Package 'nemBM'

July 22, 2025

Type Package

Title Using Network Evolution Models to Generate Networks with Selected Blockmodel Type

Version 1.00.01

Description To study network evolution models and different blockmodeling approaches. Various functions enable generating (temporal) networks with a selected blockmodel type, taking into account selected local network mechanisms. The development of this package is financially supported the Slovenian Research Agency (www.arrs.gov.si) within the research program P5<96>0168 and the research project J5-2557 (Comparison and evaluation of different approaches to blockmodeling dynamic networks by simulations with application to Slovenian coauthorship networks).

Depends ergm, blockmodeling

License GPL-2

Encoding UTF-8

RoxygenNote 7.2.0

NeedsCompilation no

Author Marjan Cugmas [aut, cre], Aleš Žiberna [aut]

Maintainer Marjan Cugmas <marjan.cugmas@fdv.uni-lj.si>

Repository CRAN

Date/Publication 2022-12-14 13:10:02 UTC

Contents

`assortativity` *Assortativity mechanism*

Description

Calculate the normalized network statistic according to the assortativity mechanism.

Usage

```
assortativity(X, actor)
```

Arguments

`X`	Binary network; of class `matrix`.
`actor`	A unit (actor; row/column number), which have an opportunity to change a link.

Details

The function returns the value 1 when actor (i.e. ego) and alter do not differ in the number of incoming ties. Otherwise, lower values indicate higher difference in the number of incoming ties between the actor and alter.

Value

A vector with the assortativity mechanism, cacluated between the actor and other units.

Author(s)

Marjan Cugmas and Aleš Žiberna

References

Cugmas, M., & Žiberna, A. (2022). Approaches to blockmodeling dynamic networks: a Monte Carlo simulation study. Social Networks, in print

Examples

```
X <- matrix(sample(c(0,1), size = 9**2, replace = TRUE), nrow = 9)
mutuality(X, actor = 2)
```

---

chooseBlockRow *Sum of squared error across blocks*

Description

The actor choose the block (i.e., column in an image matrix) in which he will change a link, based on the difference between the density of his out-degrees by blocks and the ideal block density.

Usage

```
chooseBlockRow(X, actor, partition, M, loops, randomBlock = FALSE)
```

Arguments

X	Binary network; of class matrix.
actor	A unit (actor; row/column number), which have an opportunity to change a link.
partition	A partition in a vector format. Each unique value (positive integers) represents one cluster.
M	Image matrix with block densities.
loops	Wheter loops are allowed or not.
randomBlock	How to select a block; the one with the highest difference (FALSE, default), proportionally to the differences (linear) or squared differences (square).

Value

A vector with two elements: block (selected block number) and sign (wheter the selected block is too sparse (-1) or too dense (+1)).

Author(s)

Marjan Cugmas

Examples

```
X <- matrix(sample(c(0,1), size = 9**2, replace = TRUE), nrow = 9)
diag(X) <- 0
M <- matrix(c(0.1, 0.4, 0.5, 0.3), nrow = 2)
partition <- c(1, 2, 2, 1, 1, 2, 2, 2, 1)
chooseBlockRow(X = X, actor = 3, partition = partition,
M = M, loops = FALSE, randomBlock = "square")
```

genNetworkLE *Relocating Links algorithm (RL algorithm)*

Description

Generate network with a selected blockmodel and level or errors. See details section.

Usage

```
genNetworkLE(BM = BM, LE = 0.4, size = NULL, symmetric = FALSE)
```

Arguments

BM	An image matrix of a blockmodel; of class matrix with possible values "nul" and "com".
LE	Desired level of errors.
size	A vector with the values specifying clusters' sizes. The number of elements of this vector must be the same as the number of clusters specified by an image matrix.
symmetric	Wheter a symmetric network should be generated.

Details

The level of errors (LE) is used to simulate the extent of inconsistencies in blockmodels. It is defined on a scale between 0 and 1, where 0 corresponds to an ideal blockmodel, and 1 corresponds to a totally randomised network with the same density as in the ideal blockmodel.

Value

A binary network (of class matrix) with selected blockmodel type and level of errors.

Author(s)

Marjan Cugmas

References

Cugmas, M., Žiberna, A., & Ferligoj, A. (2021). The Relative Fit measure for evaluating a blockmodel. Statistical Methods & Applications, 30(5), 1315-1335.

Examples

```
cohesiveBM <- rbind(c("com", "nul"), c("nul", "com"))
network <- genNetworkLE(BM = cohesiveBM, LE = 0.5, size = c(5, 3))
```

`globalDensity` *Network density based on an image matrix and a partition*

Description

Based on an image matrix and a partition it calculate the density of a whole network.

Usage

```
globalDensity(M, partition)
```

Arguments

`M`	Image matrix with block densities.
`partition`	A partition in a vector format. Each unique value (positive integers) represents one cluster.

Value

Density of a whole network (a single value).

Author(s)

Marjan Cugmas

Examples

```
M <- matrix(c(0.1, 0.4, 0.5, 0.3), nrow = 2)
partition <- c(1, 2, 2, 1, 1, 2, 2, 2, 1)
globalDensity(M = M, partition = partition)
```

`mutuality` *Mutuality mechanism*

Description

Calculate the normalized network statistic according to the mutuality mechanism.

Usage

```
mutuality(X, actor)
```

Arguments

`X`	Binary network; of class `matrix`.
`actor`	A unit (actor; row/column number), which have an opportunity to change a link.

Value

A vector with the normalized mutuality mechanism, cacluated between the actor and other units.

Author(s)

Marjan Cugmas and Aleš Žiberna

References

- Cugmas, M., Žiberna, A., & Ferligoj, A. (2019). Mechanisms generating asymmetric core-cohesive blockmodels. Advances in Methodology and Statistics, 16(1), 17-41.
- Cugmas, M., & Žiberna, A. (2022). Approaches to blockmodeling dynamic networks: a Monte Carlo simulation study. Social Networks, in print.

Examples

```
X <- matrix(sample(c(0,1), size = 9**2, replace = TRUE), nrow = 9)
mutuality(X, actor = 2)
```

`nem` *Generating networks according to the selected local network mechanisms*

Description

It generates random network considering the selected local network mechanisms.

Usage

```
nem(X, formula, theta, k = 5000, q, b = 0.25)
```

Arguments

`X`	Initial network; of class `matrix`.
`formula`	The list of local netork mechanisms to be considered.
`theta`	A vector with the mechanisms' weights/strengths.
`k`	The number of iterations.
`q`	The probability of establishing a link.
`b`	The share of alters among which an actor (i.e., ego) chooses to create or break a tie.

Value

The list with the following elements:

- `initialNetwork` - Initial network; of class `matrix`.
- `finalNetwork` - Final (generated) network; of class `matrix`.
- `formula` - The list of functions that define mechanisms used.
- `theta` - A vector with the mechanisms' weights/strengths used.
- `k` - The number of iterations.
- `q` - The probability of establishing a link.
- `b` - The share of alters among which an actor (i.e., ego) chooses to create or break a tie.

Author(s)

Marjan Cugmas and Aleš Žiberna

References

Cugmas, M., Žiberna, A., & Ferligoj, A. (2019). Mechanisms generating asymmetric core-cohesive blockmodels. Advances in Methodology and Statistics, 16(1), 17-41.

Examples

```
formula <- list(mutuality, popularity, assortativity)
X <- matrix(sample(c(0,1), size = 9**2, replace = TRUE), nrow = 9)
nem(X = X, formula = formula, theta = c(1, 1, 1), k = 100, q = 0.25)
```

`nemBM` *Network evolution model with a prespecified blockmodel type and partition*

Description

Generates an asymmetric network with a selected blockmodel type and partition. Considers local network mechanisms when creating links within blocks. Does not enable considering incomers and outgoers.

Usage

```
nemBM(X = X, partition, M, formula, theta, k = 10000, loops = FALSE)
```

Arguments

X	Initial binary network; of class matrix.
partition	A desired partition in a vector format. Each unique value (positive integers) represents one cluster.
M	Desired image matrix with block densities.
formula	The list of local netork mechanisms to be considered.
theta	A vector with the mechanisms' weights/strengths.
k	The number of iterations.
loops	Wheter loops are allowed or not (default FALSE).

Value

The list with the following elements:

- initialNetwork - Initial network; of class matrix.
- finalNetwork - Final (generated) network; of class matrix.
- formula - The list of functions that define mechanisms used.
- theta - A vector with the mechanisms' weights/strengths used.
- ERR - Sum of squared differences between the desired and empirical densities across blocks; for each iteration.
- iterations - The number of iterations.
- loops - Wheter loops were allowed.
- M - The desired (specified) image matrix.
- partition - The partition.
- density - Network density at each iteration.
- timeElapsed - Running time.

Author(s)

Marjan Cugmas and Aleš Žiberna

References

Cugmas, M., & Žiberna, A. (2022). Approaches to blockmodeling dynamic networks: a Monte Carlo simulation study. Social Networks, in print.

Examples

```
formula <- list(mutuality, popularity, OTPtransitivity)
X <- matrix(sample(c(0,1), size = 9**2, replace = TRUE), nrow = 9)
diag(X) <- 0
M <- matrix(c(0.1, 0.8, 0.1, 0.5), nrow = 2)
partition <- c(1, 2, 2, 1, 1, 2, 2, 2, 1)
res <- nemBM(X = X, partition = partition, formula = formula,
theta = c(1, 1, 1), M = M, k = 100, loops = FALSE)
```

nemSym *Generating symmetric networks according to the selected local network mechanisms*

Description

It generates random network considering the selected local network mechanisms.

Usage

```
nemSym(X, formula, theta, k = 5000, q)
```

Arguments

X	Initial network; of class matrix.
formula	The list of local netork mechanisms to be considered.
theta	A vector with the mechanisms' weights/strengths.
k	The number of iterations.
q	The probability of establishing a link (i.e. expected/desired density).

Value

The list with the following elements:

- initialNetwork - Initial network; of class matrix.
- finalNetwork - Final (generated) network; of class matrix.
- formula - The list of functions that define mechanisms used.
- theta - A vector with the mechanisms' weights/strengths used.
- k - The number of iterations.
- q - The probability of establishing a link.

Author(s)

Marjan Cugmas and Aleš Žiberna

References

Cugmas, M., DeLay, D., Žiberna, A., & Ferligoj, A. (2020). Symmetric core-cohesive blockmodel in preschool children's interaction networks. PloS one, 15(1), e0226801.

Examples

```
formula <- list(popularity, assortativity)
X <- matrix(sample(c(0,1), size = 9**2, replace = TRUE), nrow = 9)
diag(X) <- 0
nemSym(X = X, formula = formula, theta = c(1, 1), k = 100, q = 0.25)
```

`nemSymBMinout` *Network evolution model with a prespecified blockmodel type and partition (symmetric networks with incomers and outgoers)*

Description

Generate a symmetric network with a selected blockmodel type and partition with a specified number of incomers and outgoers. Considers local network mechanisms when creating links within blocks.

Usage

```
nemSymBMinout(
  X = X,
  partition = partition,
  M = M,
  formula = NULL,
  theta = NULL,
  nin = 5,
  nout = 20,
  minClusterSize = 5,
  k = 1000,
  loops = FALSE,
  randomizeP = 0,
  randomSD = 0.02
)
```

Arguments

`X`	Initial binary network; of class `matrix`.
`partition`	A desired partition in a vector format. Each unique value (positive integers) represents one cluster.
`M`	Desired image matrix with block densities.
`formula`	The list of local netork mechanisms to be considered.
`theta`	A vector with the mechanisms' weights/strengths.
`nin`	Number of incomers.
`nout`	Number of outgoers.
`minClusterSize`	Minimum cluster size.
`k`	Number of iterations.
`loops`	Wheter loops are allowed or not (default `FALSE`).
`randomizeP`	The share of units to be randomly relocated between clusters.
`randomSD`	The srandard deviation of a normal distribution form which the random part of weighed network statistics is sampled.

Value

The list with the following elements:

- `initialNetwork` - Initial network; of class `matrix`.
- `finalNetwork` - Final (generated) network; of class `matrix`.
- `initialPartition` - Initial partition.
- `finalPartition` - Final partition (i.e., partition after randomization and after incomers and outgoers).
- `M` - The desired (specified) image matrix.
- `k` - The number of iterations.
- `combinedPartitions` - Data frame with initial and final partition.
- `whenIncomers` - A vector of which elements tells us at which iterations the incomers were added.
- `whenOutgoers` - A vector of which elements tells us at which iterations the outgoers were removed.
- `ERR` - Sum of squared differences between the desired and empirical densities across blocks; for each iteration.
- `linkERR` - The difference in the number of links between the generated number of links and desired number of links; for each iteration.

Author(s)

Marjan Cugmas and Aleš Žiberna

Examples

```
formula <- list(mutuality, popularity, OTPtransitivity)
X <- matrix(sample(c(0,1), size = 9**2, replace = TRUE), nrow = 9)
diag(X) <- 0
M <- matrix(c(0.1, 0.8, 0.8, 0.1), nrow = 2)
partition <- c(1, 2, 2, 1, 1, 2, 2, 2, 1)
nemSymBMinout(X = X,
              partition = partition,
              formula = formula,
              theta = c(1, 1, 1),
              M = M,
              k = 100,
              minClusterSize = 2,
              nin = 10,
              nout = 5,
              loops = FALSE)
```

normalizeRsphere *Normalize values on a sphere*

Description

Normalizes values of a vector such that the sum of squared elements equal to r^2.

Usage

```
normalizeRsphere(x, r = 1)
```

Arguments

x	A vector or a matrix with values to be normalized.
r	The diameter of a sphere, default 1.

Value

It returns a data frame with normalized values.

Author(s)

Marjan Cugmas

Examples

```
normalizeRsphere(x = c(1, 0.5, 0.4))
```

OSPtransitivity *Outgoing shared partners mechanism*

Description

Calculates the network statistic according to the outgoing shared partners mechanism.

Usage

```
OSPtransitivity(X, actor)
```

Arguments

X	Binary network; of class matrix.
actor	A unit (actor; row/column number), which have an opportunity to change a link.

Value

A vector with the number of paths of length two between the actor and other units.

Author(s)

Marjan Cugmas and Aleš Žiberna

References

Cugmas, M., & Žiberna, A. (2022). Approaches to blockmodeling dynamic networks: a Monte Carlo simulation study. Social Networks, in review.

Examples

```
X <- matrix(sample(c(0,1), size = 9**2, replace = TRUE), nrow = 9)
OSPtransitivity(X, actor = 2)
```

OTPtransitivity *Outgoing two-path mechanism*

Description

Calculates the network statistic according to the outgoing two path mechanism.

Usage

```
OTPtransitivity(X, actor)
```

Arguments

X	Binary network; of class matrix.
actor	A unit (actor; row/column number), which have an opportunity to change a link.

Value

A vector with the number of paths of length two between the actor and other units.

Author(s)

Marjan Cugmas and Aleš Žiberna

References

Cugmas, M., & Žiberna, A. (2022). Approaches to blockmodeling dynamic networks: a Monte Carlo simulation study. Social Networks, in print.

Examples

```
X <- matrix(sample(c(0,1), size = 9**2, replace = TRUE), nrow = 9)
OTPtransitivity(X, actor = 2)
```

popularity *Popularity mechanism*

Description

Calculate the normalized network statistic according to the popularity mechanism.

Usage

```
popularity(X, actor = NULL)
```

Arguments

X	Binary network; of class matrix.
actor	Not used by the function, set to NULL. Necessary for using within other functions, e.g. nemBM.

Value

A vector with the normalized popularity mechanism, cacluated for each unit.

Author(s)

Marjan Cugmas and Aleš Žiberna

References

Cugmas, M., & Žiberna, A. (2022). Approaches to blockmodeling dynamic networks: a Monte Carlo simulation study. Social Networks, in print.

Examples

```
X <- matrix(sample(c(0,1), size = 9**2, replace = TRUE), nrow = 9)
popularity(X)
```

randomizePartition *Randomize a partition*

Description

It randomizes a partition by randomly relocating a given share of units between the clusters. The group sizes are preserved.

Usage

```
randomizePartition(partition, p, checkSelected = FALSE)
```

Arguments

partition	Initial partition in a vector format. Each unique value (positive integers) represents one cluster.
p	The share of relocated units.
checkSelected	If TRUE (default is FALSE) a given unit can be relocated only once.

Value

A partition (in a vector format).

Author(s)

Marjan Cugmas and Aleš Žiberna

Examples

```
randomizePartition(partition = c(1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3), p = 0.3)
```

RL *Relocating Links algorithm (RL algorithm)*

Description

It generates random network considering the selected types of triads.

Usage

```
RL(ideal.net, initial.net, triads = "forb", k = 100, custom.triads = NULL)
```

Arguments

ideal.net	Network with a desired blockmodel without inconsistencies; of class matrix.
initial.net	Initial network; of class matrix.
triads	What types of triads has to be considered (allowed allow, forbidden forb, all all or custom cust). Provide a list of triad types as used in package ergm.
k	Number of iterations.
custom.triads	A list with names of a subset of triads to be considered. The same names must be used as in ERGM package. Only if triads = "cust".

Value

A list contiainig: new.network which is the generated network (of class matrix); and CR which is a vector of CR values (calculated after each iteration).

Author(s)

Marjan Cugmas and Aleš Žiberna

References

Cugmas M, Ferligoj A, Žiberna A (2018) Generating global network structures by triad types. PLoS ONE 13(5): e0197514. https://doi.org/10.1371/journal.pone.0197514

Examples

```
# generate initial and ideal network
cohesiveBM <- rbind(c("com", "nul"), c("nul", "com"))
ideal <- genNetworkLE(BM = cohesiveBM, LE = 0, size = c(4, 4))
random <- genNetworkLE(BM = cohesiveBM, LE = 1, size = c(4, 4))
# generate network with the RL algorithm
generatedNetwork <- RL(ideal.net = ideal, initial.net = random, triads = "all", k = 10)
```

SSEblock *Sum of squared error across blocks*

Description

It calculates the sum of square differences between the desired (specified by an image matrix M) denstities and empirical densities.

Usage

```
SSEblock(X, M, partition, loops)
```

Arguments

X	Initial binary network; of class matrix.
M	Image matrix with block densities.
partition	A partition in a vector format. Each unique value (positive integers) represents one cluster.
loops	Wheter loops are allowed or not.

Value

Sum of squared error (a single value).

Author(s)

Marjan Cugmas

Examples

```
X <- matrix(sample(c(0,1), size = 9**2, replace = TRUE), nrow = 9)
diag(X) <- 0
M <- matrix(c(0.1, 0.4, 0.5, 0.3), nrow = 2)
partition <- c(1, 2, 2, 1, 1, 2, 2, 2, 1)
SSEblock(X = X, M = M, partition = partition, loops = TRUE)
```

`WeightedNetworkStatistics` *Weighted network statistics*

Description

It calculates the weighted network statistics, considering the selected local network mecahnisms and their weights.

Usage

```
WeightedNetworkStatistics(X, formula, theta, actor, randomSD = 0)
```

Arguments

`X`	Binary network; of class `matrix`.
`formula`	The list of local netork mechanisms to be considered.
`theta`	A vector with the mechanisms' weights/strengths.
`actor`	A unit (actor; row/column number), which have an opportunity to change a link.
`randomSD`	The srandard deviation of a normal distribution form which the random part of weighed network statistics is sampled.

Value

The data frame with one column and the number of rows equal to the number of units.

Author(s)

Marjan Cugmas and Aleš Žiberna

Examples

```
formula <- list(mutuality, popularity, OTPtransitivity)
X <- matrix(sample(c(0,1), size = 9**2, replace = TRUE), nrow = 9)
WeightedNetworkStatistics(X = X, formula = formula, theta = c(1, 1, 1), actor = 1)
```

Index