Package: sglasso (via r-universe)

September 29, 2024

Type Package

Title Lasso Method for RCON(V,E) Models

Version 1.2.6

Date 2023-12-03

Author Luigi Augugliaro

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

Depends Matrix, igraph, R (>= 4.2)

Imports methods, graphics, grDevices

Description RCON(V, E) models are a kind of restriction of the Gaussian Graphical Models defined by a set of equality constraints on the entries of the concentration matrix. 'sglasso' package implements the structured graphical lasso (sglasso) estimator proposed in Abbruzzo et al. (2014) for the weighted l1-penalized RCON(V, E) model. Two cyclic coordinate algorithms are implemented to compute the sglasso estimator, i.e. a cyclic coordinate minimization (CCM) and a cyclic coordinate descent (CCD) algorithm.

License GPL (>= 2)

LazyLoad yes

URL http://www.unipa.it/persone/docenti/a/luigi.augugliaro

NeedsCompilation yes

Repository CRAN

Date/Publication 2023-12-03 08:40:02 UTC

Contents

sglasso-package 2
fglasso 3
gplot 5
gplot.fglasso 6
gplot.sglasso 7

Kh 9
klcv 10
loglik 11
neisseria 13
plot.klcv 13
plot.sglasso 14
sglasso 15
summary.sglasso 18

Index **20**

`sglasso-package` *Lasso Method for RCON(V, E) Models*

Description

RCON(V, E) models (Hojsgaard, et al., 2008) are a kind of restriction of the Gaussian Graphical Models defined by a set of equality constraints on the entries of the concentration matrix. `sglasso` package implements the structured graphical lasso (sglasso) estimator proposed in Abbruzzo et al. (2014) for the weighted l1-penalized *RCON(V, E)* model. Two cyclic coordinate algorithms are implemented to compute the sglasso estimator, i.e. a cyclic coordinate minimization (CCM) and a cyclic coordinate descent (CCD) algorithm.

Details

Package:	sglasso
Type:	Package
Version:	1.2.6
Date:	2023-12-03
License:	GPL (>=2)

Author(s)

Luigi Augugliaro

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

References

Abbruzzo, A., Augugliaro, L., Mineo, A. M. and Wit, E. C. (2014) Cyclic coordinate for penalized Gaussian Graphical Models with symmetry restrictions. In *Proceeding of COMPSTAT 2014 - 21th International Conference on Computational Statistics*, Geneva, August 19-24, 2014.

Hojsgaard, S. and Lauritzen, S. L. (2008) Graphical gaussian models with edge and vertex symmetries. *J. Roy. Statist. Soc. Ser. B.*, Vol. **70(5)**, 1005–1027.

`fglasso` *L1-penalized Factorial Graphical Lasso Model*

Description

Fit the weight l1-penlized factorial dynamic Gaussian Graphical Model.

Usage

```
fglasso(S, model, tp, p, ...)
```

Arguments

`S`	the empirical variance/covariance matrix;
`model`	a list or a matrix used to specify the factorial dynamic Gaussian Graphical Model (see Details);
`tp`	number of time points;
`p`	number of random variables observed for each time point;
`...`	further arguments passed to `sglasso`.

Details

The factorial dynamic Gaussian Graphical Model (Abbruzzo et al., 2015) is a special kind of *RCON(V, E)* model (Hojsgaard, et al.,2008) proposed to study dynamic networks. Let $X_t = (X_{it}, \ldots, X_{it})'$ be a p-dimensional random variable at time t. Assuming that $X = (X_1', \ldots, X_T')$ follows a multivariate normal distribution, the concentration matrix K has the following block structure

$$K = \begin{pmatrix} K_{1,1} & K_{1,2} & \ldots & K_{1,T} \\ K_{2,1} & K_{2,2} & \ldots & K_{2,T} \\ \vdots & \vdots & \ddots & \vdots \\ K_{T,1} & K_{T,2} & \ldots & K_{T,T} \end{pmatrix},$$

where $K_{t,t}$ give information about the conditinal independence structure among the p random variables at time t, and $K_{t,t+h}$ give information about the conditional independence structure between X_t and X_{t+h}. An interpretation of the elements of the submatrices $K_{t,t+h}$ brings to the notion of natural structure, i.e.,

$$K_{t,t+h} = \begin{pmatrix} k_{1,1}^{t,t+h} & 0 & \ldots & 0 \\ 0 & k_{2,2}^{t,t+h} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & k_{p,p}^{t,t+h} \end{pmatrix} + \begin{pmatrix} 0 & k_{1,2}^{t,t+h} & \ldots & k_{1,p}^{t,t+h} \\ k_{2,1}^{t,t+h} & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ k_{p,1}^{t,t+h} & k_{p,2}^{t,t+h} & \ldots & 0 \end{pmatrix}.$$

The entries of the first matrix are called **self-self conditinal dependences** at temporal lag h and represents the (negative) self-similarity of a given random variable across different time points. The entries of the second matrix are the **conditional dependence** among the p random variables. To

make the interpretation of the results more relevant and, at the same time, reduce the number of parameters, the authors propose the following equality constraints:

	$k_{i,i}^{t,t+h}$	effect	R code		$k_{i,j}^{t,t+h}$	effect	R code
i.	0	zero	"."	iv.	0	zero	"."
ii.	s^h	costant	"c"	ii.	n^h	costant	"c"
iii.	s_i^h	unit	"u"	iii.	n_i^h	unit	"u"
iv.	$s^{t,h}$	time	"t"	iv.	$n^{t,h}$	time	"t"
v.	$s_i^{t,h}$	interaction	"ut"	v.	$n_{i,j}^{t,h}$	interaction	"ut"

Argument `model` is used to specify the restrinctions previously describted. This argument can be a named list or a matrix with dimension $nlag \times 2$, where $nlag \leq$ `tp`. To gain more insight, suppose that we want to model only the sub-matrices $K_{t,t}$ and $K_{t,t+1}$, i.e., the sub-matrices corresponding to the temporal lag zero and one. A possible `R code` is

```
model.mat <- matrix("", nrow = 2, ncol = 2)
rownames(model.mat) <- c("lag0", "lag1")
colnames(model.mat) <- c("s", "n")
model.mat[1, ] <- c("c", "ut")
model.mat[2, ] <- c("t", ".")
```

In this example we are modelling the diagonal elements of the sub-matrices $K_{t,t}$ with the constant effect while the off-diagonal elements are modelled by the interaction effect. In the same way, the diagonal elements of the sub-matrices $K_{t,t+1}$ are modelled by the time effect while the remaning elements are equal to zero. The `fglasso` function passes the matrix `model.mat` to the internal function `fglasso_model2mask`, i.e.,

```
mask <- fglasso_model2mask(model.mat, tp = 3, p = 3)
```

which returns the mask used in `sglasso` to fit the specified factorial dynamic Gaussian Graphical model. The same model can be specified by the following named list

```
model.list <- list(lag0 = c(s = "c", n = "ut"), lag1 = c(s = "t", n = "."))
```

See the example below for more details.

Value

`fglasso` returns an obejct with S3 class `"sglasso"`. See the corresponding manual for more details.

Author(s)

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

References

Wit, E. C. and Abbruzzo, A.(2015) Dynamic factorial graphical models for dynamic networks. *Network Science*, Vol. **3(1)**, 37– 57

Abbruzzo, A., Augugliaro, L., Mineo, A.M. and Wit, E.C. (2014) Cyclic coordinate for penalized Gaussian Graphical Models with symmetry restrictions. In *Proceeding of COMPSTAT 2014 - 21th International Conference on Computational Statistics*. Geneva, August 19-24, 2014.

Hojsgaard, S. and Lauritzen, S.L. (2008) Graphical gaussian models with edge and vertex symmetries. *J. Roy. Statist. Soc. Ser. B.*, Vol. **70(5)**, 1005–1027.

See Also

`sglasso` function.

Examples

```
######################
# fglasso solution path
######################
N <- 50
tp <- 3
p <- 3
X <- matrix(rnorm(N * p * tp), N, tp * p)
S <- crossprod(X) / N
model <- list(lag0 = c(s = "c", n = "ut"), lag1 = c(s = "t", n = "."))
out.fglasso <- fglasso(S = S, model = model, tp = tp, p = p)
out.fglasso
```

`gplot` *Plotting Sparse Graph*

Description

`gplot` is a generic function for plotting sparse graphs.

Usage

```
gplot(object, ...)
```

Arguments

`object`	fitted `sglasso`/`fglasso` object;
`...`	other parameters passed to `gplot.sglasso` or `gplot.fglasso`.

Details

gplot is a generic function used to plot a graph estimated by sglasso or fglasso. See the method function gplot.sglasso or gplot.fglasso for more details about the specific arguments.

Author(s)

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

See Also

gplot.sglasso and gplot.fglasso method functions.

---

gplot.fglasso *Plotting Sparse Factorial Dynamic Gaussian Graphical Model*

Description

gplot.fglasso shows the sequence of graphs estimated by fglasso.

Usage

```
## S3 method for class 'fglasso'
gplot(object, rhoid, tp = c(1, 2), sub.tp1, sub.tp2, cex.sub = 1,
    k = 1.5, layout = layout.circle, ...)
```

Arguments

object	fitted fglasso object;
rhoid	an integer used to specificy the ρ-value used to fit the fglasso model;
tp	a vector of length equal to two used to specify the time points of the two graphs that will be compared. By default the first two time points are used;
sub.tp1	sub title for the graph estimated at time point tp[1];
sub.tp2	sub title for the graph estimated at time point tp[2];
cex.sub	a numerical value giving the amount by which plotting sub titles should be magnified relateve to the default;
k	value used to specify the distance between the two graphs;
layout	a function or a matrix used to specify the layout of the graphs that will be plotted. By default the layout.circle function is used;
...	further graphical parameters used to plot the graphs. See package **igraph** for more details.

Details

For a given value of the tuning parameter, specified by the argument `rhoid`, `gplot.fglasso` shows the graphs estimated at the time points `tp[1]` and `tp[2]`. By convention, the graph associated to the sub matrix $K_{tp[1],tp[2]}$ is represented by a directed graph where a directed edge is drawn by an arrow from a vertex in the first graph pointing forwards a vertex in the second graph.

Value

`gplot.fglasso` returns a list with components:

`graph.tp1`	an object with class `igraph` representing the undirected graph estimated at the time point `tp[1]`;
`graph.tp2`	an object with class `igraph` representing the undirected graph estimated at the time point `tp[2]`;
`graph.net`	an object with class `igraph` representing the directed graph associated to the submatrix $K_{tp[1],tp[2]}$;
`layout`	the matrix used to specify the placement of the vertices.

Author(s)

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

See Also

`fglasso` function.

Examples

```
N <- 50
tp <- 3
p <- 3
X <- matrix(rnorm(N * p * tp), N, tp * p)
S <- crossprod(X) / N
model <- list(lag0 = c(s = "c", n = "ut"), lag1 = c(s = "t", n = "t"))
out.fglasso <- fglasso(S = S, model = model, tp = tp, p = p)
gplot(out.fglasso, rhoid = 50, sub.tp1 = "First graph",
   sub.tp2 = "Second graph")
```

---

gplot.sglasso *Plotting Sparse Graphs*

Description

`gplot.sglasso` shows the sequence of graphs estimated by `sglasso`.

Usage

```
## S3 method for class 'sglasso'
gplot(object, rhoid, layout = layout.circle, ...)
```

Arguments

`object`	fitted `sglasso` object;
`rhoid`	vector of integers used to specificy the ρ-values used to fit the sglasso model. By default `gplot.sglasso` shows the sequence of graphs estimated by `sglasso`. Only topologically different graphs are plotted;
`layout`	a function or a matrix used to specify the layout of the graphs that will be plotted. By default the `layout.circle` function is used;
`...`	further graphical parameters used to plot the graphs. See package **igraph** for more details.

Details

`gplot.sglasso` shows the sequence of topologically different graphs estimated by `sglasso`. To specify the layout of the graphs, the user can use any layout function available in the R package **igraph**. The user can also specify the placement of the vertices by a matrix with two columns and the same number of rows as the number of vertices.

Author(s)

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

See Also

`sglasso` function.

Examples

```
N <- 100
p <- 5
X <- matrix(rnorm(N * p), N, p)
S <- crossprod(X)/N
mask <- outer(1:p, 1:p, function(i,j) 0.5^abs(i-j))
mask[1,5] <- mask[1,4] <- mask[2,5] <- NA
mask[5,1] <- mask[4,1] <- mask[5,2] <- NA
out.sglasso_path <- sglasso(S, mask, tol = 1.0e-13)
gplot(out.sglasso_path)
gplot(out.sglasso_path, rhoid = 1:5)
```

Kh *Extract Sparse Structured Precision Matrices*

Description

Function Kh computes the sequence of sparse structured precision matrices estimated by sglasso function.

Usage

```
Kh(object, rho)
```

Arguments

object	fitted sglasso object;
rho	a subset of the values of the tuning parameter used in sglasso to compute the solution path. By default, the entire sequence of estimated sparse structured precision matrices is returned.

Value

Kh returns a named list containing the sequence of estimated sparse structured precision matrices.

Author(s)

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

See Also

sglasso function.

Examples

```
N <- 100
p <- 5
X <- matrix(rnorm(N * p), N, p)
S <- crossprod(X) / N
mask <- outer(1:p, 1:p, function(i, j) 0.5^abs(i - j))
out.sglasso_path <- sglasso(S, mask, nrho = 5, tol = 1.0e-13)
out.sglasso_path
Kh(out.sglasso_path)
rho <- out.sglasso_path$rho[3]
out.sglasso_single <- sglasso(S, mask, nrho = 1, min_rho = rho,
   tol = 1.0e-13, algorithm = "ccm")
Kh(out.sglasso_single)
```

klcv *Cross-Validated Kullback-Leibler Divergence*

Description

Model selection criterion based on the leave-one-out cross-validated Kullback-Leibler divergence.

Usage

```
klcv(object, X, scale = 1)
```

Arguments

object	fitted sglasso/fglasso object;
X	the matrix used to compute the empirical variance/covariance matrix. Its dimension is N $\times$ p, where p is the number of random variables and N is the samlpe size;
scale	scalar value used to scale the estimated degrees-of-freedom. See below for more details.

Details

klcv function implements the leave-one-out cross-validate Kullback-Leibler divergence criterion proposed in Vujacic et al. (2015). For l_1-penalized Gaussian Graphical Models this measure of goodness-of-fit has the following form

$$klcv(\rho) = -\frac{\ell(\hat{K}(\rho))}{N} + \frac{\texttt{scale}}{2N} gdf(\hat{K}(\rho)),$$

where $\hat{K}(\rho)$ is the glasso estimate of the concentration matrix, $\ell(\hat{K}(\rho))$ is the corresponding value of the log-likelihood function, scale is a scale factor for the complexity part, i.e. $gdf(\hat{K}(\rho))$, which is defined as

$$gdf(\hat{K}(\rho)) = \frac{1}{N-1} \sum_{k=1}^{N} vec\{(\hat{K}(\rho)^{-1} - S_k) \circ 1_\rho\}' vec[\hat{K}(\rho)\{(S - S_k) \circ 1_\rho\}\hat{K}(\rho)].$$

In the previous expression S is the empirical variance/covariance matrix, $S_k = X_k X_k'$, 1_ρ is a matrix with entries $I(\hat{k}_{ij}(\rho) \neq 0)$ and $\circ$ is the Hadamard product operator.

Value

klcv returns an S3 object with calls klcv, i.e. a named list with the following components:

klcv	the vector with the leave-one-out cross-validated Kullback-Leibler divergence;
rho	the rho-values used to compute the leave-one-out cross-validated Kullback-Leibler divergence;

loglik	a vector with the log-likelihood computed for the sequence of weighted l1-penalized *RCON(V, E);*
gdf	a vector returning the generalized degrees-of-freedom;
scale	the scale value used to define the leave-one-out cross-validated Kullback-Leibler divergence;
min.klcv	minimum value of the leave-one-out cross-validated Kullback-Leibler divergence;
rho.opt	the rho-value corresponding to minimum leave-one-out cross-validated Kullback-Leibler divergence;
rhoid	the index of the rho-value identified by the leave-one-out cross-validated Kullback-Leibler divergence.

Author(s)

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

References

Vujacic, I., Abbruzzo, A. and Wit, E. C. (2015) A computationally fast alternative to cross-validation in penalized Gaussian graphical models. *J. Stat. Comput. Simul.*

See Also

sglasso, loglik functions and plot.klcv method.

Examples

```
N <- 100
p <- 5
X <- matrix(rnorm(N * p), N, p)
S <- crossprod(X) / N
mask <- outer(1:p, 1:p, function(i,j) 0.5^abs(i-j))
mask[1,5] <- mask[1,4] <- mask[2,5] <- NA
mask[5,1] <- mask[4,1] <- mask[5,2] <- NA
out.sglasso_path <- sglasso(S, mask, tol = 1.0e-13)
out.klcv <- klcv(out.sglasso_path, X)
out.klcv
```

---

loglik *Extract Log-Likelihood*

Description

This function extracts the log-likelihood for the sequence of weighted l1-penalized *RCON(V, E)* models estimated by sglasso function.

Usage

```
loglik(object, N = 2)
```

Arguments

`object`	a fitted `sglasso` object;
`N`	sample size. Default value is 2 to remove the constant term in the log-likelihood function. See below for more details.

Details

Denoted with $\psi = (\eta', \theta')'$ the parameter vector of the structured concentration matrix $K(\psi)$, the log-likelihood function of the *RCON(V, E)* model is equal, up to a constant, to the following expression

$$\ell(\psi) = \frac{N}{2}[\log detK(\psi) - tr\{SK(\psi)\}],$$

where $S = N^{-1}\sum_{i=1}^{N} X_i X_i^T$, N is the sample size and X_i is the ith observed p-dimensional vector. Denoted with $\hat{\psi} = (\hat{\eta}', \hat{\theta}')'$ the sglasso estimates, straightforward algebra shows that

$$\ell(\hat{\psi}) = \frac{N}{2}[\log detK(\hat{\psi}) - p + \rho \sum_{m=1}^{S} w_m |\hat{\theta}_m|],$$

where ρ is the tuning parameter and w_m are the weights used to define the weighted l1-norm.

Value

`loglik` returns a vector containing the log-likelihood computed for the sequence of weighted l1-penalized *RCON(V, E)*.

Author(s)

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

See Also

`summary.sglasso` method and `sglasso` function.

Examples

```
N <- 100
p <- 5
X <- matrix(rnorm(N * p), N, p)
S <- crossprod(X) / N
mask <- outer(1:p, 1:p, function(i, j) 0.5^abs(i-j))
out.sglasso_path <- sglasso(S, mask, nrho = 5, tol = 1.0e-13)
out.sglasso_path
loglik(out.sglasso_path, N = N)
rho <- out.sglasso_path$rho[3]
```

```
out.sglasso_single <- sglasso(S, mask, nrho = 1, min_rho = rho,
   tol = 1.0e-13, algorithm = "ccm")
loglik(out.sglasso_single, N = N)
```

`neisseria` *Neisseria Data Set*

Description

This data set contains the gene expression data from a high-resolution time-course experiment besed on the sequenced Neisseria meningitidis serogroup strain B strain MC58. Specifically, the expression level of ten genes is measured at ten different time points. Each column is standardized to have zero mean and standard deviation equal to one.

Usage

```
data("neisseria")
```

`plot.klcv` *Plot Method for Leave-One-Out Cross-Validated Kullback-Leibler Divergence*

Description

`plot.klcv` produces a plot to study the sequence of leave-one-out cross-validated Kullback-Leibler divergences computed by `klcv`.

Usage

```
## S3 method for class 'klcv'
plot(x, ...)
```

Arguments

`x`	fitted `klcv` object;
`...`	other parameters to be passed through the plotting function.

Details

This method function produces a plot showing the sequence of leave-one-out cross-validated Kullback-Leibler as function of the tuning parameter rho. The optimal value of the tuning parameter is identified by a vertical dashed line.

Author(s)

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

See Also

klcv function.

Examples

```
N <- 100
p <- 5
X <- matrix(rnorm(N * p), N, p)
S <- crossprod(X) / N
mask <- outer(1:p, 1:p, function(i,j) 0.5^abs(i-j))
mask[1,5] <- mask[1,4] <- mask[2,5] <- NA
mask[5,1] <- mask[4,1] <- mask[5,2] <- NA
out.sglasso_path <- sglasso(S, mask, tol = 1.0e-13)
out.klcv <- klcv(out.sglasso_path, X)
plot(out.klcv)
```

---

plot.sglasso *Plot Method for the Weighted l1-Penalized RCON(V, E) Model*

Description

plot.sglasso produces two plots to study the sequence of models estimates by sglasso or fglasso.

Usage

```
## S3 method for class 'sglasso'
plot(x, ...)
```

Arguments

x	fitted sglasso/fglasso object;
...	other parameters to be passed through the plotting function.

Details

This function produces two different plots. The first one shows the path of the estimated parameters as function of the tuning parameter ρ. In the same way, the second plot shows the path of the weighted scores as function of ρ.

Author(s)

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

See Also

sglasso function and summary.sglasso method.

Examples

```
N <- 100
p <- 5
X <- matrix(rnorm(N * p), N, p)
S <- crossprod(X)/N
mask <- outer(1:p, 1:p, function(i,j) 0.5^abs(i-j))
mask[1,5] <- mask[1,4] <- mask[2,5] <- NA
mask[5,1] <- mask[4,1] <- mask[5,2] <- NA
out.sglasso_path <- sglasso(S, mask, tol = 1.0e-13)
plot(out.sglasso_path)
```

---

sglasso *Lasso Method for the RCON(V, E) Models*

Description

Fit the weighted l1-penalized *RCON(V, E)* models using a cyclic coordinate algorithm.

Usage

```
sglasso(S, mask, w = NULL, flg = NULL, min_rho = 1.0e-02, nrho = 50,
        nstep = 1.0e+05, algorithm = c("ccd","ccm"), truncate = 1e-05,
        tol = 1.0e-03)
```

Arguments

`S`	the empirical variance/covariance matrix;
`mask`	a symmetric matrix used to specify the equality constraints on the entries of the concentration matrix. See the example bellow for more details;
`w`	a vector specifying the weights used to compute the weighted l1-norm of the parameters of the *RCON(V, E)* model;
`flg`	a logical vector used to specify if a parameter is penalized, i.e., if `flg[i] = TRUE` then the i-th parameter is penalized, otherwise (`flg[i] = FALSE`) the maximum likelihood estimate is computed;
`min_rho`	last value of the sequence of tuning parameters used to compute the sglasso solution path. If `nrho = 1`, then `min_rho` is the value used to compute the sglasso estimate. Default value is 1.0e-02;
`nrho`	number of tuning parameters used to compute the sglasso solution path. Default is 50;
`nstep`	nonnegative integer used to specify the maximun number of iterations of the two cyclic coordinate algorithms. Default is 1.0e+05;
`algorithm`	character by means of to specify the algorithm used to fit the model, i.e., a cyclic coordinate descente (`ccd`) algorithm or a cyclic coordinate minimization (`ccm`) algorithm. Default is `ccd`;
`truncate`	at convergence all estimates below this value will be set to zero. Default is 1e-05;
`tol`	value used for convergence. Default value is 1.0e-05.

Details

The *RCON(V, E)* model (Hojsgaard et al., 2008) is a kind of restriction of the Gaussian Graphical Model defined using a coloured graph to specify a set of equality constraints on the entries of the concentration matrix. Roughly speaking, a coloured graph implies a partition of the vertex set into R disjoint subsets, called vertex colour classes, and a partition of the edge set into S disjoint subsets, called edge colour classes. At each vertex/edge colour class is associated a specific colour. If we denote by $K = (k_{ij})$ the concentration matrix, i.e. the inverse of the variance/covariance matrix Σ, the coloured graph implies the following equality constraints:

1. $k_{ii} = \eta_n$ for any index i belonging to the nth vertex colour class;
2. $k_{ij} = \theta_m$ for any pair (i, j) belonging to the mth edge colour class.

Denoted with $\psi = (\eta', \theta')'$ the $(R+S)$-dimensional parameter vector, the concentration matrix can be defined as

$$K(\psi) = \sum_{n=1}^{R} \eta_n D_n + \sum_{m=1}^{S} \theta_m T_m,$$

where D_n is a diagonal matrix with entries $D^n_{ii} = 1$ if the index i belongs to the nth vertex colour class and zero otherwise. In the same way, T_m is a symmetrix matrix with entries $T^m_{ij} = 1$ if the pair (i, j) belongs to the mth edge colour class. Using the previous specification of the concentration matrix, the structured graphical lasso (sglasso) estimator (Abbruzzo et al., 2014) is defined as

$$\hat{\psi} = \arg\max_{\psi} \log detK(\psi) - tr\{Sk(\psi)\} - \rho \sum_{m=1}^{S} w_m |\theta_m|,$$

where S is the empirical variance/covariance matrix, ρ is the tuning parameter used to control the ammount of shrinkage and w_m are weights used to define the weighted ℓ_1-norm. By default, the `sglasso` function sets the weights equal to the cardinality of the edge colour classes.

Value

`sglasso` returns an obejct with S3 class `"sglasso"`, i.e. a named list containing the following components:

`call`	the call that produced this object;
`nv`	number of vertex colour classes;
`ne`	number of edge colour classes;
`theta`	the matrix of the sglasso estimates. The first `nv` rows correspond to the unpenalized parameters while the remaining rows correspond to the weighted l1-penalized parameters;
`w`	the vector of weights used to define the weighted l1-norm;
`df`	`nrho`-dimensional vector of the number of estimated nonzero parameters;
`rho`	`nrho`-dimensional vector of the sequence of tuning parameters;
`grd`	the matrix of the scores;
`nstep`	nonnegative integer used to specify the maximum number of iterations of the algorithms;

nrho	number of tuning parameters used to compute the sglasso solution path;
algorithm	the algorithm used to fit the model;
truncate	the value used to set to zero the estimated parameters;
tol	a nonnegative value used to define the convergence of the algorithms;
S	the empirical variace/covariance matrix used to compute the sglasso solution path;
mask	the mask used to define the equality constraints on the entries of the concentration matrix;
n	number of interations of the algorithm;
conv	an integer value used to encode the warnings related to the algorihtms. If conv = 0 the convergence has been achieved otherwise the maximum number of iterations has been achieved.

Author(s)

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

References

Abbruzzo, A., Augugliaro, L., Mineo, A. M. and Wit, E. C. (2014) Cyclic coordinate for penalized Gaussian Graphical Models with symmetry restrictions. In *Proceeding of COMPSTAT 2014 - 21th International Conference on Computational Statistics*, Geneva, August 19-24, 2014.

Hojsgaard, S. and Lauritzen, S. L. (2008) Graphical gaussian models with edge and vertex symmetries. *J. Roy. Statist. Soc. Ser. B.*, Vol. **70(5)**, 1005–1027.

See Also

summary.sglasso, plot.sglasso gplot.sglasso and methods.

The function Kh extracts the estimated sparse structured concentration matrices.

Examples

```
##########################################################
# sglasso solution path
#
## structural zeros:
## there are two ways to specify structural zeros which are
## related to the kind of mask. If mask is a numeric matrix
## NA is used to identify the structural zero. If mask is a
## character matrix then the structural zeros are specified
## using NA or ".".
N <- 100
p <- 5
X <- matrix(rnorm(N * p), N, p)
S <- crossprod(X) / N
mask <- outer(1:p, 1:p, function(i,j) 0.5^abs(i-j))
```

```
mask[1,5] <- mask[1,4] <- mask[2,5] <- NA
mask[5,1] <- mask[4,1] <- mask[5,2] <- NA
mask

out.sglasso_path <- sglasso(S, mask, tol = 1.0e-13)
out.sglasso_path

rho <- out.sglasso_path$rho[20]
out.sglasso <- sglasso(S, mask, nrho = 1, min_rho = rho, tol = 1.0e-13, algorithm = "ccm")
out.sglasso

out.sglasso_path$theta[, 20]
out.sglasso$theta[, 1]
```

`summary.sglasso` *Summarizing sglasso Fits*

Description

`summary` method for class `"sglasso"`.

Usage

```
## S3 method for class 'sglasso'
summary(object, N, k = c("bic","aic"),
        digits = max(3, getOption("digits") - 3), ...)
```

Arguments

`object`	fitted `sglasso` object;
`N`	sample size;
`k`	character/numeric argument used to specify the 'weight' of the complexity part in the measure of goodness-of-fit used to select the best model (see below for more details). Default is `k = "bic"`;
`digits`	significant digits in printout;
`...`	additional print arguments.

Details

`summary.sglasso` gives us information about the sequence of models estimated by the sglasso estimator. To select the best model, summary method uses a measure of Goodness-of-Fit (GoF) defined as follows:

$$-2\ell(\hat{\psi}) + k \times df,$$

where $\ell(\hat{\psi})$ is the log-likelihood of the estimated weighted l1-penalized *RCON(V, E)* model, df is the number of nonzero estimated parameters and k is a non-negative value used to weight the complexity part in the measure of goodness-of-fit. By default the summary method computes the BIC criterion to select the best model (`k = "bic"`). The AIC criterion can be easily computed

setting k = "aic". The user can also define other measures of goodness-of-fit specifying k as any non-negative value.

The output of the summary method is divided in two sections. First section shows the call producing the argument object followed by a data.frame. The column named rho shows the sequence of the ρ values used to compute the solution curve, while the column log-lik shows the corresponding values of the log-likelihood function. The remaining columns show the number of estimated non-zero parameters, the values of the GoF and the asscoated ranking of the estimated models. Finally, the second section shows the estimated parameters of the best model identified by the used GoF criterion. Informations about the algorithm and the corresponding convergence are also provided.

Value

A list with components table and theta_gof is silently returned. The table component is the data.frame previously described while the component theta_gof is the vector of the estimated parameters corresponding to the best models identified by the GoF criterion.

Author(s)

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

See Also

sglasso and loglik functions.

Examples

```
N <- 100
p <- 5
X <- matrix(rnorm(N * p), N, p)
S <- crossprod(X) / N
mask <- outer(1:p, 1:p, function(i,j) 0.5^abs(i-j))
mask[1,5] <- mask[1,4] <- mask[2,5] <- NA
mask[5,1] <- mask[4,1] <- mask[5,2] <- NA
out.sglasso_path <- sglasso(S, mask, tol = 1.0e-13)
summary(out.sglasso_path, N)
rho <- out.sglasso_path$rho[20]
out.sglasso <- sglasso(S, mask, nrho = 1, min_rho = rho, tol = 1.0e-13)
summary(out.sglasso, N)
```

Index

∗ **datasets**
    neisseria, 13
∗ **graph**
    fglasso, 3
    sglasso, 15
    sglasso-package, 2
∗ **models**
    fglasso, 3
    gplot, 5
    gplot.fglasso, 6
    gplot.sglasso, 7
    Kh, 9
    klcv, 10
    loglik, 11
    plot.klcv, 13
    plot.sglasso, 14
    sglasso, 15
    summary.sglasso, 18
∗ **multivariate**
    fglasso, 3
    sglasso, 15
    sglasso-package, 2
∗ **package**
    sglasso-package, 2

fglasso, 3, *7*

gplot, 5
gplot.fglasso, *6*, 6
gplot.sglasso, *6*, 7, *17*

Kh, 9, *17*
klcv, 10, *14*

loglik, *11*, 11, *19*

neisseria, 13

plot.klcv, *11*, 13
plot.sglasso, 14, *17*
print.klcv *(*klcv*)*, 10
print.sglasso *(*sglasso*)*, 15

sglasso, *5*, *8*, *9*, *11*, *12*, *14*, 15, *19*
sglasso-package, 2
summary.sglasso, *12*, *14*, *17*, 18