Package ‘sglasso’

February 20, 2015

Type Package
Title Lasso method for RCON(V,E) models
Version 1.1-0
Date 2014-09-04
Author Luigi Augugliaro
Maintainer Luigi Augugliaro <luigi.augugliaro@unipa.it>
Depends Matrix
Encoding latin1
Description RCON(V, E) models (Højsgaard, 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) algorithm and a cyclic coordinate descent (CCD) algorithm.
License GPL (>= 2)
LazyLoad yes
NeedsCompilation yes
Repository CRAN
Date/Publication 2014-09-22 17:31:46

R topics documented:
sglasso-package .................................................. 2
glasso ............................................................... 2
Kh ................................................................. 5
kcv ................................................................. 6
loglik ............................................................... 7
plot.sglasso ..................................................... 9
sglasso ............................................................ 10
summary.sglasso ................................................. 13

Index 15
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 11-penalized RCON(V, E) model. Two cyclic coordinate algorithms are implemented to compute the sglasso estimator, i.e., a cyclic coordinate minimization (CCM) algorithm and a cyclic coordinate descent (CCD) algorithm.

Details

Package: sglasso
Type: Package
Version: 1.1-0
Date: 2014-09-04
License: GPL (>=2)

Author(s)

Luigi Augugliaro

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

References


Factorial Graphical Lasso Model

Description

Fit the factorial graphical lasso 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 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 graphical model (Abbruzzo et al., to appear) is a special kind of the Graphical Gaussian Model 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

\[
K = \begin{pmatrix}
K_{1,1} & K_{1,2} & \cdots & K_{1,T} \\
K_{2,1} & K_{2,2} & \cdots & K_{2,T} \\
\vdots & \vdots & \ddots & \vdots \\
K_{T,1} & K_{T,2} & \cdots & K_{T,T}
\end{pmatrix},
\]

where \( K_{t,t} \) give information about the conditional 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.,

\[
K_{t,t+h} = \begin{pmatrix}
k_{1t,1(t+h)} & 0 & \cdots & 0 \\
0 & k_{2t,2(t+h)} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & k_{pt,p(t+h)}
\end{pmatrix} + \begin{pmatrix}
0 & k_{1t,2(t+h)} & \cdots & k_{1t,p(t+h)} \\
k_{2t,1(t+h)} & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
k_{pt,1(t+h)} & k_{pt,2(t+h)} & \cdots & 0
\end{pmatrix}.
\]

The entries of the first matrix are called **self-self conditional 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:

<table>
<thead>
<tr>
<th>Self-Self Dependence</th>
<th>Effect</th>
<th>R Code</th>
<th>Conditional Dependence</th>
<th>Effect</th>
<th>R Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>i. ( k_{it,i(t+h)} = 0 )</td>
<td>zero</td>
<td>&quot;.&quot;</td>
<td>iv. ( k_{it,j(t+h)} = 0 )</td>
<td>zero</td>
<td>&quot;.&quot;</td>
</tr>
<tr>
<td>ii. ( k_{it,i(t+h)} = s_i^h )</td>
<td>constant</td>
<td>&quot;c&quot;</td>
<td>ii. ( k_{it,j(t+h)} = n_i^h )</td>
<td>constant</td>
<td>&quot;c&quot;</td>
</tr>
<tr>
<td>iii. ( k_{it,i(t+h)} = s_i^h )</td>
<td>unit</td>
<td>&quot;u&quot;</td>
<td>iii. ( k_{it,j(t+h)} = n_i^h )</td>
<td>unit</td>
<td>&quot;u&quot;</td>
</tr>
<tr>
<td>iv. ( k_{it,i(t+h)} = s_i^{t,h} )</td>
<td>time</td>
<td>&quot;t&quot;</td>
<td>iv. ( k_{it,j(t+h)} = n_i^{t,h} )</td>
<td>time</td>
<td>&quot;t&quot;</td>
</tr>
<tr>
<td>v. ( k_{it,i(t+h)} = s_{i,t} )</td>
<td>interaction</td>
<td>&quot;ut&quot;</td>
<td>v. ( k_{it,j(t+h)} = n_{i,t}^{t,h} )</td>
<td>interaction</td>
<td>&quot;ut&quot;</td>
</tr>
</tbody>
</table>
Argument `model` is used to specify the restrictions previously described. This argument can be a named list or a matrix with dimension `nlag` x 2, where `nlag` ≤ `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

```r
> 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 remaining element are equal to zero. The `fglasso` function passes the matrix `model.mat` to the internal function `fglasso_model2mask`, i.e.,

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

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

```r
> 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 object with S3 class "sglasso". See the corresponding manual for more details.

**Author(s)**

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

**References**


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

Extract Sparse Structured Precision Matrices

Description

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

Usage

Kh(object, rho)

Arguments

object fitted sglasso object;
rho a vector specifying 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

```r
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**

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

**Arguments**

- `object` fitted `sglasso` object;
- `X` a matrix with dimension `N × p`, where `p` is the number of random variables;
- `scale` scalar value used to scale the degrees-of-freedom. See below for more details.

**Details**

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

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

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., $\text{gdf}(\hat{K}(\rho))$, which is defined as

$$ \text{gdf}(\hat{K}(\rho)) = \frac{1}{N - 1} \sum_{k=1}^{N} \text{vec}((\hat{K}(\rho)^{-1} - S_k) \odot 1_p)' \text{vec}((\hat{K}(\rho) - S) \odot 1_p) \hat{K}(\rho). $$

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

Value

klcv returns a named list with the following components:

klcv  a vector with 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 scale value used to define the leave-one-out cross-validated Kullback-Leibler divergence.

Author(s)

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

References


See Also

sglasso and loglik 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, tol = 1.0e-13)
out.klcv <- klcv(out.sglasso_path, X)
out.klcv$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 the sglasso function.

Usage

loglik(object, size = 2)
Arguments

object a fitted sglasso object;
size 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

$$\ell(\psi) = \frac{N}{2} \left[ \log \det K(\psi) - tr\{SK(\psi)\} \right],$$

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

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

where $\rho$ 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

```r
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, size = 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, size = N)
```
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.

Usage

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

Arguments

x 
fitting sglasso object;

... 
other parameters to be passed through the plotting functions.

Details

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

Author(s)

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

See Also

sglasso function and summary.sglasso method.

Examples

```r
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, 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, min_rho = 1.0e-02, nrho = 50, nstep = 1.0e+05, algorithm = c("ccd","ccm"), tol = 1.0e-05)

Arguments

S  the empirical variance/covariance matrix;
mask  a symmetric matrix used to define the structure of the concentration matrix. See the example below 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;
min_rho  last value of the sequence of tuning parameters used to compute the sglasso solution path. If the argument 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. Default is 50;
nstep  nonnegative integer used to specify the maximum number of iterations of the two cyclic coordinate algorithms. Default is 1.0e+05;
algorith  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;
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 $\Sigma$, the coloured graph implies the following equality constraints:

1. $k_{ii} = \eta_n$ for any index $i$ belonging to the $n$th vertex colour class;
2. $k_{ij} = \theta_m$ for any pair $(i, j)$ belonging to the $m$th edge colour class.
Denoted with $\psi = (\eta', \theta')'$ the $(R+S)$-dimensional parameter vector, the previous constraints imply that 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 $D_n^i = 1$ if the index $i$ belongs to the $n$th vertex colour class and zero otherwise. In the same way, $T_m$ is a symmetric matrix with entries $T_m^{ij} = 1$ if the pair $(i,j)$ belongs to the $m$th edge colour class. Using the previous specification of the concentration matrix, the structured graphical lasso (sglasso) estimator (Abbruzzo et al., 2014) is defined as

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

where $S$ is the empirical variance/covariance matrix, $\rho$ is the tuning parameter used to control the amount of shrinkage and $w_m$ are weights used to define the weighted $\ell_1$-norm. In order to stabilize the variance of the elements of the score vector at the starting point, we let $w_m = tr(T_m S T_m S)$ (default value of the argument $w$ in the sglasso function).

**Value**

sglasso returns an object 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 $\ell_1$-penalized parameters;
- **w**: the vector of weights used to define the weighted $\ell_1$-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;
- **tol**: a nonnegative value used to define the convergence of the algorithms;
- **S**: the empirical variance/covariance matrix used to compute the sglasso solution path;
- **mask**: the mask used to define the structure of the concentration matrix;
- **n**: number of iterations of the algorithm;
- **conv**: an integer value used to encode the warnings related to the algorithms. 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


See Also
summary.sglasso and plot.sglasso 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

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]
```
Description

summary method for class "sglasso".

Usage

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

Arguments

object      fitted sglasso object;
size        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 informations 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 integer used to weight the complexity part in the measure of goodness-of-fit. By default the summary method computed 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 integer.

The output of the summary method is divided in two sections. The first section shows the call that produced the argument object followed by a data.frame. The column named Sequence gives us information on how is changed the set of estimated non-zero parameters along the path. The column named rho shows the sequence of the \( \rho \) values used to compute the solution curve, while the column log-lik shows the corresponding value of the log-likelihood function. The remaining columns show the number of estimated non-zero parameters. Finally, the second section shows the estimated parameters of the best model identified by the used GoF criterion. Like for the print.dglar.s method, informations about the 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 using the GoF criterion.

Author(s)

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

See Also

sglasso and loglik functions.

Examples

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

*Topic graph
  fglasso, 2
  sglasso, 10
  sglasso-package, 2

*Topic models
  fglasso, 2
  Kh, 5
  klcv, 6
  loglik, 7
  plot.sglasso, 9
  sglasso, 10
  summary.sglasso, 13

*Topic multivariate
  fglasso, 2
  sglasso, 10
  sglasso-package, 2

*Topic package
  sglasso-package, 2

fglasso, 2
Kh, 5, 12
klcv, 6
loglik, 7, 7, 14
plot.sglasso, 9, 12
print.sglasso(sglasso), 10
sglasso, 5, 7–9, 10, 14
sglasso-package, 2
summary.sglasso, 8, 9, 12, 13