Package 'graphclust'

May 8, 2026

Type Package

Title Hierarchical Graph Clustering for a Collection of Networks

Version 1.3

Author Tabea Rebafka [aut, cre]

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

Description Graph clustering using an agglomerative algorithm to maximize the integrated classification likelihood criterion and a mixture of stochastic block models. The method is described in the article ``Model-based clustering of multiple networks with a hierarchical algorithm" by T. Rebafka (2022) <doi:10.48550/arXiv.2211.02314>.

License GPL-2

Encoding UTF-8

Imports blockmodels, igraph, parallel, sClust

RoxygenNote 7.2.3

NeedsCompilation no

Repository CRAN

Date/Publication 2023-06-07 16:50:02 UTC

Contents

`ARI` *Adjusted Rand index*

Description

ARI to compare two clusterings or to compare two entire lists of clusterings

Usage

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

Arguments

`x`	vector with clustering, matrix with hot-one-encoding of the clustering, or a list of clusterings (in vector or matrix form)
`y`	as x

Value

ARI (scalar of vector)

Examples

```
x <- c(1,1,2,2,3,3)
y <- c(1,1,1,2,2,2)
ARI(x,y)

x <- matrix(0, 3, 6)
x[1,1] <- x[1,2] <- x[2,3] <- x[2,4] <- x[3,5] <- x[3,6] <- 1
y <- matrix(0, 2, 6)
y[1,1] <- y[1,2] <- y[1,3] <- y[2,4] <- y[2,5] <- y[2,6] <- 1
ARI(x,y)

X <- list(c(1,1,2,2,3,3), rep(1,10))
Y <- list(c(1,1,1,2,2,2), rep(1:2,each=5))
ARI(X,Y)
```

degreeSort — *Sort stochastic block model parameter in a unique way using its graphon*

Description

Sort stochastic block model parameter in a unique way using its graphon

Usage

```
degreeSort(thetaInit, outTheta = TRUE, outPerm = FALSE)
```

Arguments

thetaInit	stochastic block model parameter to be sorted
outTheta	if TRUE returns the sorted stochastic block model parameter
outPerm	if TRUE returns the permutation of the blocks of the stochastic block model to provide the sorted stochastic block model parameter

Value

according to the values of outTheta and outPerm the function returns the sorted stochastic block model parameter or the associated permutation of the blocks of the stochastic block model or a list with both of them

Examples

```
theta1 <- list(pi=c(.5,.5), gamma=matrix((1:4)/8,2,2))
degreeSort(theta1)
theta2 <- list(pi=c(.5,.5), gamma=matrix(4:1/8,2,2))
degreeSort(theta2)
```

fitSBMcollection — *Fit a unique stochastic block model to a collection of networks*

Description

fitSBMcollection() is a subversion of graphClustering() where no stopping criterion is applied. So all networks are ultimately merged to a single cluster and considered as i.i.d realisations of a single stochastic block model.

Usage

```
fitSBMcollection(
  allAdj,
  hyperParam = list(alpha = 0.5, eta = 0.5, zeta = 0.5, lambda = 0.5),
  nbCores = 1
)
```

Arguments

allAdj	list of adjacency matrices
hyperParam	hyperparameters of prior distributions
nbCores	number of cores for parallelization

Value

list with the following fields: $nodeClusterings is a list with the node labels for each networks, $theta contains the estimated SBM parameter, $ICL is the value of the ICL criterion of the final clustering

Examples

```
theta <- list(pi=c(.5,.5), gamma=matrix((1:4)/8,2,2))
obs <- rCollectSBM(rep(10,4), theta)$listGraphs
res <- fitSBMcollection(obs, nbCores=1)
```

---

fitSimpleSBM *Fit a stochastic block model to every network in a collection of networks.*

Description

Applies the variational EM-algorithm implemented in the package blockmodels to every network.

Usage

```
fitSimpleSBM(
  allAdj,
  directed = TRUE,
  nbSBMBlocks = Inf,
  nbCores = 1,
  outCountStat = TRUE
)
```

Arguments

allAdj	list of adjacency matrices
directed	Networks are directed (TRUE by default) or undirected (FALSE).
nbSBMBlocks	upper bound for the number of blocks in the SBMs of the mixture components. Default is Inf
nbCores	number of cores for parallelization.
outCountStat	If TRUE (default), the output is a list of count statistics for every network. If FALSE, the output is a list of parameters of the stochastic block models fitted to every network.

Value

list of count statistics for every network or list of parameters of the stochastic block models fitted to every network.

Examples

```
theta <- list(pi=c(.5,.5), gamma=matrix((1:4)/8,2,2))
obs <- rCollectSBM(rep(10,4), theta)$listGraphs
res <- fitSimpleSBM(obs, outCountStat=FALSE, nbCores=2)
```

---

graphClustering *Hierarchical graph clustering algorithm*

Description

Applies the hierarchical graph clustering algorithm to a collection of networks and fits a finite mixture model of stochastic block models to the data

Usage

```
graphClustering(
  allAdj,
  hyperParam = list(alpha = 0.5, eta = 0.5, zeta = 0.5, lambda = 0.5),
  returnInitial = FALSE,
  nbClust = NULL,
  nbSBMBlocks = Inf,
  initCountStat = NULL,
  initDeltaICL = NULL,
  nbCores = 1
)
```

Arguments

allAdj	list of adjacency matrices
hyperParam	hyperparameters of prior distributions
returnInitial	Boolean. Return SBM parameters from initialization or not. Default is FALSE.
nbClust	desired number of clusters. Default NULL, which means that the number of clusters is chosen automatically via the ICL criterion
nbSBMBlocks	upper bound for the number of blocks in the SBMs of the mixture components. Default is Inf
initCountStat	initial count statistics may be provided to the method. Default is NULL.
initDeltaICL	initial deltaICL-matrix may be provided to the method. Default is NULL.
nbCores	number of cores for parallelization

Value

list with the following fields: $graphGroups is the graph clustering, $nodeClusterings is a list with the node labels for each networks, $thetaMixSBM contains the estimated parameter of the mixture of SBMs, $ICL is the value of the ICL criterion of the final clustering, $histGraphGroups traces the history of the cluster aggregations, $histDeltaICL traces the evolution of the deltaICL value, $histFusedClusters traces the history of the aggregated cluster numbers

Examples

```
theta <- list(pi=c(.5,.5), gamma=matrix((1:4)/8,2,2))
obs <- rCollectSBM(rep(10,4), theta)$listGraphs
res <- graphClustering(obs, nbCores=1)
```

---

graphMomentsClustering

Graph clustering method using graph moments

Description

Graph clustering method based on graph moments by Mukherjee et al. (2017)

Usage

```
graphMomentsClustering(Networks, nbMoments = 3, nbClusters)
```

Arguments

Networks	list of adjacency matrices
nbMoments	order of the largest graph moments to be considered
nbClusters	desired number of clusters

Value

vector with the clustering of the networks

Examples

```
param <- vector('list', 3)
param[[1]] <- list(prop = 1/3, # component 1 :  alpha > beta
                   alpha = .04,
                   beta = .02,
                   deltaIn = 100,
                   deltaOut = 100,
                   R = 500
)
param[[2]] <- list(prop = 1/3, # component 2 : just permute alpha and beta ;
                   alpha = .01,
                   beta = .02,
                   deltaIn = 100,
                   deltaOut = .1,
                   R = 1000
)
param[[3]] <- list(prop = 1/3, # component 3 : alpha=beta
                   alpha = .015,
                   beta = .015,
                   deltaIn = .1,
                   deltaOut = .1,
                   R = 1000
)
obs <- sampleDPAMixture(M=20, param)
res <- graphMomentsClustering(obs$listAdj, 3, 3)
table(res, obs$graphGroups)
```

---

graphonL2norm *(squared) L2-norm of the graphons associated with two stochastic block model parameters*

Description

(squared) L2-norm of the graphons associated with two stochastic block model parameters

Usage

```
graphonL2norm(theta1, theta2)
```

Arguments

theta1	a stochastic block model parameter
theta2	a stochastic block model parameter

Value

(squared) L2-norm of the graphons associated with two stochastic block model parameters

Examples

```
theta1 <- list(pi=c(.5,.5), gamma=matrix((1:4)/8,2,2))
theta2 <- list(pi=c(.5,.5), gamma=matrix(4:1/8,2,2))
graphonL2norm(theta1, theta2)
```

---

graphonSpectralClustering

Graph clustering using the pairwise graphon distances and spectral clustering

---

Description

Graph clustering using the pairwise graphon distances and spectral clustering

Usage

```
graphonSpectralClustering(allAdj, nbClusters, sig = 0.1, nbCores = 1)
```

Arguments

allAdj	list of adjacency matrices
nbClusters	number of clusters to be found
sig	parameter for Gaussian kernel used for the similarity matrix
nbCores	number of cores for parallelization.

Value

list with the obtained graph clusteirng ($clust) and the matrix with the pairwise graphon distances between all pairs of networks

Examples

```
theta <- list(pi=c(.5,.5), gamma=matrix((1:4)/8,2,2))
obs <- rCollectSBM(rep(10,4), theta)$listGraphs
res <- graphonSpectralClustering(obs, 2, nbCores=1)
```

metagraph — *Plot the metagraph of the parameter of the stochastic block model associated with one of the estimated graph clusters*

Description

Plot the metagraph of the parameter of the stochastic block model associated with one of the estimated graph clusters

Usage

```
metagraph(nb, res, title = NULL, edge.width.cst = 10)
```

Arguments

nb	number of the cluster we are interested in
res	output of graphClustering()
title	title of the figure
edge.width.cst	width of edges in the metagraph

Value

none

Examples

```
theta <- list(pi=c(.5,.5), gamma=matrix((1:4)/8,2,2))
obs <- rCollectSBM(rep(10,4), theta)$listGraphs
res <- graphClustering(obs, nbCores=2)
metagraph(1, res)
```

moments — *Computation of graph moments of a network*

Description

Computation of graph moments of a network

Usage

```
moments(A, k = 3)
```

Arguments

A	adjacency matrix
k	order of the largest graph moments to be considered

Value

vector with the first k (normalized) graph moments of the network A

Examples

```
param <- list(R = 500, alpha = .04, beta = .02, deltaIn = 100, deltaOut = 100)
A <- sampleDPA(param)
moments(A)
```

`permutParam` *Permute block labels of a stochastic block model parameter*

Description

Permute block labels of a stochastic block model parameter

Usage

```
permutParam(theta, permut)
```

Arguments

`theta`	a SBM parameter with say K blocks
`permut`	a permutation of the block labels 1,2,...,K

Value

stochastic block model parameter with permuted block labels

Examples

```
theta1 <- list(pi=c(.5,.5), gamma=matrix((1:4)/8,2,2))
theta2 <- list(pi=c(.5,.5), gamma=matrix(4:1/8,2,2))
permutParam(theta1, 2:1)
permutParam(theta2, 2:1)
```

plotDendrogram — *Plot dendrogram to visualize the clustering obtained by the hierarchical clustering algorithm*

Description

Plot dendrogram to visualize the clustering obtained by the hierarchical clustering algorithm

Usage

```
plotDendrogram(res, labels = NULL, labcex = 0.5)
```

Arguments

res	output of graphClustering()
labels	network labels, default (NULL) network number.
labcex	size of labels in the figure

Value

dendrogram

Examples

```
theta <- list(pi=c(.5,.5), gamma=matrix((1:4)/8,2,2))
obs <- rCollectSBM(rep(10,4), theta)$listGraphs
res <- graphClustering(obs, nbCores=2)
plotDendrogram(res)
```

rCollectSBM — *Simulate a sample of networks of a stochastic block model*

Description

Simulate a sample of networks of a stochastic block model

Usage

```
rCollectSBM(vec_n, theta, directed = TRUE)
```

Arguments

vec_n	vector with number of vertices
theta	stochastic block model parameter with latent group probabilities $pi and connectivy parameters $gamma
directed	directed networks (TRUE by default) or undirected (FALSE)

Value

list with a list of adjacency matrices ($listGraphs) and a list of node labels ($listLatentZ)

Examples

```
theta1 <- list(pi=c(.5,.5), gamma=matrix((1:4)/8,2,2))
rCollectSBM(2:4, theta1)
```

`rMixSBM` *Simulate a collection of networks of a mixture of stochastic block models*

Description

Simulate a collection of networks of a mixture of stochastic block models

Usage

```
rMixSBM(vec_n, thetaMixSBM, directed = TRUE)
```

Arguments

`vec_n`	vector with number of vertices
`thetaMixSBM`	K-list for a mixture with K components. Each field is a list with the stochastic block model parameter ($pi and $gamma) and a cluster proportion ($prop)
`directed`	directed networks (TRUE by default) or undirected (FALSE)

Value

list with a list of adjacency matrices ($listGraphs), a list of node labels ($listLatentZ) and a vector with the graph clustering ($label)

Examples

```
theta1 <- list(prop=.2, pi=c(.5,.5), gamma=matrix((1:4)/8,2,2))
theta2 <- list(prop=.8, pi=c(.5,.5), gamma=matrix(4:1/8,2,2))
thetaMixSBM <- list(NULL)
thetaMixSBM[[1]] <- theta1
thetaMixSBM[[2]] <- theta2
obs <- rMixSBM(vec_n=rep(10,3), thetaMixSBM)
```

rsbm — *Simulate a network of a stochastic block model*

Description

Simulate a network of a stochastic block model

Usage

```
rsbm(n, theta, directed = TRUE)
```

Arguments

n	number of vertices
theta	stochastic block model parameter with latent group probabilities $pi and connectivy parameters $gamma
directed	directed network (TRUE by default) or undirected (FALSE)

Value

list with simulated adjacency matrix ($adj) and node labels ($Z)

Examples

```
theta1 <- list(pi=c(.5,.5), gamma=matrix((1:4)/8,2,2))
rsbm(10, theta1)
```

sampleDPA — *generation of a network of the directed preferential attachment (DPA) model*

Description

generation of a network of the directed preferential attachment (DPA) model

Usage

```
sampleDPA(param)
```

Arguments

param	list with the following elements: $R (= number of iterations), $alpha, $beta , $deltaIn, $deltaOut (parameters of the DPA model)

Value

adjacency matrix of generated network

Examples

```
param <- list(R = 500, alpha = .04, beta = .02, deltaIn = 100, deltaOut = 100)
A <- sampleDPA(param)
A
```

`sampleDPAMixture` *Generation of a mixture of directed preferential attachment (DPA) models*

Description

Generation of a mixture of directed preferential attachment (DPA) models

Usage

```
sampleDPAMixture(M, param)
```

Arguments

`M`	number of desired networks
`param`	list of list of parameters of the DPA models. Each element of param is a list with the following elements: $prop (weight of the mixture component), $R (= number of iterations), $alpha, $beta , $deltaIn, $deltaOut (parameters of the DPA model)

Value

list of 2 lists : the first ($listAdj) is a list of M adjacency matrices, the second a list ($graphGroups) contains the true cluster labels

Examples

```
param <- vector('list', 3)
param[[1]] <- list(prop = 1/3, # component 1 :  alpha > beta
                   alpha = .04,
                   beta = .02,
                   deltaIn = 100,
                   deltaOut = 100,
                   R = 500
)
param[[2]] <- list(prop = 1/3, # component 2 : just permute alpha and beta ;
                   alpha = .01,
                   beta = .02,
                   deltaIn = 100,
                   deltaOut = .1,
                   R = 1000
)
param[[3]] <- list(prop = 1/3, # component 3 : alpha=beta
                   alpha = .015,
```

```
                    beta = .015,
                    deltaIn = .1,
                    deltaOut = .1,
                    R = 1000
)
obs <- sampleDPAMixture(M=20, param)
```

---

sbmNorm *(squared) norm between two stochastic block models*

Description

the norm is the minimal graphon distance between two stochastic block model parameters obtained with the best permutations of the parameters

Usage

```
sbmNorm(theta1, theta2)
```

Arguments

theta1	a stochastic block model parameter
theta2	a stochastic block model parameter

Value

(squared) norm between two stochastic block models

Examples

```
theta1 <- list(pi=c(.5,.5), gamma=matrix((1:4)/8,2,2))
theta2 <- list(pi=c(.5,.5), gamma=matrix(4:1/8,2,2))
theta3 <- list(pi=c(.5,.5), gamma=matrix(1:4/4,2,2))
sbmNorm(theta1, theta2)
sbmNorm(theta1, theta3)
sbmNorm(theta2, theta3)
```

Index