Package: networksem (via r-universe)

March 3, 2025

Type Package

Title Network Structural Equation Modeling

Version 0.2

Maintainer Zhiyong Zhang <johnnyzhz@gmail.com>

Description Several methods have been developed to integrate structural equation modeling techniques with network data analysis to examine the relationship between network and non-network data. Both node-based and edge-based information can be extracted from the network data to be used as observed variables in structural equation modeling. To facilitate the application of these methods, model specification can be performed in the familiar syntax of the 'lavaan' package, ensuring ease of use for researchers. Technical details and examples can be found at <https://bigsem.psychstat.org>.

License GPL

Imports sna, influential, lavaan, network, igraph, methods

Depends latentnet

Encoding UTF-8

RoxygenNote 7.2.3

NeedsCompilation no

Author Zhiyong Zhang [aut, cre] (<https://orcid.org/0000-0003-0590-2196>), Ziqian Xu [aut]

Date/Publication 2025-01-31 15:20:02 UTC

Config/pak/sysreqs libglpk-dev make libicu-dev libxml2-dev

Repository https://johnnyzhz.r-universe.dev

RemoteUrl https://github.com/cran/networksem

RemoteRef HEAD

RemoteSha 99e5be55cf743b33810e0ce701b68bb336fe2dfb

Contents

`path.networksem` *Calculate a mediation effect from a networksem model*

Description

Calculate a mediation effect from a networksem model

Usage

```
path.networksem(res, predictor, mediator, outcome)
```

Arguments

`res`	a networksem output file
`predictor`	a character string of the predictor variable
`mediator`	a character string of the mediator variable
`outcome`	a character string of the outcome variable

Value

a target path, associated estimates, and z-score

`sem.net` *Fit a Structural Equation Model (SEM) with both network and non-network data by incorporating node-level network statistics as variables.*

Description

Fit a Structural Equation Model (SEM) with both network and non-network data by incorporating node-level network statistics as variables.

Usage

```
sem.net(
  model = NULL,
  data = NULL,
  netstats = NULL,
  ordered = NULL,
  sampling.weights = NULL,
  data.rescale = FALSE,
  netstats.rescale = FALSE,
  group = NULL,
  cluster = NULL,
  constraints = "",
  WLS.V = NULL,
  NACOV = NULL,
  netstats.options = NULL,
  ...
)
```

Arguments

`model`	A model string specified in lavaan model syntax that includes relationships among the network and non-network variables.
`data`	A list containing the data. The list has two named components, "network" and "nonnetwork"; "network" is a list of named adjacency matrices for the network data, and "nonnetwork" is the dataframe of non-network covariates.
`netstats`	A user-specified list of network statistics to be calculated and used in the SEM. Available options include "degree", "betweenness", "closeness", "evcent", "stresscent", and "infocent" from the "sna" package and "ivi", "hubness.score", "spreading.score" and "clusterRank" from the "influential" package.
`ordered`	Parameter same as "ordered" in the lavaan sem() function; whether to treat data as ordinal.
`sampling.weights`	Parameter same as "sampling.weights" in the lavaan sem() function; whether to apply weights to data.

`data.rescale` TRUE or FALSE, whether to rescale the whole dataset (with restructured network and nonnetwork data) to have mean 0 and standard deviation 1 when fitting it to SEM, default to FALSE.

`netstats.rescale` TRUE or FALSE, whether to rescale the network statistics to have mean 0 and standard deviation 1, default to FALSE.

`group` Parameter same as "group" in the lavaan sem() function; whether to fit a multigroup model.

`cluster` Parameter same as "cluster" in the lavaan sem() function; whether to fit a cluster model.

`constraints` Parameter same as "constraints" in the lavaan sem() function; whether to apply constraints to the model.

`WLS.V` Parameter same as "WLS.V" in the lavaan sem() function; whether to use WLS.V estimator.

`NACOV` Parameter same as "NACOV" in the lavaan sem() function; whether to use NACOV estimator.

`netstats.options` A user-specified named list with element names corresponding to the network statistics names and element values corresponding to options for that network statistics used as optional arguments in the corresponding functions in the "sna" or "influential" packages. e.g., netstats.options=list("degree"=list("cmode"="freeman"), "closeness"=list("cmode"="undirected"), "clusterRank"=list("directed"=FALSE)).

`...` Optional arguments for the sem() function.

Value

A networksem object containing the updated model specification string with the reconstructed network statistics as variables and a lavaan SEM object.

Examples

```
set.seed(100)
nsamp = 20
net <- ifelse(matrix(rnorm(nsamp^2), nsamp, nsamp) > 1, 1, 0)
mean(net) # density of simulated network
lv1 <- rnorm(nsamp)
lv2 <- rnorm(nsamp)
nonnet <- data.frame(x1 = lv1*0.5 + rnorm(nsamp),
                     x2 = lv1*0.8 + rnorm(nsamp),
                     x3 = lv2*0.5 + rnorm(nsamp),
                     x4 = lv2*0.8 + rnorm(nsamp))

model <-'
  lv1 =~ x1 + x2
  lv2 =~ x3 + x4
 net ~ lv2
  lv1 ~ net + lv2
'
```

```
data = list(network = list(net = net), nonnetwork = nonnet)
set.seed(100)
res <- sem.net(model = model, data = data, netstats = c('degree'))
summary(res)
```

`sem.net.addvar` *Compute user-specified network statistics for a specific network.*

Description

Compute user-specified network statistics for a specific network.

Usage

```
sem.net.addvar(
  model.network.stat.var.list = NULL,
  data = NULL,
  netstats = NULL,
  model.network.var.i = NULL,
  netstats.rescale = TRUE,
  netstats.options = NULL
)
```

Arguments

`model.network.stat.var.list`	a list of elements with names corresponding to the network names and values corresponding to lists of network statistics that will be calculated for the corresponding network
`data`	a list containing both the non-network and network data
`netstats`	a list of user-specified network statistics
`model.network.var.i`	the index of a network within all networks
`netstats.rescale`	a logical value indicating whether to rescale network statistics to have mean 0 and sd 1
`netstats.options`	a list with element names corresponding to the network statistics and element values corresponding to another list. The list corresponding to each network statistics has element names being the argument names for calculating the network statistics, and values being the argument values

Value

a list with the first value being the list of network statistics names and the second value being the data frame with added network statistics variables

`sem.net.addvar.influential` *Compute a list of user-specified network statistics using the "influential" package and add it to the existing data.*

Description

Compute a list of user-specified network statistics using the "influential" package and add it to the existing data.

Usage

```
sem.net.addvar.influential(
  model.network.stat.var.list,
  data,
  model.network.var.i,
  stats,
  statsname,
  netstats.rescale,
  netstats.options = NULL
)
```

Arguments

`model.network.stat.var.list`	a list of elements with names corresponding to the network names and values corresponding to lists of network statistics that will be calculated for the corresponding network
`data`	a list containing both the non-network and network data
`model.network.var.i`	an index indicating a specific network within all networks
`stats`	a network statistics that can be calculated using package "influential"
`statsname`	name of the network statistics
`netstats.rescale`	a logical value indicating whether to rescale network statistics to have mean 0 and sd 1
`netstats.options`	a list with names being the argument names for calculating the network statistics, and values being the argument values

Value

a list with the first value being the list of network statistics names and the second value being the data frame with added network statistics

`sem.net.addvar.stat` *Compute a list of user-specified network statistics values using the "sna" package and add them to the non-network data.*

Description

Compute a list of user-specified network statistics values using the "sna" package and add them to the non-network data.

Usage

```
sem.net.addvar.stat(
  model.network.stat.var.list,
  data,
  model.network.var.i,
  stats,
  statsname,
  netstats.rescale,
  netstats.options = NULL
)
```

Arguments

`model.network.stat.var.list`	a list of elements with names corresponding to the network names and values corresponding to lists of network statistics that will be calculated for the corresponding network
`data`	a list containing both the non-network and network data
`model.network.var.i`	an index indicating a specific network within all networks
`stats`	a network statistics that can be calculated using package "sna"
`statsname`	name of the network statistics
`netstats.rescale`	a logical value indicating whether to rescale network statistics to have mean 0 and sd 1
`netstats.options`	a list with names being the argument names for calculating the network statistics, and values being the argument values

Value

a list with the first value being the list of network statistics names and the second value being the data frame with added network statistics

`sem.net.edge` *Fit a Structural Equation Model (SEM) with both network and non-network data by transforming nonnetwork data into paired values corresponding to network edge values.*

Description

Fit a Structural Equation Model (SEM) with both network and non-network data by transforming nonnetwork data into paired values corresponding to network edge values.

Usage

```
sem.net.edge(
  model = NULL,
  data = NULL,
  type = "difference",
  ordered = NULL,
  sampling.weights = NULL,
  data.rescale = FALSE,
  group = NULL,
  cluster = NULL,
  netstats.rescale = FALSE,
  constraints = "",
  WLS.V = NULL,
  NACOV = NULL,
  ...
)
```

Arguments

`model`	A model string specified in lavaan model syntax that includes relationships among the network and non-network variables.
`data`	A list containing the data. The list has two named components, "network" and "nonnetwork"; "network" is a list of named adjacency matrices for the network data, and "nonnetwork" is the dataframe of non-network covariates.
`type`	Option for transforming nonnework data; "difference" for using the difference between two individuals as the edge covariate; "average" for using the average between two individuals as the edge covariate.
`ordered`	Parameter same as "ordered" in the lavaan sem() function; whether to treat data as ordinal.
`sampling.weights`	Parameter same as "sampling.weights" in the lavaan sem() function; whether to apply weights to data.
`data.rescale`	TRUE or FALSE, whether to rescale the whole dataset (with restructured network and nonnetwork data) to have mean 0 and standard deviation 1 when fitting it to SEM, default to FALSE.

group	Parameter same as "group" in the lavaan sem() function; whether to fit a multi-group model.
cluster	Parameter same as "cluster" in the lavaan sem() function; whether to fit a cluster model.
netstats.rescale	TRUE or FALSE, whether to rescale the network statistics to have mean 0 and standard deviation 1, default to FALSE.
constraints	Parameter same as "constraints" in the lavaan sem() function; whether to apply constraints to the model.
WLS.V	Parameter same as "WLS.V" in the lavaan sem() function; whether to use WLS.V estimator.
NACOV	Parameter same as "NACOV" in the lavaan sem() function; whether to use NACOV estimator.
...	Optional arguments for the sem() function.

Value

A networksem object containing the updated model specification string with the reconstructed network statistics as variables and a lavaan SEM object.

Examples

```
set.seed(100)
nsamp = 20
net <- data.frame(ifelse(matrix(rnorm(nsamp^2), nsamp, nsamp) > 1, 1, 0))
mean(net) # density of simulated network
lv1 <- rnorm(nsamp)
lv2 <- rnorm(nsamp)
nonnet <- data.frame(x1 = lv1*0.5 + rnorm(nsamp),
                     x2 = lv1*0.8 + rnorm(nsamp),
                     x3 = lv2*0.5 + rnorm(nsamp),
                     x4 = lv2*0.8 + rnorm(nsamp))

model <-'
  lv1 =~ x1 + x2
  lv2 =~ x3 + x4
  lv1 ~ net
  lv2 ~ lv1
'
data = list(network = list(net = net), nonnetwork = nonnet)
set.seed(100)
res <- sem.net.edge(model = model, data = data, type = 'difference')
summary(res)
```

`sem.net.edge.lsm` *Fit a Structural Equation Model (SEM) with both network and non-network data by transforming nonnetwork data into paired values corresponding to network latent distance pairs.*

Description

Fit a Structural Equation Model (SEM) with both network and non-network data by transforming nonnetwork data into paired values corresponding to network latent distance pairs.

Usage

```
sem.net.edge.lsm(
  model = NULL,
  data = NULL,
  type = "difference",
  latent.dim = 2,
  data.rescale = FALSE,
  ordered = NULL,
  sampling.weights = NULL,
  group = NULL,
  cluster = NULL,
  netstats.rescale = FALSE,
  constraints = "",
  WLS.V = NULL,
  NACOV = NULL,
  ...
)
```

Arguments

`model`	A model string specified in lavaan model syntax that includes relationships among the network and non-network variables.
`data`	A list containing the data. The list has two named components, "network" and "nonnetwork"; "network" is a list of named adjacency matrices for the network data, and "nonnetwork" is the dataframe of non-network covariates.
`type`	"difference" for using the difference between the network statistics of the two actors as the edge covariate; "average" for using the average of the network statistics of the two actors as the edge covariate.
`latent.dim`	The number of network latent dimensions to use in extracting latent positions of network nodes.
`data.rescale`	TRUE or FALSE, whether to rescale the whole dataset (with restructured network and nonnetwork data) to have mean 0 and standard deviation 1 when fitting it to SEM, default to FALSE.
`ordered`	Parameter same as "ordered" in the lavaan sem() function; whether to treat data as ordinal.

`sampling.weights`
: Parameter same as "sampling.weights" in the lavaan sem() function; whether to apply weights to data.

`group`
: Parameter same as "group" in the lavaan sem() function; whether to fit a multi-group model.

`cluster`
: Parameter same as "cluster" in the lavaan sem() function; whether to fit a cluster model.

`netstats.rescale`
: TRUE or FALSE, whether to rescale the network statistics to have mean 0 and standard deviation 1, default to FALSE.

`constraints`
: Parameter same as "constraints" in the lavaan sem() function; whether to apply constraints to the model.

`WLS.V`
: Parameter same as "WLS.V" in the lavaan sem() function; whether to use WLS.V estimator.

`NACOV`
: Parameter same as "NACOV" in the lavaan sem() function; whether to use NACOV estimator.

`...`
: Optional arguments for the sem() function.

Value

A networksem object containing the updated model specification string with the reconstructed network statistics as variables, a lavaan SEM output object, and a latentnet ergm object.

Examples

```
set.seed(10)
nsamp = 20
lv1 <- rnorm(nsamp)
net <- ifelse(matrix(rnorm(nsamp^2) , nsamp, nsamp) > 1, 1, 0)
lv2 <- rnorm(nsamp)
nonnet <- data.frame(x1 = lv1*0.5 + rnorm(nsamp),
                     x2 = lv1*0.8 + rnorm(nsamp),
                     x3 = lv2*0.5 + rnorm(nsamp),
                     x4 = lv2*0.8 + rnorm(nsamp))

model <-'
  lv1 =~ x1 + x2
  lv2 =~ x3 + x4
  net ~ lv1
  lv2 ~ net
'
data = list(network = list(net = net), nonnetwork = nonnet)
set.seed(100)
res <- sem.net.edge.lsm(model = model, data = data, latent.dim = 1)
summary(res)
```

`sem.net.lsm` *Fit a Structural Equation Model (SEM) with both network and non-network data by incorporating network latent positions as variables.*

Description

Fit a Structural Equation Model (SEM) with both network and non-network data by incorporating network latent positions as variables.

Usage

```
sem.net.lsm(
  model = NULL,
  data = NULL,
  latent.dim = 2,
  ordered = NULL,
  sampling.weights = NULL,
  data.rescale = FALSE,
  netstats.rescale = FALSE,
  group = NULL,
  cluster = NULL,
  constraints = "",
  WLS.V = NULL,
  NACOV = NULL,
  ...
)
```

Arguments

`model`	A model string specified in lavaan model syntax that includes relationships among the network and non-network variables.
`data`	A list containing the data. The list has two named components, "network" and "nonnetwork"; "network" is a list of named adjacency matrices for the network data, and "nonnetwork" is the dataframe of non-network covariates.
`latent.dim`	The number of network latent dimensions to use in extracting latent positions of network nodes.
`ordered`	Parameter same as "ordered" in the lavaan sem() function; whether to treat data as ordinal.
`sampling.weights`	Parameter same as "sampling.weights" in the lavaan sem() function; whether to apply weights to data.
`data.rescale`	TRUE or FALSE, whether to rescale the whole dataset (with restructured network and nonnetwork data) to have mean 0 and standard deviation 1 when fitting it to SEM, default to FALSE.

`netstats.rescale`
: TRUE or FALSE, whether to rescale the network statistics to have mean 0 and standard deviation 1, default to FALSE.

`group`
: Parameter same as "group" in the lavaan sem() function; whether to fit a multi-group model.

`cluster`
: Parameter same as "cluster" in the lavaan sem() function; whether to fit a cluster model.

`constraints`
: Parameter same as "constraints" in the lavaan sem() function; whether to apply constraints to the model.

`WLS.V`
: Parameter same as "WLS.V" in the lavaan sem() function; whether to use WLS.V estimator.

`NACOV`
: Parameter same as "NACOV" in the lavaan sem() function; whether to use NACOV estimator.

`...`
: Optional arguments for the sem() function.

Value

A networksem object containing the updated model specification string with the reconstructed network statistics as variables, a lavaan SEM output object, and a latentnet ergmm object.

Examples

```
set.seed(10)
nsamp = 20
net <- ifelse(matrix(rnorm(nsamp^2), nsamp, nsamp) > 1, 1, 0)
mean(net) # density of simulated network
lv1 <- rnorm(nsamp)
lv2 <- rnorm(nsamp)
nonnet <- data.frame(x1 = lv1*0.5 + rnorm(nsamp),
                     x2 = lv1*0.8 + rnorm(nsamp),
                     x3 = lv2*0.5 + rnorm(nsamp),
                     x4 = lv2*0.8 + rnorm(nsamp))

model <-'
  lv1 =~ x1 + x2
  lv2 =~ x3 + x4
  net ~ lv2
  lv1 ~ net + lv2
'
data = list(network = list(net = net), nonnetwork = nonnet)
set.seed(100)
res <- sem.net.lsm(model = model, data = data, latent.dim = 2)
summary(res)
```

summary.networksem *Summarize output from networksem functions includeing sem.net, sem.net.lsm, sem.net.edge, sem.net.edge.lsm.*

Description

Summarize output from networksem functions includeing sem.net, sem.net.lsm, sem.net.edge, sem.net.edge.lsm.

Usage

```
## S3 method for class 'networksem'
summary(object, ...)
```

Arguments

object	A networksem output object.
...	Optional arguments.

Value

a summary sheet of the networksem output.

Index