer used for printing out information.

Author(s)
Luigi Augugliaro (<luigi.augugliaro@unipa.it>)

References
Biostatistics (to appear).

See Also
glasso, cglasso, to_graph, and the method functions summary, coef, plot, aic, bic and ebic.

Examples
library("cglasso")
set.seed(123)

# cglaso model #
# cglaso model #

n <- 100L
p <- 5L
mu <- rep.int(0L, times = p)
X <- rdatccgm(n = n, mu = mu, probr = 0.05)
out <- cglasso(X = X)
out_mle <- mle(out)
out_mle

inherits(out_mle, "cglasso")
class(out_mle)

# inheriting method functions from 'cglasso': some examples
coef(out_mle, nrho = 10L, print.info = TRUE)
to_graph(out_mle, nrho = 10L, weighted = TRUE)
out_aic <- aic(out_mle)
out_aic
plot(out_mle, typeplot = "graph", gof = out_aic)

# glasso model #
# glasso model #

X <- MASS::mvrnorm(n = n, mu = mu, Sigma = diag(p))
out <- glasso(X)
out_mle <- mle(out)
out_mle

inherits(out_mle, "glasso")
class(out_mle)

# inheriting method functions from 'glasso': some examples
coeff(out_mle, nrho = 10L, print.info = TRUE)
to_graph(out_mle, nrho = 10L, weighted = TRUE)
out_aic <- aic(out_mle)
out_aic
plot(out_mle, typeplot = "graph", gof = out_aic)

plot

Plot for ‘glasso’ and ‘cglasso’ Object

Description

The method functions ‘plot.glasso’ and ‘plot.cglasso’ produce plots to study the sequence of models estimated by glasso or cglasso.

Usage

## S3 method for class 'glasso'
plot(x, typeplot = c("path", "graph"),
     gof, diag = FALSE, nrho, weighted = FALSE, ...)

## S3 method for class 'cglasso'
plot(x, typeplot = c("path", "graph"),
     gof, diag = FALSE, nrho, weighted = FALSE, ...)

Arguments

x an object of class ‘glasso’, ‘ggm’, ‘cglasso’ or ‘cggm’.
typeplot a string specifying the plot produced.
gof an object with class ‘gof’. See examples below.
diag flag specifying whether the diagonal elements of the concentration matrix are plotted.
nrho argument availables only if ‘typeplot = graph’: integer specifying the model used to produce the ‘igraph’ object. This argument is overwritten if ‘gof’ is available. See examples below.
weighted argument availables only if ‘typeplot = graph’. Flag specifying whether to create a weighted graph.
... additional arguments passed to the method function ‘plot.default’.
**Details**

The plot produced by the method functions `plot.glasso` and `plot.cglasso` depends on the argument `typeplot`.

If `typeplot = "path"`, the regularization paths are produced; in this case, if an object with class 'gof' is passed by the argument 'gof', then a vertical dashed line is added to identify the optimal \( \rho \)-value.

If `typeplot = "graph"`, the method functions `plot.glasso` and `plot.cglasso` produce the undirected graph associated to the model specified by the argument 'nrho'. If an object with class 'gof' is passed by 'gof', the undirected graph of the model selected by the function 'aic', 'bic' or 'ebic' is produced.

**Value**

If `typeplot = "graph"` then the method functions `plot.glasso` and `plot.cglasso` return an igraph object (see example below).

**Author(s)**

Luigi Augugliaro (luigi.augugliaro@unipa.it)

**Examples**

```r
library("cglasso")
set.seed(123)

# cglasso model 

# plotting the regularization paths + 'gof' object
plot(out, typeplot = "path")
plot(out, typeplot = "path", gof = out_aic)

# plotting the graph associated to the fitted model
# specified by 'nrho'
out_graph <- plot(out, typeplot = "graph", nrho = 10L)
out_graph

# plotting the graph associated to the fitted model
# specified by 'gof'
out_graph <- plot(out, typeplot = "graph", gof = out_aic)
out_graph

# cglasso model 
```
rdatacggm

Simulate from a Censored Gaussian Graphical Model

Description

rdatacggm function is used to produce one or more samples from the specified censored Gaussian graphical model.

Usage

rdatacggm(n, mu, Sigma, probl, probr, lo, up, ...)

Arguments

n the number of samples required.
mu a vector giving the means of the variables. By default all the expected values are equal to zero.
Sigma a positive-definite symmetric matrix specifying the covariance matrix of the variables. By default the identity matrix is used as covariance matrix.
probl a vector giving the probabilities that the random variables are left-censored.
probr a vector giving the probabilities that the random variables are right-censored.
lo a vector giving the left-censoring values.
up a vector giving the right-censoring values.
... further arguments passed to 'mvrnorm'.
Details

`rdatacggm` function simulates a dataset from a censored Gaussian graphical model and returns an object with class `datacggm`.

The dataset is simulated in two steps:

1. in the first step the arguments `n`, `mu`, `Sigma` and ... are passed to the function `mvtnorm` to simulate one or more samples from the specified multivariate Gaussian distribution.
2. in the second step, the values that are below or upper the censoring values are replaced.

The user can specify the censoring values in two equivalent ways. The simplest way is to use the arguments `lo` and `up`; a warning is produced if a full-observed dataset is simulated (see the last example). Alternatively, the censoring values can be implicitly specified using the arguments `probl` and `probr`. The $j$th lower censoring value, denoted by $l_j$, is such that:

$$\text{probl}[j] = \Pr\{X_j \leq l_j\}.$$

In the same way, the $j$th upper censoring value, denoted by $u_j$, is such that:

$$\text{probr}[j] = \Pr\{X_j \geq u_j\}.$$

Value

`rdatacggm` returns an object with class `datacggm`.

Author(s)

Luigi Augugliaro (<luigi.augugliaro@unipa.it>)

References


See Also

datacggm, event, cglasso and the method function `summary.datacggm`.

Examples

```r
set.seed(123)
library("cglasso")

n <- 1000L
p <- 3L
mu <- rep(1L, p)
rho <- 0.3
Sigma <- outer(1L:p, 1L:p, function(i, j) rho*abs(i - j))

# simulating a dataset from a left-censored Gaussian graphical model
X <- rdatacggm(n = n, mu = mu, Sigma = diag(p), prob1 = 0.05)
# the same: X <- rdatacggm(n = n, mu = mu, Sigma = Sigma, prob1 = 0.05, probr = 0)
```
# the same: X <- rdatacggm(n = n, mu = mu, Sigma = Sigma, probl = 0.05, up = +Inf)
summary(X)

# simulating a dataset from a right-censored Gaussian graphical model
X <- rdatacggm(n = n, mu = mu, Sigma = diag(p), probl = 0.05)
# the same: X <- rdatacggm(n = n, mu = mu, Sigma = Sigma, probl = 0.05, prob = 0)
# the same: X <- rdatacggm(n = n, mu = mu, Sigma = Sigma, probl = 0.05, lo = -Inf)
summary(X)

# simulating a dataset from a censored Gaussian graphical model
X <- rdatacggm(n = n, mu = mu, Sigma = Sigma, prob = 0.05, prob = 0.05)
summary(X)

# simulating a full observed dataset: a warning is produced
X <- rdatacggm(n = n, mu = mu, Sigma = Sigma, lo = -Inf, up = Inf)
summary(X)

## summary

### Summary Method

<table>
<thead>
<tr>
<th>summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
</tr>
<tr>
<td>‘summary’ produces a summary of the sequence of fitted models.</td>
</tr>
</tbody>
</table>

### Usage

```r
## S3 method for class 'glasso'
summary(object, gof = c("BIC", "AIC", "eBIC"), par.gof, digits = 4L, ...)

## S3 method for class 'cglasso'
summary(object, gof = c("BIC", "AIC", "eBIC"), par.gof, digits = 4L, ...)
```

### Arguments

<table>
<thead>
<tr>
<th>object</th>
<th>an object of class ‘glasso’, ‘ggm’, ‘cglasso’ or ‘cggm’.</th>
</tr>
</thead>
<tbody>
<tr>
<td>gof</td>
<td>string specifying the measure of goodness-of-fit used to evaluate the fitted models. Default is ‘BIC’.</td>
</tr>
<tr>
<td>par.gof</td>
<td>the parameter of the measure of goodness-of-fit used to evaluate the fitted models.</td>
</tr>
<tr>
<td>digits</td>
<td>the minimum number of significant digits to be used: see ‘print.default’.</td>
</tr>
<tr>
<td>...</td>
<td>further arguments passed to the method function ‘print.data.frame’.</td>
</tr>
</tbody>
</table>
Details

The method functions `summary.glasso` and `summary.cglasso` give information about the sequence of fitted models. The output is divided into two sections.

First section shows the call that produced object followed by a `data.frame` reporting the values of the tuning parameter used to fit the model (\( \rho \)), the number of non-zero estimates (\( \text{df} \)), the values of the pseudo R-squared (\( \text{R}^2 \)) described in `glasso` and `cglasso`, the values of the measure of goodness-of-fit used to evaluate the fitted models and the ranking of the fitted models (Rank). The model with the lowest measure of goodness-of-fit is pointed out by an arrow.

Second section shows the details of the selected model plus the number of connected components and the number of vertices per component.

Value

The functions `summary.glasso` and `summary.cglasso` compute and return a list of summary statistics with the following elements:

- `table`: a `data.frame` containing the summary statistics used to evaluate the sequence of fitted models.
- `which.min`: the number of the model with the lowest measure of goodness-of-fit.

Author(s)

Luigi Augugliaro (<luigi.augugliaro@unipa.it>)

See Also

cglasso, glasso, mle, aic bic and ebic.

Examples

```r
library("cglasso")
set.seed(123)

# cglasso model #
#------------------
n <- 100L
p <- 5L
mu <- rep.int(0L, times = p)
X <- rdatacgmm(n = n, mu = mu, probr = 0.05)
out <- cglasso(X = X)
summary(out, gof = "AIC")
summary(out, gof = "BIC")
summary(out, gof = "eBIC")

# cggm model #
#------------
out_mle <- mle(out)
summary(out_mle, gof = "AIC")
```
summary.datacggm

```
summary(out_mle, gof = "BIC")
summary(out_mle, gof = "eBIC")

# glasso model #
X <- MASS::mvtnorm(n = n, mu = mu, Sigma = diag(p))
out <- glasso(X)
summary(out, gof = "AIC")
summary(out, gof = "BIC")
summary(out, gof = "eBIC")

# ggm model #
out_mle <- mle(out)
summary(out_mle, gof = "AIC")
summary(out_mle, gof = "BIC")
summary(out_mle, gof = "eBIC")
```

---

**summary.datacggm**  
*Summarizing Objects of Class ‘cggmdata’*

**Description**

'`summary.datacggm' function is used to produce summaries of an object of class ‘datacggm’.

**Usage**

```
## S3 method for class 'datacggm'
summary(object, digits = max(3L, getOption("digits") - 3L), ...)
```

**Arguments**

- **object**  
an object of class ‘datacggm’.  
- **digits**  
integer used for number formatting with ‘format()’.  
- **...**  
further arguments passed to ‘format()’.  

**Details**

The method function ‘summary.datacggm’ extends the results given by ‘summary.matrix()’ taking into account the censoring values. For each variable, the mean and the quartiles are computing using only the observed values; the lower and upper censoring values (denoted by ‘Lower’ and ‘Upper’) are also reported. The number of observed and censured values are computed and showed in the second part of the output (see example below).
Value

‘summary.dataccgm’ returns a matrix with class ‘table’, obtained by computing the summary measures to each variable and collating the results.

Author(s)

Luigi Augugliaro (<luigi.augugliaro@unipa.it>)

References


See Also

dataccgm and rdataccgm.

Examples

```r
set.seed(123)
library("cglasso")

n <- 1000L
p <- 3L
mu <- rep(1L, p)
rho <- 0.3
Sigma <- outer(1:p, 1:p, function(i, j) rho^abs(i - j))
X <- rdataccgm(n = n, mu = mu, Sigma = Sigma, probr = 0.05, probl = 0.05)
summary(X)
```

Description

‘to_graph’ function is used to create an undirected graph from objects returned by the function ‘glasso’ or ‘cglasso’.

Usage

to_graph(object, nrho = 1L, weighted = FALSE)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>object</td>
<td>an object of class ‘glasso’, ‘ggm’, ‘cglasso’ or ‘cggm’.</td>
</tr>
<tr>
<td>nrho</td>
<td>integer specifying the model used to create the undirected graph. Default is 1L.</td>
</tr>
<tr>
<td>weighted</td>
<td>flag specifying whether to create a weighted graph from the adjacency matrix. Default is FALSE.</td>
</tr>
</tbody>
</table>
to_graph

Details

The adjacency matrix `object$Adj[, , nrho]` is passed to `graph_from_adjacency_matrix`, available in the R package `igraph`, to create the undirected graph estimated by the glasso or cglasso model. If `weighted = TRUE` then the precision matrix `object$Tht[, , nrho]` is used to create a weighted undirected graph. In this case, an edge associated to a negative partial correlation coefficient is plotted using a dashed line (see examples below).

Value

`to_graph` returns an `igraph` object.

Author(s)

Luigi Augugliaro (<luigi.augugliaro@unipa.it>)

References


See Also

glasso and cglasso. For more details about the object of class `igraph`, the interested reader is referred to the package `igraph`.

Examples

```r
library("cglasso")
set.seed(123)

# cglasso model #
n <- 100L
p <- 10L
mu <- rep.int(0L, times = p)
x <- rdatacgmm(n = n, mu = mu, probr = 0.05)
out <- cglasso(x)
out

# creating the undirected graph associated to the fitted model number 3
graph_cglasso <- to_graph(out, nrho = 3L)
plot(graph_cglasso, layout = layout_in_circle(graph_cglasso))

# creating the weighted graph associated to the fitted model number 3
graph_cglasso <- to_graph(out, nrho = 3L, weighted = TRUE)
```

```r
graph_cglasso
E(graph_cglasso)
E(graph_cglasso)$weight
plot(graph_cglasso, layout = layout_in_circle(graph_cglasso))

#################################################
# glasso model #
#################################################
n <- 100L
p <- 10L
X <- matrix(rnorm(n * p), nrow = n, ncol = p)
out <- glasso(X)
out

# creating the undirected graph associated to the fitted
# model number 3
graph_cglasso <- to_graph(out, nrho = 3L)
graph_cglasso
V(graph_cglasso)
E(graph_cglasso)
plot(graph_cglasso, layout = layout_in_circle(graph_cglasso))

# creating the weighted graph associated to the fitted
# model number 3
graph_cglasso <- to_graph(out, nrho = 3L, weighted = TRUE)
graph_cglasso
E(graph_cglasso)
E(graph_cglasso)$weight
plot(graph_cglasso, layout = layout_in_circle(graph_cglasso))
```
Index

*Topic datagen
  datacggm, 11
  event, 16
  rdatacggm, 29
  summary.datacggm, 33

*Topic distribution
  rdatacggm, 29

*Topic graphs
  cglasso, 5
  coef, 9
  glasso, 17
  mle, 23
  plot, 27
  to_graph, 34

*Topic hplot
  plot, 27

*Topic models
  aic, 3
  cglasso, 5
  cglasso-package, 2
  coef, 9
  ebic, 13
  glasso, 17
  loglik, 21
  mle, 23
  summary, 31

*Topic multivariate
  cglasso, 5
  cglasso-package, 2
  coef, 9
  datacggm, 11
  event, 16
  glasso, 17
  mle, 23
  rdatacggm, 29
  summary, 31
  summary.datacggm, 33
  to_graph, 34

*Topic package
  cglasso-package, 2
  [.datacggm(datacggm), 11
  aic, 3, 9, 15, 20, 22, 26, 28, 32
  as.matrix.datacggm(datacggm), 11
  bic, 9, 15, 20, 22, 26, 28, 32
  bic(aic), 3
  cglasso, 3, 4, 5, 10, 13, 15, 20, 21, 23, 25–27, 30, 32, 35
  cglasso-package, 2
  coef, 9, 9, 10, 20, 26
  datacggm, 6, 9, 11, 17, 30, 34
  dim.datacggm(datacggm), 11
  dimnames.datacggm(datacggm), 11
  dimnames<-.datacggm(datacggm), 11
  ebic, 4, 9, 13, 20, 22, 26, 28, 32
  event, 12, 13, 16, 30
  glasso, 4, 9, 10, 14, 15, 17, 21–24, 26, 27, 32, 35
  graph_from_adjacency_matrix, 35
  igraph, 28, 35
  is.datacggm(datacggm), 11
  loglik, 3, 4, 15, 21
  mle, 3, 4, 9, 10, 14, 15, 20, 22, 23, 32
  mle.cglasso, 21
  mle.glasso, 21
  mvrnorm, 30
  plot, 4, 9, 15, 20, 22, 26, 27
  plot.gof(aic), 3
  print.cglasso(cglasso), 5
  print.data.frame, 31
  print.datacggm(datacggm), 11
print.default, 31
print.glasso (glasso), 17
print.gof (aic), 3
print.loglik (loglik), 21
rdatacggm, 13, 17, 29, 34
summary, 4, 9, 15, 20, 26, 31
summary.datacggm, 12, 13, 17, 30, 33
to_graph, 9, 20, 26, 34