Package 'SEset'

October 12, 2022

Type Package

Title Computing Statistically-Equivalent Path Models

Version 1.0.1

Author Oisín Ryan

Maintainer Oisín Ryan <o.ryan@uu.nl>

Description Tools to compute and analyze the set of statistically-equivalent (Gaussian, linear) path models which generate the input precision or (partial) correlation matrix.
This procedure is useful for understanding how statistical network models such as the Gaussian Graphical Model (GGM) perform as causal discovery tools.
The statistical-equivalence set of a given GGM expresses the uncertainty we have about the sign, size and direction of directed relationships based on the weights matrix of the GGM alone.
The derivation of the equivalence set and its use for understanding GGMs as causal discovery tools is described by Ryan, O., Bringmann, L.F., & Schuurman, N.K. (2022) <doi:10.31234/osf.io/ryg69>.

BugReports https://github.com/ryanoisin/SEset

License GPL-3

Encoding UTF-8

LazyData true

Imports combinat, Matrix, Rdpack, stats

RdMacros Rdpack

RoxygenNote 7.1.1

Suggests qgraph

NeedsCompilation no

Repository CRAN

Date/Publication 2022-03-17 08:50:08 UTC

R topics documented:

`cov_to_path` *Path model from covariance matrix with ordering*

Description

Helper function. Takes a covariance matrix and ordering and generates a lower-triangular weights matrix.

Usage

```
cov_to_path(sigma, ordering = NULL, digits = 2)
```

Arguments

`sigma`	input matrix, with rows and columns in desired topological ordering Must be an invertible square matrix
`ordering`	character vector containing the dimension names of the input matrix in the desired ordering
`digits`	the number of digits used to round the output

Value

lower triangular matrix containing regression weights of the path model. Element ij represents the effect of X_j on X_i

See Also

`network_to_path`

find_parents — *Return parent indices from a (weighted) DAG for a given child*

Description

Return parent indices from a (weighted) DAG for a given child

Usage

```
find_parents(mat, child)
```

Arguments

mat	An $p \times p$ weights or adjacency matrix
child	Index giving the position of the child node

Value

a vector containing index numbers defining the parent nodes

References

Ryan O, Bringmann LF, Schuurman NK (upcoming). "The challenge of generating causal hypotheses using network models." *in preperation*.

See Also

r2_distribution

get_psi — *Calculate residual-covariance matrix based on a path model and covariance matrix*

Description

Takes an ordered path model and corresponding variance-covariance matrix and computes the appropriate residual covariance matrix (psi)

Usage

```
get_psi(B, sigma, digits = 3)
```

Arguments

B	input $p \times p$ linear SEM weights matrix
sigma	variance-covariance matrix of the variables
digits	how many digits to round the result to

Value

a $p \times p$ residual variance-covariance matrix

`network_to_path` *Path model from ordered precision matrix*

Description

Takes a precision matrix and generates a lower-triangular weights matrix.

Usage

```
network_to_path(omega, input_type = "precision", digits = 20)
```

Arguments

`omega`	input matrix, with rows and columns in desired topological ordering Must be an invertible square matrix
`input_type`	specifies what type of matrix ‘omega‘ is. default is "precision", other options include a matrix of partial correlations ("parcor") or a covariance matrix "covariance"
`digits`	desired rounding of the output matrix

Value

lower triangular matrix containing regression weights of the path model. Element ij represents the effect of X_j on X_i

References

Ryan O, Bringmann LF, Schuurman NK (upcoming). “The challenge of generating causal hypotheses using network models.” *in preperation*.

Shojaie A, Michailidis G (2010). “Penalized likelihood methods for estimation of sparse high-dimensional directed acyclic graphs.” *Biometrika*, **97**(3), 519–538.

Bollen KA (1989). *Structural equations with latent variables*. Oxford, England, John Wiley \& Sons.

See Also

`network_to_SEset`

Examples

```
data(riskcor)
omega <- (qgraph::EBICglasso(riskcor, n = 69, returnAllResults = TRUE))$optwi
# qgraph method estimates a non-symmetric omega matrix, but uses forceSymmetric to create
# a symmetric matrix (see qgraph:::EBICglassoCore line 65)
omega <- as.matrix(Matrix::forceSymmetric(omega)) # returns the precision matrix

B <- network_to_path(omega, digits=2)

# Path model can be plotted as a weighted DAG
pos <- matrix(c(2,0,-2,-1,-2,1,0,2,0.5,0,0,-2),6,2,byrow=TRUE)

# qgraph reads matrix elements as "from row to column"
# regression weights matrices are read "from column to row"
# path model weights matrix must be transposed for qgraph
qgraph::qgraph(t(B), labels=rownames(riskcor), layout=pos,
repulsion=.8, vsize=c(10,15), theme="colorblind", fade=FALSE)
```

`network_to_SEset` *SE-set from precision matrix*

Description

Takes a precision matrix and generates the SE-set, a set of statistically equivalent path models. Unless otherwise specified, the SEset will contain one weights matrix for every possible topological ordering of the input precision matrix

Usage

```
network_to_SEset(
  omega,
  orderings = NULL,
  digits = 20,
  rm_duplicates = FALSE,
  input_type = "precision"
)
```

Arguments

`omega`	input $p \times p$ precision matrix
`orderings`	An optional matrix of n orderings from which to generate the SE-set. Must be in the form of a $p \times n$ matrix with each column a vector of dimension names in the desired order. If unspecified, all $p!$ possible orderings are used
`digits`	desired rounding of the output weights matrices in the SE-set, in decimal places. Defaults to 20.
`rm_duplicates`	Logical indicating whether only unique DAGs should be returned

`input_type` specifies what type of matrix 'omega' is. default is "precision", other options include a matrix of partial correlations ("parcor") or a model implied covariance or correlation matrix "MIcov"

Value

a $p! \times p$ matrix containing the SE-set (or $n \times p$ matrix if a custom set of n orderings is specified). Each row represents a lower-triangular weights matrix, stacked column-wise.

References

Ryan O, Bringmann LF, Schuurman NK (upcoming). "The challenge of generating causal hypotheses using network models." *in preperation*.

Shojaie A, Michailidis G (2010). "Penalized likelihood methods for estimation of sparse high-dimensional directed acyclic graphs." *Biometrika*, **97**(3), 519–538.

Bollen KA (1989). *Structural equations with latent variables*. Oxford, England, John Wiley \& Sons.

See Also

`network_to_path`, `reorder_mat`, `order_gen`

Examples

```
# first estimate the precision matrix
data(riskcor)
omega <- (qgraph::EBICglasso(riskcor, n = 69, returnAllResults = TRUE))$optwi
# qgraph method estimates a non-symmetric omega matrix, but uses forceSymmetric to create
# a symmetric matrix (see qgraph:::EBICglassoCore line 65)
omega <- as.matrix(Matrix::forceSymmetric(omega)) # returns the precision matrix

SE <- network_to_SEset(omega, digits=3)

# each row of SE defines a path-model weights matrix.
# We can extract element 20 by writing it to a matrix
example <- matrix(SE[20,],6,6)

# Example path model can be plotted as a weighted DAG
pos <- matrix(c(2,0,-2,-1,-2,1,0,2,0.5,0,0,-2),6,2,byrow=TRUE)

# qgraph reads matrix elements as "from row to column"
# regression weights matrices are read "from column to row"
# path model weights matrix must be transposed for qgraph

qgraph::qgraph(t(example), labels=rownames(riskcor), layout=pos,
repulsion=.8, vsize=c(10,15), theme="colorblind", fade=FALSE)
```

order_gen *Generate all topological orderings*

Description

Takes a matrix and generates a matrix containing all orderings of the rows and columns

Usage

```
order_gen(omega)
```

Arguments

omega	input p-dimensional square matrix

Value

a $p \times p!$ matrix of dimension orderings. Each column represents an ordering of dimension names as character strings.

References

Chasalow S (2012). *combinat: combinatorics utilities*. R package version 0.0-8, https://CRAN.R-project.org/package=combinat.

See Also

reorder_mat, network_to_SEset

Examples

```
data(riskcor)
orderings <- order_gen(riskcor)

# Each column of orderings defines an ordering of variables
print(orderings[,1])
# in the second element, the fifth and sixth variable are switched
print(orderings[,2])
```

`path_to_network` *Precision matrix from ordered path model*

Description

Takes a path model and generates the corresponding (standardized) precision matrix or covariance matrix. The inverse of `network_to_path`.

Usage

```
path_to_network(B, psi = NULL, output = "precision")
```

Arguments

`B`	input $p \times p$ weights matrix
`psi`	variance-covariance matrix for the residuals. If NULL (the default) will impose the constraint that the variables have variance 1 and the residuals are uncorrelated
`output`	Function returns the precision ("precision") or covariance ("covariance") matrix

Value

a $p \times p$ precision or covariance matrix

References

Ryan O, Bringmann LF, Schuurman NK (upcoming). "The challenge of generating causal hypotheses using network models." *in preperation*.

Shojaie A, Michailidis G (2010). "Penalized likelihood methods for estimation of sparse high-dimensional directed acyclic graphs." *Biometrika*, **97**(3), 519–538.

Bollen KA (1989). *Structural equations with latent variables*. Oxford, England, John Wiley \& Sons.

See Also

`network_to_path`, `SEset_to_network`

propcal *Edge frequency in the SE-set*

Description

A function used to analyse the SEset results. Calculates the proportion of path models in a given SEset in which a particular edge is present

Usage

```
propcal(SEmatrix, names = NULL, rm_duplicate = TRUE, directed = TRUE)
```

Arguments

SEmatrix	An $n \times p$ matrix containing the SEset, where each row represents a $p \times p$ weights matrix stacked column-wise
names	optional character vector containing dimension names
rm_duplicate	Should duplicate weights matrices be removed from the SEset. Defaults to TRUE.
directed	If FALSE, the directionality of edges is ignored, and the output reflects in what proportion of the SEset an edge of any direction is present. If TRUE, the proportion is calculated seperately for edges of either direction. Defaults to TRUE

Value

a $p \times p$ matrix showing in what proportion particular edges are present. If directed=TRUE, elements ij denote the proportion of weights matrices containing a path from X_j to X_i. If directed=F, the output will be a symmetric matrix, with element ij denoting in what proprtion an edge of either direction connects X_i to X_j.

References

Ryan O, Bringmann LF, Schuurman NK (upcoming). "The challenge of generating causal hypotheses using network models." *in preperation*.

See Also

network_to_SEset

`r2_distribution` *Compute Controllability Distribution in the SE-set*

Description

A function used to analyse the SEset results. For each member of the SE-set, calculate the proportion of explained variance in each child node, when predicted by all of its parent nodes

Usage

```
r2_distribution(SEmatrix, cormat, names = NULL, indices = NULL)
```

Arguments

`SEmatrix`	An $n \times p$ matrix containing the SEset, where each row represents a $p \times p$ weights matrix stacked column-wise
`cormat`	A $p \times p$ matrix containing the marginal covariances or correlations
`names`	optional character vector containing dimension names
`indices`	option vector of matrix indices, indicating which variables to compute the R^2 distribution for

Value

Returns an $n \times p$ matrix of R^2 values. For each member of the SE-set, this represents the variance explained in node X_i by it's parents in that weighted DAG.

References

Ryan O, Bringmann LF, Schuurman NK (upcoming). "The challenge of generating causal hypotheses using network models." *in preperation*. Haslbeck JM, Waldorp LJ (2018). "How well do network models predict observations? On the importance of predictability in network models." *Behavior Research Methods*, **50**(2), 853–861.

See Also

`network_to_SEset`, `find_parents`

Examples

```
# first estimate the precision matrix
data(riskcor)
omega <- (qgraph::EBICglasso(riskcor, n = 69, returnAllResults = TRUE))$optwi
# qgraph method estimates a non-symmetric omega matrix, but uses forceSymmetric to create
# a symmetric matrix (see qgraph:::EBICglassoCore line 65)
omega <- as.matrix(Matrix::forceSymmetric(omega)) # returns the precision matrix

SEmatrix <- network_to_SEset(omega, digits=3)
```

```
r2set  <- r2_distribution(SEmatrix, cormat = riskcor, names = NULL, indices = c(1,3,4,5,6))
# Plot results
apply(r2set,2,hist)
# For ggplot format, execute
# r2set <- tidyr::gather(r2set)
```

`reorder_mat` *Re-order rows and columns*

Description

Takes a matrix and re-orders the rows and columns to some target ordering

Usage

```
reorder_mat(matrix, names)
```

Arguments

`matrix`	input matrix to be re-arranged. Must have rows and columns named
`names`	character vector containing the dimension names of the input matrix in the desired ordering

Value

input matrix with rows and columns sorted according to names

See Also

`order_gen`, `network_to_SEset`

Examples

```
data(riskcor)

# first define an ordered vector of names
row_names <- rownames(riskcor)
row_names_new <- row_names[c(1,2,3,4,6,5)]

reorder_mat(riskcor,row_names_new)

# The fifth and sixth row and column have been switched
print(riskcor)
```

riskcor *Cognitive risk sample correlation matrix*

Description

Reported sample correlation matrix from a cross-sectional study on cognitive risk and resilience factors in remitted depression patients, from Hoorelebeke, Marchetti, DE Schryver and Koster (2016). The study was conducted with 69 participants, and the correlation matrix consists of six variables. The variables are as follows:

Usage

```
data(riskcor)
```

Format

A 6 by 6 correlation matrix

Details

* 'BRIEF_WM': working memory complaints, a self-report measure of perceived cognitive control * 'PASAT_ACC': PASAT accuracy, performance on behavioural measure of congitive control * 'Adapt ER': self-report adaptive emotion regulation strategies * 'Maladapt ER': self-report maladaptive emotion regulation strategies * 'Resilience': self-report resilience * 'Resid Depress': self-report residual depressive symptoms

Source

<https://ars.els-cdn.com/content/image/1-s2.0-S0165032715313252-mmc1.pdf>

References

Hoorelbeke K, Marchetti I, De Schryver M, Koster EH (2016). "The interplay between cognitive risk and resilience factors in remitted depression: a network analysis." *Journal of Affective Disorders*, **195**, 96–104.

Examples

```
data(riskcor)
print(rownames(riskcor))
print(riskcor)
```

SEset_to_network *Precision matrices from the SEset*

Description

Takes the SE-set and calculates for each weights matrix the corresponding precision matrix. Used to check the results of network_to_SEset to assess deviations from statistical equivalence induced due to rounding, thresholding, and numerical approximations.

Usage

```
SEset_to_network(
  SEmatrix,
  order.ref = NULL,
  order.mat = NULL,
  output = "raw",
  omega = NULL
)
```

Arguments

SEmatrix	a $n \times p$ matrix containing the SE-set. The output of network_to_SEset
order.ref	an optional character vector with variable names, the reference ordering of the precision matrix.
order.mat	a $n \times p$ matrix of character strings, defining the ordering of the matrix corresponding to each row of SEmatrix. If NULL it is assumed that all orderings are included and they are generated using order_gen
output	Output as "raw" or "summary". See value below
omega	Comparision precision matrix, e.g. original input precision matrix to network_to_SEset. Only necessary if output = "summary"

Value

If output = "raw", a $n \times p$ matrix of precision matrices stacked column-wise in n rows. If output = "summary" returns a list containing the bias, MSE and RMSE for each re-calculated precision matrix, relative to comparison omega matrix supplied.

References

Ryan O, Bringmann LF, Schuurman NK (upcoming). "The challenge of generating causal hypotheses using network models." *in preperation*.

Shojaie A, Michailidis G (2010). "Penalized likelihood methods for estimation of sparse high-dimensional directed acyclic graphs." *Biometrika*, **97**(3), 519–538.

Bollen KA (1989). *Structural equations with latent variables*. Oxford, England, John Wiley \& Sons.

See Also

network_to_path, path_to_network

Index