Package ‘noisySBM’

July 22, 2025

Type Package

Title Noisy Stochastic Block Mode: Graph Inference by Multiple Testing

Version 0.1.4

Author Tabea Rebafka [aut, cre],
Etienne Roquain [ctb],
Fanny Villers [aut]

Maintainer Tabea Rebafka <tabea.rebafka@sorbonne-universite.fr>

Description Variational Expectation-
Maximization algorithm to fit the noisy stochastic block model to an observed dense graph and to perform a node clustering. Moreover, a graph inference procedure to recover the underlying binary graph. This procedure comes with a control of the false discovery rate. The method is described
in the article ``Powerful graph inference with false discovery rate control'' by T. Rebafka, E. Roquain, F. Villers (2020) <doi:10.48550/arXiv.1907.10176>.

License GPL-2

Encoding UTF-8

LazyData true

Imports parallel, gtools, ggplot2, RColorBrewer

RoxygenNote 7.1.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

Depends R (>= 2.10)

NeedsCompilation no

Repository CRAN

Date/Publication 2020-12-16 10:40:06 UTC

Contents

addRowToTau — *split group q of provided tau randomly into two into*

Description

split group q of provided tau randomly into two into

Usage

```
addRowToTau(tau, q)
```

Arguments

`tau`	provided tau
`q`	indice of group to split

Value

new tau

ARI — *Evalute the adjusted Rand index*

Description

Compute the adjusted Rand index to compare two partitions

Usage

```
ARI(x, y)
```

Arguments

`x`	vector (of length n) or matrix (with n columns) providing a partition
`y`	vector or matrix providing a partition

Details

the partitions may be provided as n-vectors containing the cluster memeberships of n entities, or by Qxn - matrices whose entries are all 0 and 1 where 1 indicates the cluster membership

Value

the value of the adjusted Rand index

Examples

```
clust1 <- c(1,2,1,2)
clust2 <- c(2,1,2,1)
ARI(clust1, clust2)

clust3 <- matrix(c(1,1,0,0, 0,0,1,1), nrow=2, byrow=TRUE)
clust4 <- matrix(c(1,0,0,0, 0,1,0,0, 0,0,1,1), nrow=3, byrow=TRUE)
ARI(clust3, clust4)
```

`classInd` *convert a clustering into a 0-1-matrix*

Description

convert a clustering into a 0-1-matrix

Usage

```
classInd(cl, nbClusters)
```

Arguments

`cl`	cluster in vector form
`nbClusters`	number of clusters

Value

a 0-1-matrix encoding the clustering

`convertGroupPair` *transform a pair of block identifiers (q,l) into an identifying integer*

Description

this is the inverse function of convertGroupPairIdentifier()

Usage

```
convertGroupPair(q, l, Q, directed)
```

Arguments

`q`	indicator of a latent block
`l`	indicator of a latent block
`Q`	number of latent blocks
`directed`	indicates if the graph is directed

`convertGroupPairIdentifier` *takes a scalar indice of a group pair (q,l) and returns the values q and l*

Description

this is the inverse function of convertGroupPair()

Usage

```
convertGroupPairIdentifier(ind_ql, Q)
```

Arguments

`ind_ql`	indicator for a pair of latent blocks
`Q`	number of latent blocks

`convertNodePair` *transform a pair of nodes (i,j) into an identifying integer*

Description

Associates an identifying integer with a pair of nodes (i,j)

Usage

```
convertNodePair(i, j, n, directed)
```

Arguments

`i`	scalar or vector
`j`	scalar or vector, same length as i
`n`	number of vertices
`directed`	booelan to indicate whether the model is directed or undirected

Details

returns the row number of the matrix build by listNodePairs(n) containing the pair (i,j)

`correctTau` *corrects values of the variational parameters tau that are too close to the 0 or 1*

Description

corrects values of the variational parameters tau that are too close to the 0 or 1

Usage

```
correctTau(tau)
```

Arguments

`tau`	variational parameters

`emv_gamma` *compute the MLE in the Gamma model using the Newton-Raphson method*

Description

compute the MLE in the Gamma model using the Newton-Raphson method

Usage

```
emv_gamma(L, M, param.old, epsilon = 0.001, nb.iter.max = 10)
```

Arguments

`L`	weighted mean of log(data)
`M`	weighted mean of the data
`param.old`	parameters of the Gamma distribution
`epsilon`	threshold for the stopping criterion
`nb.iter.max`	maximum number of iterations

Value

updated parameters of the Gamma distribution `fitNSBM` *VEM algorithm to adjust the noisy stochastic block model to an observed dense adjacency matrix*

Description

`fitNSBM()` estimates model parameters of the noisy stochastic block model and provides a clustering of the nodes

Usage

```
fitNSBM(
  dataMatrix,
  model = "Gauss0",
  sbmSize = list(Qmin = 1, Qmax = NULL, explor = 1.5),
  filename = NULL,
  initParam = list(nbOfTau = NULL, nbOfPointsPerTau = NULL, maxNbOfPasses = NULL,
    minNbOfPasses = 1),
  nbCores = parallel::detectCores()
)
```

Arguments

`dataMatrix` observed dense adjacency matrix

`model` Implemented models:

- `Gauss` all Gaussian parameters of the null and the alternative distributions are unknown ; this is the Gaussian model with maximum number of unknown parameters
- `Gauss0` compared to `Gauss`, the mean of the null distribution is set to 0
- `Gauss01` compared to `Gauss`, the null distribution is set to N(0,1)
- `GaussEqVar` compared to `Gauss`, all Gaussian variances (of both the null and the alternative) are supposed to be equal, but unknown
- `Gauss0EqVar` compared to `GaussEqVar`, the mean of the null distribution is set to 0
- `Gauss0Var1` compared to `Gauss`, all Gaussian variances are set to 1 and the null distribution is set to N(0,1)
- `Gauss2distr` the alternative distribution is a single Gaussian distribution, i.e. the block memberships of the nodes do not influence on the alternative distribution
- `GaussAffil` compared to `Gauss`, for the alternative distribution, there's a distribution for inter-group and another for intra-group interactions
- `Exp` the null and the alternatives are all exponential distributions (i.e. Gamma distributions with shape parameter equal to one) with unknown scale parameters

- `ExpGamma` the null distribution is an unknown exponential, the alterantive distribution are Gamma distributions with unknown parameters

`sbmSize` list of parameters determining the size of SBM (the number of latent blocks) to be expored

- `Qmin` minimum number of latent blocks
- `Qmax` maximum number of latent blocks
- `explor` if `Qmax` is not provided, then `Qmax` is automatically determined as `explor` times the number of blocks where the ICL is maximal

`filename` results are saved in a file with this name (if provided)

`initParam` list of parameters that fix the number of initializations

- `nbOfTau` number of initial points for the node clustering (i. e. for the variational parameters `tau`)
- `nbOfPointsPerTau` number of initial points of the latent binary graph
- `maxNbOfPasses` maximum number of passes through the SBM models, that is, passes from `Qmin` to `Qmax` or inversely
- `minNbOfPasses` minimum number of passes through the SBM models

`nbCores` number of cores used for parallelization

Details

`fitNSBM()` supports different probability distributions for the edges and can estimate the number of node blocks

Value

Returns a list of estimation results for all numbers of latent blocks considered by the algorithm. Every element is a list composed of:

- `theta` estimated parameters of the noisy stochastic block model; a list with the following elements:
  - `pi` parameter estimate of pi
  - `w` parameter estimate of w
  - `nu0` parameter estimate of nu0
  - `nu` parameter estimate of nu
- `clustering` node clustering obtained by the noisy stochastic block model, more precisely, a hard clustering given by the maximum aposterior estimate of the variational parameters `sbmParam$edgeProba`
- `sbmParam` further results concerning the latent binary stochastic block model. A list with the following elements:
  - `Q` number of latent blocks in the noisy stochastic block model
  - `clusterProba` soft clustering given by the conditional probabilities of a node to belong to a given latent block. In other words, these are the variational parameters `tau`; (Q x n)-matrix
  - `edgeProba` conditional probabilities `rho` of an edges given the node memberships of the interacting nodes; (N_Q x N)-matrix
  - `ICL` value of the ICL criterion at the end of the algorithm

`convergence` a list of convergence indicators:

- `J` value of the lower bound of the log-likelihood function at the end of the algorithm
- `complLogLik` value of the complete log-likelihood function at the end of the algorithm
- `converged` indicates if algorithm has converged
- `nbIter` number of iterations performed

Examples

```
n <- 10
theta <- list(pi= c(0.5,0.5), nu0=c(0,.1),
       nu=matrix(c(-2,10,-2, 1,1,1),3,2),  w=c(.5, .9, .3))
obs <- rnsbm(n, theta, modelFamily='Gauss')
res <- fitNSBM(obs$dataMatrix, sbmSize = list(Qmax=3),
       initParam=list(nbOfTau=1, nbOfPointsPerTau=1), nbCores=1)
```

---

getBestQ   *optimal number of SBM blocks*

Description

returns the number of SBM blocks that maximizes the ICL

Usage

```
getBestQ(bestSolutionAtQ)
```

Arguments

`bestSolutionAtQ`
output of `fitNSBM()`, i.e. a list of estimation results for varying number of latent blocks

Value

a list the maximal value of the ICL criterion among the provided solutions along with the best number of latent blocks

Examples

```
# res_gauss is the output of a call of fitNSBM()
getBestQ(res_gauss)
```

getRho — *compute rho associated with given values of w, nu0 and nu*

Description

compute rho associated with given values of w, nu0 and nu

Usage

```
getRho(Q, w, nu0, nu, data, modelFamily)
```

Arguments

`Q`	number of latent blocks in the noisy stochastic block model
`w`	weight parameter in the noisy stochastic block model
`nu0`	null parameter in the noisy stochastic block model
`nu`	alternative parameter in the noisy stochastic block model
`data`	data vector in the undirected model, data matrix in the directed model
`modelFamily`	probability distribution for the edges. Possible values: `Gauss`, `Gamma`

Value

a matrix of conditional probabilities of an edge given the node memberships of the interacting nodes

getTauql — *Evaluate tau_q*tau_l in the noisy stochastic block model*

Description

Evaluate tau_q*tau_l in the noisy stochastic block model

Usage

```
getTauql(q, l, tau, n, directed)
```

Arguments

`q`	indicator of a latent block
`l`	indicator of a latent block
`tau`	variational parameters
`n`	number of vertices
`directed`	booelan to indicate whether the model is directed or undirected

graphInference *new graph inference procedure*

Description

new graph inference procedure

Usage

```
graphInference(
  dataMatrix,
  nodeClustering,
  theta,
  alpha = 0.05,
  modelFamily = "Gauss"
)
```

Arguments

`dataMatrix`	observed adjacency matrix, nxn matrix
`nodeClustering`	n-vector of hard node Clustering
`theta`	parameter of the noisy stochastic block model
`alpha`	confidence level
`modelFamily`	probability distribution for the edges. Possible values: `Gauss` and `Gamma`

Details

graph inference procedure based on conditional q-values in the noisy stochastic block model. It works in the Gaussian model, and also in the Gamma model, but only if the shape parameters of the Gamma distributions under the null and the alternatives are identical (e.g. when all distributions are exponentials).

Value

a list with:

`A` resulting binary adjacency matrix

`qvalues` vector with conditional q-values in the noisy stochastic block model

Examples

```
set.seed(1)
theta <- list(pi=c(.5,.5), w=c(.8,.1,.2), nu0=c(0,1), nu=matrix(c(-1,5,10, 1,1,1), ncol=2))
obs <- rnsbm(n=30, theta)
# res_gauss <- fitNSBM(obs$dataMatrix, nbCores=1)
resGraph <- graphInference(obs$dataMatrix, res_gauss[[2]]$clustering, theta, alpha=0.05)
sum((resGraph$A))/2 # nb of derived edges
sum(obs$latentAdj)/2 # correct nb of edges
```

ICL_Q *computation of the Integrated Classification Likelihood criterion*

Description

computation of the Integrated Classification Likelihood criterion for a result provided by mainVEM_Q()

Usage

```
ICL_Q(solutionThisRun, model)
```

Arguments

`solutionThisRun`
: result provided by mainVEM_Q()

`model`
: Implemented models:

  - `Gauss` all Gaussian parameters of the null and the alternative distributions are unknown ; this is the Gaussian model with maximum number of unknown parameters
  - `Gauss0` compared to `Gauss`, the mean of the null distribution is set to 0
  - `Gauss01` compared to `Gauss`, the null distribution is set to N(0,1)
  - `GaussEqVar` compared to `Gauss`, all Gaussian variances (of both the null and the alternative) are supposed to be equal, but unknown
  - `Gauss0EqVar` compared to `GaussEqVar`, the mean of the null distribution is set to 0
  - `Gauss0Var1` compared to `Gauss`, all Gaussian variances are set to 1 and the null distribution is set to N(0,1)
  - `Gauss2distr` the alternative distribution is a single Gaussian distribution, i.e. the block memberships of the nodes do not influence on the alternative distribution
  - `GaussAffil` compared to `Gauss`, for the alternative distribution, there's a distribution for inter-group and another for intra-group interactions
  - `Exp` the null and the alternatives are all exponential distributions (i.e. Gamma distributions with shape parameter equal to one) with unknown scale parameters
  - `ExpGamma` the null distribution is an unknown exponential, the alterantive distribution are Gamma distributions with unknown parameters

Value

value of the ICL criterion

`initialPoints` *compute a list of initial points for the VEM algorithm*

Description

compute a list of initial points of tau and rhofor the VEM algorithm for a given number of blocks; returns nbOfTau*nbOfPointsPerTau inital points

Usage

```
initialPoints(
  Q,
  dataMatrix,
  nbOfTau,
  nbOfPointsPerTau,
  modelFamily,
  model,
  directed
)
```

Arguments

`Q` number of latent blocks in the noisy stochastic block model

`dataMatrix` observed dense adjacency matrix

`nbOfTau` number of initializations for the latent block memberships

`nbOfPointsPerTau` number of initializations of the latent binary graph associated with each initial latent block memberships

`modelFamily` probability distribution for the edges. Possible values: `Gauss`, `Gamma`

`model` Implemented models:

- `Gauss` all Gaussian parameters of the null and the alternative distributions are unknown ; this is the Gaussian model with maximum number of unknown parameters
- `Gauss0` compared to `Gauss`, the mean of the null distribution is set to 0
- `Gauss01` compared to `Gauss`, the null distribution is set to N(0,1)
- `GaussEqVar` compared to `Gauss`, all Gaussian variances (of both the null and the alternative) are supposed to be equal, but unknown
- `Gauss0EqVar` compared to `GaussEqVar`, the mean of the null distribution is set to 0
- `Gauss0Var1` compared to `Gauss`, all Gaussian variances are set to 1 and the null distribution is set to N(0,1)
- `Gauss2distr` the alternative distribution is a single Gaussian distribution, i.e. the block memberships of the nodes do not influence on the alternative distribution

`GaussAffil` compared to `Gauss`, for the alternative distribution, there's a distribution for inter-group and another for intra-group interactions

`Exp` the null and the alternatives are all exponential distributions (i.e. Gamma distributions with shape parameter equal to one) with unknown scale parameters

`ExpGamma` the null distribution is an unknown exponential, the alterantive distribution are Gamma distributions with unknown parameters

`directed` booelan to indicate whether the model is directed or undirected

Value

list of inital points of tau and rho of length nbOfTau*nbOfPointsPerTau

---

`initialPointsByMerge` *Construct initial values with Q groups by meging groups of a solution obtained with Q+1 groups*

Description

Construct initial values with Q groups by meging groups of a solution obtained with Q+1 groups

Usage

```
initialPointsByMerge(
  tau_Qp1,
  nbOfTau,
  nbOfPointsPerTau,
  data,
  modelFamily,
  model,
  directed
)
```

Arguments

`tau_Qp1` tau for a model with Q+1 latent blocks

`nbOfTau` number of initializations for the latent block memberships

`nbOfPointsPerTau` number of initializations of the latent binary graph associated with each initial latent block memberships

`data` data vector in the undirected model, data matrix in the directed model

`modelFamily` probability distribution for the edges. Possible values: `Gauss`, `Gamma`

`model` Implemented models:

`Gauss` all Gaussian parameters of the null and the alternative distributions are unknown ; this is the Gaussian model with maximum number of unknown parameters

`Gauss0` compared to `Gauss`, the mean of the null distribution is set to 0

`Gauss01` compared to `Gauss`, the null distribution is set to N(0,1)

`GaussEqVar` compared to `Gauss`, all Gaussian variances (of both the null and the alternative) are supposed to be equal, but unknown

`Gauss0EqVar` compared to `GaussEqVar`, the mean of the null distribution is set to 0

`Gauss0Var1` compared to `Gauss`, all Gaussian variances are set to 1 and the null distribution is set to N(0,1)

`Gauss2distr` the alternative distribution is a single Gaussian distribution, i.e. the block memberships of the nodes do not influence on the alternative distribution

`GaussAffil` compared to `Gauss`, for the alternative distribution, there's a distribution for inter-group and another for intra-group interactions

`Exp` the null and the alternatives are all exponential distributions (i.e. Gamma distributions with shape parameter equal to one) with unknown scale parameters

`ExpGamma` the null distribution is an unknown exponential, the alterantive distribution are Gamma distributions with unknown parameters

`directed` booelan to indicate whether the model is directed or undirected

Value

list of inital points of tau and rho of length nbOfTau*nbOfPointsPerTau

---

`initialPointsBySplit` *Construct initial values with Q groups by splitting groups of a solution obtained with Q-1 groups*

Description

Construct initial values with Q groups by splitting groups of a solution obtained with Q-1 groups

Usage

```
initialPointsBySplit(
  tau_Qm1,
  nbOfTau,
  nbOfPointsPerTau,
  data,
  modelFamily,
  model,
  directed
)
```

Arguments

- `tau_Qm1` tau for a model with Q-1 latent blocks
- `nbOfTau` number of initializations for the latent block memberships
- `nbOfPointsPerTau` number of initializations of the latent binary graph associated with each initial latent block memberships
- `data` data vector in the undirected model, data matrix in the directed model
- `modelFamily` probability distribution for the edges. Possible values: `Gauss`, `Gamma`
- `model` Implemented models:
  - `Gauss` all Gaussian parameters of the null and the alternative distributions are unknown ; this is the Gaussian model with maximum number of unknown parameters
  - `Gauss0` compared to `Gauss`, the mean of the null distribution is set to 0
  - `Gauss01` compared to `Gauss`, the null distribution is set to N(0,1)
  - `GaussEqVar` compared to `Gauss`, all Gaussian variances (of both the null and the alternative) are supposed to be equal, but unknown
  - `Gauss0EqVar` compared to `GaussEqVar`, the mean of the null distribution is set to 0
  - `Gauss0Var1` compared to `Gauss`, all Gaussian variances are set to 1 and the null distribution is set to N(0,1)
  - `Gauss2distr` the alternative distribution is a single Gaussian distribution, i.e. the block memberships of the nodes do not influence on the alternative distribution
  - `GaussAffil` compared to `Gauss`, for the alternative distribution, there's a distribution for inter-group and another for intra-group interactions
  - `Exp` the null and the alternatives are all exponential distributions (i.e. Gamma distributions with shape parameter equal to one) with unknown scale parameters
  - `ExpGamma` the null distribution is an unknown exponential, the alterantive distribution are Gamma distributions with unknown parameters
- `directed` booelan to indicate whether the model is directed or undirected

Value

list of inital points of tau and rho of length nbOfTau*nbOfPointsPerTau

---

`initialRho` *compute initial values of rho*

Description

for every provided initial point of tau nbOfPointsPerTau initial values of rho are computed in the Gamma model also initial values of nu are computed

Usage

```
initialRho(listOfTau, nbOfPointsPerTau, data, modelFamily, model, directed)
```

Arguments

- `listOfTau` output of initialTau()
- `nbOfPointsPerTau` number of initializations of the latent binary graph associated with each initial latent block memberships
- `data` data vector in the undirected model, data matrix in the directed model
- `modelFamily` probability distribution for the edges. Possible values: `Gauss`, `Gamma`
- `model` Implemented models:
  - `Gauss` all Gaussian parameters of the null and the alternative distributions are unknown ; this is the Gaussian model with maximum number of unknown parameters
  - `Gauss0` compared to `Gauss`, the mean of the null distribution is set to 0
  - `Gauss01` compared to `Gauss`, the null distribution is set to N(0,1)
  - `GaussEqVar` compared to `Gauss`, all Gaussian variances (of both the null and the alternative) are supposed to be equal, but unknown
  - `Gauss0EqVar` compared to `GaussEqVar`, the mean of the null distribution is set to 0
  - `Gauss0Var1` compared to `Gauss`, all Gaussian variances are set to 1 and the null distribution is set to N(0,1)
  - `Gauss2distr` the alternative distribution is a single Gaussian distribution, i.e. the block memberships of the nodes do not influence on the alternative distribution
  - `GaussAffil` compared to `Gauss`, for the alternative distribution, there's a distribution for inter-group and another for intra-group interactions
  - `Exp` the null and the alternatives are all exponential distributions (i.e. Gamma distributions with shape parameter equal to one) with unknown scale parameters
  - `ExpGamma` the null distribution is an unknown exponential, the alterantive distribution are Gamma distributions with unknown parameters
- `directed` booelan to indicate whether the model is directed or undirected

Value

list of inital points of tau and rho

`initialTau` *compute intial values for tau*

Description

returns a list of length nbOfTau of initial points for tau using spectral clustering with absolute values, kmeans and random perturbations of these points

Usage

```
initialTau(Q, dataMatrix, nbOfTau, percentageOfPerturbation, directed)
```

Arguments

`Q`	number of latent blocks in the noisy stochastic block model
`dataMatrix`	observed dense adjacency matrix
`nbOfTau`	number of initializations for the latent block memberships
`percentageOfPerturbation`	percentage of node labels that are perturbed to obtain further initial points
`directed`	booelan to indicate whether the model is directed or undirected

Value

a list of length nbOfTau of initial points for tau

`J.gamma` *evaluate the objective in the Gamma model*

Description

evaluate the objective in the Gamma model

Usage

```
J.gamma(param, L, M)
```

Arguments

`param`	parameters of the Gamma distribution
`L`	weighted mean of log(data)
`M`	weighted mean of the data

Value

value of the lower bound of the log-likelihood function

`JEvalMstep` *evaluation of the objective in the Gauss model*

Description

evaluation of the objective in the Gauss model

Usage

```
JEvalMstep(VE, mstep, data, modelFamily, directed)
```

Arguments

`VE`	list with variational parameters tau and rho
`mstep`	list with current model parameters and additional auxiliary terms
`data`	data vector in the undirected model, data matrix in the directed model
`modelFamily`	probability distribution for the edges. Possible values: `Gauss`, `Gamma`
`directed`	booelan to indicate whether the model is directed or undirected

Value

value of the ELBO and the complete log likelihood function

`listNodePairs` *returns a list of all possible node pairs (i,j)*

Description

returns a list of all possible node pairs (i,j)

Usage

```
listNodePairs(n, directed = FALSE)
```

Arguments

`n`	number of nodes
`directed`	indicates if the graph is directed

Value

a 2-column matrix with all possible node pairs (i,j)

lvaluesNSBM *compute conditional l-values in the noisy stochastic block model*

Description

compute conditional l-values in the noisy stochastic block model

Usage

```
lvaluesNSBM(dataVec, Z, theta, directed = FALSE, modelFamily = "Gauss")
```

Arguments

`dataVec`	data vector
`Z`	a node clustering
`theta`	list of parameters for a noisy stochastic block model
`directed`	indicates if the graph is directed
`modelFamily`	probability distribution for the edges. Possible values: `Gauss` and `Gamma`

Value

conditional l-values in the noisy stochastic block model

mainVEM_Q *main function of VEM algorithm with fixed number of SBM blocks*

Description

main function of VEM algorithm with fixed number of SBM blocks

Usage

```
mainVEM_Q(init, modelFamily, model, data, directed)
```

Arguments

`init`	list of initial points for the algorithm
`modelFamily`	probability distribution for the edges. Possible values: `Gauss`, `Gamma`
`model`	Implemented models: `Gauss` all Gaussian parameters of the null and the alternative distributions are unknown ; this is the Gaussian model with maximum number of unknown parameters `Gauss0` compared to `Gauss`, the mean of the null distribution is set to 0

Gauss01 compared to Gauss, the null distribution is set to N(0,1)

GaussEqVar compared to Gauss, all Gaussian variances (of both the null and the alternative) are supposed to be equal, but unknown

Gauss0EqVar compared to GaussEqVar, the mean of the null distribution is set to 0

Gauss0Var1 compared to Gauss, all Gaussian variances are set to 1 and the null distribution is set to N(0,1)

Gauss2distr the alternative distribution is a single Gaussian distribution, i.e. the block memberships of the nodes do not influence on the alternative distribution

GaussAffil compared to Gauss, for the alternative distribution, there's a distribution for inter-group and another for intra-group interactions

Exp the null and the alternatives are all exponential distributions (i.e. Gamma distributions with shape parameter equal to one) with unknown scale parameters

ExpGamma the null distribution is an unknown exponential, the alterantive distribution are Gamma distributions with unknown parameters

data data vector in the undirected model, data matrix in the directed model

directed booelan to indicate whether the model is directed or undirected

Value

list of estimated model parameters and a node clustering; like the output of fitNSBM()

---

mainVEM_Q_par *main function of VEM algorithm for fixed number of latent blocks in parallel computing*

Description

runs the VEM algorithm the provided initial point

Usage

```
mainVEM_Q_par(s, ListOfTauRho, modelFamily, model, data, directed)
```

Arguments

s indice of initial point in ListOfTauRho to be used for this run

ListOfTauRho a list of initial points

modelFamily probability distribution for the edges. Possible values: Gauss, Gamma

model Implemented models:

Gauss all Gaussian parameters of the null and the alternative distributions are unknown ; this is the Gaussian model with maximum number of unknown parameters

- `Gauss0` compared to `Gauss`, the mean of the null distribution is set to 0
- `Gauss01` compared to `Gauss`, the null distribution is set to N(0,1)
- `GaussEqVar` compared to `Gauss`, all Gaussian variances (of both the null and the alternative) are supposed to be equal, but unknown
- `Gauss0EqVar` compared to `GaussEqVar`, the mean of the null distribution is set to 0
- `Gauss0Var1` compared to `Gauss`, all Gaussian variances are set to 1 and the null distribution is set to N(0,1)
- `Gauss2distr` the alternative distribution is a single Gaussian distribution, i.e. the block memberships of the nodes do not influence on the alternative distribution
- `GaussAffil` compared to `Gauss`, for the alternative distribution, there's a distribution for inter-group and another for intra-group interactions
- `Exp` the null and the alternatives are all exponential distributions (i.e. Gamma distributions with shape parameter equal to one) with unknown scale parameters
- `ExpGamma` the null distribution is an unknown exponential, the alterantive distribution are Gamma distributions with unknown parameters

`data`	data vector in the undirected model, data matrix in the directed model
`directed`	booelan to indicate whether the model is directed or undirected

Value

list of estimated model parameters and a node clustering; like the output of fitNSBM()

`modelDensity` *evaluate the density in the current model*

Description

evaluate the density in the current model

Usage

```
modelDensity(x, nu, modelFamily = "Gauss")
```

Arguments

`x`	vector with points where to evaluate the density
`nu`	distribution parameter
`modelFamily`	probability distribution for the edges. Possible values: `Gauss`, `Gamma`, `Poisson`

Mstep *M-step*

Description

performs one M-step, that is, update of pi, w, nu, nu0

Usage

```
Mstep(VE, mstep, model, data, modelFamily, directed)
```

Arguments

`VE`	list with variational parameters tau and rho
`mstep`	list with current model parameters and additional auxiliary terms
`model`	Implemented models: `Gauss` all Gaussian parameters of the null and the alternative distributions are unknown ; this is the Gaussian model with maximum number of unknown parameters `Gauss0` compared to `Gauss`, the mean of the null distribution is set to 0 `Gauss01` compared to `Gauss`, the null distribution is set to N(0,1) `GaussEqVar` compared to `Gauss`, all Gaussian variances (of both the null and the alternative) are supposed to be equal, but unknown `Gauss0EqVar` compared to `GaussEqVar`, the mean of the null distribution is set to 0 `Gauss0Var1` compared to `Gauss`, all Gaussian variances are set to 1 and the null distribution is set to N(0,1) `Gauss2distr` the alternative distribution is a single Gaussian distribution, i.e. the block memberships of the nodes do not influence on the alternative distribution `GaussAffil` compared to `Gauss`, for the alternative distribution, there's a distribution for inter-group and another for intra-group interactions `Exp` the null and the alternatives are all exponential distributions (i.e. Gamma distributions with shape parameter equal to one) with unknown scale parameters `ExpGamma` the null distribution is an unknown exponential, the alterantive distribution are Gamma distributions with unknown parameters
`data`	data vector in the undirected model, data matrix in the directed model
`modelFamily`	probability distribution for the edges. Possible values: `Gauss`, `Gamma`
`directed`	booelan to indicate whether the model is directed or undirected

Value

updated list `mstep` with current model parameters and additional auxiliary terms

plotGraphs — *plot the data matrix, the inferred graph and/or the true binary graph*

Description

plot the data matrix, the inferred graph and/or the true binary graph

Usage

```
plotGraphs(dataMatrix = NULL, inferredGraph = NULL, binaryTruth = NULL)
```

Arguments

`dataMatrix`	observed data matrix
`inferredGraph`	graph inferred by the multiple testing procedure via graphInference()
`binaryTruth`	true binary graph

Value

a list of FDR and TDR values, if possible

plotICL — *plot ICL curve*

Description

plot ICL curve

Usage

```
plotICL(res)
```

Arguments

`res`	output of fitNSBM()

Value

figure of ICL curve

Examples

```
# res_gauss is the output of a call of fitNSBM()
plotICL(res_gauss)
```

qvaluesNSBM *compute q-values in the noisy stochastic block model*

Description

compute q-values in the noisy stochastic block model

Usage

```
qvaluesNSBM(
  dataVec,
  Z,
  theta,
  lvalues,
  modelFamily = "Gauss",
  directed = FALSE
)
```

Arguments

dataVec	data vector
Z	a node clustering
theta	list of parameters for a noisy stochastic block model
lvalues	conditional l-values in the noisy stochastic block model
modelFamily	probability distribution for the edges. Possible values: Gauss and Gamma
directed	indicates if the graph is directed

Value

q-values in the noisy stochastic block model

q_delta_ql *auxiliary function for the computation of q-values*

Description

auxiliary function for the computation of q-values

Usage

```
q_delta_ql(theta, ind, t, modelFamily = "Gauss")
```

Arguments

`theta`	list of parameters for a noisy stochastic block model
`ind`	indicator for a pair of latent blocks
`t`	l-values
`modelFamily`	probability distribution for the edges. Possible values: `Gauss` and `Gamma`

`res_exp` *Output of fitNSBM() on a dataset applied in the exponential NSBM*

Description

Parameter estimates fitted on a dataset given in the vignette

Usage

```
res_exp
```

Format

List with estimation results for different number of SBM blocks. Output of fitNSBM()

`res_gamma` *Output of fitNSBM() on a dataset applied in the Gamma NSBM*

Description

Parameter estimates fitted on a dataset given in the vignette

Usage

```
res_gamma
```

Format

List with estimation results for different number of SBM blocks. Output of fitNSBM()

`res_gauss` *Output of fitNSBM() on a dataset applied in the Gaussian NSBM*

Description

Parameter estimates fitted on a dataset given in the vignette

Usage

```
res_gauss
```

Format

List with estimation results for different number of SBM blocks. Output of fitNSBM()

`rnsbm` *simulation of a graph according the noisy stochastic block model*

Description

simulation of a graph according the noisy stochastic block model

Usage

```
rnsbm(n, theta, modelFamily = "Gauss", directed = FALSE)
```

Arguments

`n`	number of nodes
`theta`	model parameters of the noisy stochastic block model **pi** latent block proportions, Q-vector **w** connectivity parameters, N_Q-vector **nu0** parameters of the null distribution **nu** parameters of the alternative distribution
`modelFamily`	probability distribution for the edges. Possible values: `Gauss`, `Gamma`, `Poisson`
`directed`	indicates if the graph is directed (boolean)

Value

a list with:

dataMatrix simulated matrix from the noisy stochastic block model

theta model parameters of the noisy stochastic block model

latentZ underlying latent node memberships

latentAdj underlying latent binary graph

Examples

```
n <- 10
Q <- 2
theta <- list(pi= rep(1/Q,Q), nu0=c(0,1))
theta$nu <- matrix(c(-2,10,-2, 1,1,1),nrow=Q*(Q+1)/2,ncol=2)
theta$w <- c(.5, .9, .3)
obs <- rnsbm(n, theta, modelFamily='Gauss')
obs
```

`spectralClustering` *spectral clustering with absolute values*

Description

performs absolute spectral clustering of an adjacency matrix

Usage

```
spectralClustering(A, K)
```

Arguments

`A`	adjacency matrix
`K`	number of desired clusters

Value

a vector containing a node clustering into K groups

`tauDown` *Create new initial values by merging pairs of groups of provided tau*

Description

Create nbOfMerges new initial values by merging nbOfMerges (or all possible) pairs of groups of provided tau

Usage

```
tauDown(tau, nbOfMerges)
```

Arguments

`tau`	soft node clustering
`nbOfMerges`	number of required merges of blocks

Value

a list of length nbOfMerges (at most) of initial points for tau

---

`tauUp` *Create new values of tau by splitting groups of provided tau*

Description

Create nbOfSplits (or all) new values of tau by splitting nbOfSplits (or all) groups of provided tau

Usage

```
tauUp(tau, nbOfSplits = 1)
```

Arguments

`tau`	soft node clustering
`nbOfSplits`	number of required splits of blocks

Value

a list of length nbOfSplits (at most) of initial points for tau

---

`tauUpdate` *Compute one iteration to solve the fixed point equation in the VE-step*

Description

Compute one iteration to solve the fixed point equation in the VE-step

Usage

```
tauUpdate(tau, log.w, log.1mw, data, VE, mstep, modelFamily, directed)
```

Arguments

`tau`	current value of tau
`log.w`	value of log(w)
`log.1mw`	value of log(1-w)
`data`	data vector in the undirected model, data matrix in the directed model
`VE`	list with variational parameters tau and rho
`mstep`	list with current model parameters and additional auxiliary terms
`modelFamily`	probability distribution for the edges. Possible values: `Gauss`, `Gamma`
`directed`	booelan to indicate whether the model is directed or undirected

Value

updated value of tau

`update_newton_gamma` *Perform one iteration of the Newton-Raphson to compute the MLE of the parameters of the Gamma distribution*

Description

Perform one iteration of the Newton-Raphson to compute the MLE of the parameters of the Gamma distribution

Usage

```
update_newton_gamma(param, L, M)
```

Arguments

`param`	current parameters of the Gamma distribution
`L`	weighted mean of log(data)
`M`	weighted mean of the data

Value

updated parameters of the Gamma distribution

`VEstep` *VE-step*

Description

performs one VE-step, that is, update of tau and rho

Usage

```
VEstep(VE, mstep, data, modelFamily, directed, fix.iter = 5)
```

Arguments

`VE`	list with variational parameters tau and rho
`mstep`	list with current model parameters and additional auxiliary terms
`data`	data vector in the undirected model, data matrix in the directed model
`modelFamily`	probability distribution for the edges. Possible values: `Gauss`, `Gamma`
`directed`	booelan to indicate whether the model is directed or undirected
`fix.iter`	maximal number of iterations for fixed point equation

Value

updated list `VE` with variational parameters tau and rho

Index