Package 'noisysbmGGM'

March 7, 2024

Title Noisy Stochastic Block Model for GGM Inference

Version 0.1.2.3

Author Valentin Kilian [aut, cre],
Fanny Villers [aut]

Maintainer Valentin Kilian <valentin.kilian@ens-rennes.fr>

Description Greedy Bayesian algorithm to fit the noisy stochastic block model to an observed sparse graph. Moreover, a graph inference procedure to recover Gaussian Graphical Model (GGM) from real data. This procedure comes with a control of the false discovery rate. The method is described in the article ``Enhancing the Power of Gaussian Graphical Model Inference by Modeling the Graph Structure'' by Kilian, Rebafka, and Villers (2024) <arXiv:2402.19021>.

License GPL-2

Encoding UTF-8

Imports parallel,ppcor,SILGGM,stats,igraph,huge,Rcpp,RcppArmadillo,MASS,RColorBrewer

RoxygenNote 7.2.3

Depends R (>= 3.1.0)

LazyData true

LinkingTo Rcpp, RcppArmadillo

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation yes

Repository CRAN

Date/Publication 2024-03-07 10:40:02 UTC

R topics documented:

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)
```

GGMtest *GGM for test*

Description

Example of a GGM

Usage

```
GGMtest
```

Format

`dataMatrix` A n-sample of a p Gaussian Vector associated to a GGM G

`Z.true` True latent clustering

`A.true` True latent adjacency matrix of the graph G

Examples

```
main_noisySBM_GGM(GGMtest$dataMatrix,Meth="Ren",NIG=TRUE,Qup=10,nbOfZ=1,nbCores=1)

#Note : These data were created using the following instructions
n=30
p=10
u=0.1
v=0.3
theta=list(pi=c(1/3,2/3),w=0.25*cbind(c(1/6,1/120),c(1/120,1/6)))
Q=2
Z <- sample(1:Q, p, replace=TRUE, prob=theta$pi)
A <- matrix(0, p, p)
for (i in 1:(p-1)){
    A[i,(i+1):p] <- stats::rbinom(p-i, 1, theta$w[Z[i],Z[(i+1):p]])}
 A.true <- A + t(A)
Omega <- A.true*v
diag(Omega) = abs(min(eigen(Omega)$values)) + 0.1 + u
Sigma = stats::cov2cor(solve(Omega))
X = MASS::mvrnorm(n, rep(0, p), Sigma)
GGMtest=list(dataMatrix=X,Z.true=Z,A.true=A.true)
```

main_noisySBM *Graph Inference from Noisy Data by Multiple Testing*

Description

The `main_noisySBM()` function is a core component of the `noisysbmGGM` package, responsible for applying the greedy algorithm to estimate model parameters, perform node clustering, and conduct a multiple testing procedure to infer the underlying graph. This function is versatile, offering various options and providing useful outputs for further analysis

Usage

```
main_noisySBM(
  X,
  NIG = FALSE,
  threshold = 0.5,
  Nbrepet = 2,
  rho = NULL,
  tau = NULL,
  a = NULL,
  b = NULL,
  c = NULL,
  d = NULL,
  n0 = 1,
  eta0 = 1,
  zeta0 = 1,
  alpha = 0.1,
  Qup = NULL,
  nbCores = parallel::detectCores(),
  nbOfZ = 12,
  sigma0 = 1,
  sigma1 = 1,
  percentageOfPerturbation = 0.3,
  verbatim = TRUE
)
```

Arguments

X	A p-square matrix containing the data
NIG	A Boolean. If FALSE (by default), the variance under the alternative hypothesis in assumed to be known. If TRUE, the variances under the alternatives are unknown and estimated with the NIG method
threshold	Threshold use when updating the latent graphs structure from l-values (by default threshold=0.5)
Nbrepet	Number of times the algorithm is repeated (by default Nbrepet=2)
rho	Hyperparameter of the non-NIG method (by default rho=1)
tau	Hyperparameter of the non-NIG method (by default tau=1)
a	Hyperparameter of the NIG method (by default a=0)
b	Hyperparameter of the NIG method (by default b=1)
c	Hyperparameter of the NIG method (by default c=1)
d	Hyperparameter of the NIG method (by default d=1)
n0	Hyperparameter (by default n0=1)
eta0	Hyperparameter (by default eta0=1)
zeta0	Hyperparameter (by default zeta0=1)
alpha	Level of significance of the multiple testing procedure (by default alpha=0.1)

Qup	Maximal number of cluster (by default Qup =10)
nbCores	Nb of cores to be used during calculations (by default nbCores=parallel::detectCores())
nbOfZ	Nb of initialization (by default nbOfZ=12)
sigma0	standard deviation under the null hypothesis (by default sigma0=1)
sigma1	standard deviation under the alternative hypothesis in the non-NIG method (by default sigma1=1)
percentageOfPerturbation	perturbation during initialization (by default percentageOfPerturbation=0.3)
verbatim	print information messages

Value

A	the adjacency matrix of the inferred graph
Z	the inferred clustering
theta	the parameters of the noisySBM at the end
Q	the number of clusters at the end

Examples

```
main_noisySBM(NSBMtest$dataMatrix,NIG=TRUE,Qup=10,nbOfZ=1,nbCores=1)
```

---

main_noisySBM_GGM *GGM Inference from Noisy Data by Multiple Testing using SILGGM and Drton test statistics*

Description

The main_noisySBM_GGM() function is a key feature of the noisysbmGGM package, dedicated to Gaussian Graphical Model (GGM) inference. This function takes an n-sample of a Gaussian vector of dimension p and provides the GGM associated with the partial correlation structure of the vector. GGM inference is essential in capturing the underlying relationships between the vector's coefficients, helping users uncover meaningful interactions while controlling the number of false discoveries.

Usage

```
main_noisySBM_GGM(
  X,
  Meth = "Ren",
  NIG = NULL,
  threshold = 0.5,
  Nbrepet = 2,
  rho = NULL,
  tau = NULL,
  a = NULL,
```

```
  b = NULL,
  c = NULL,
  d = NULL,
  n0 = 1,
  eta0 = 1,
  zeta0 = 1,
  alpha = 0.1,
  Qup = NULL,
  nbCores = parallel::detectCores(),
  nbOfZ = 12,
  sigma0 = 1,
  sigma1 = 1,
  percentageOfPerturbation = 0.3,
  verbatim = TRUE
)
```

Arguments

X	A n by p matrix containing a n-sample of a p-vector
Meth	Choice of test statistics between "Ren", "Jankova_NW", "Jankova_GL", "Liu_SL", "Liu_L", and "zTransform" (warning "zTransform" only work if n>p)
NIG	A Boolean (automatically chosen according to the selected method : NIG=FALSE except for "Liu_SL" and "Liu_L" test statistics as input). If FALSE, the variance under the alternative hypothesis in assumed to be known. If TRUE, the variances under the alternatives are unknown and estimated with the NIG method.
threshold	Threshold use when updating the latent graphs structure from l-values (by default threshold=0.5)
Nbrepet	Number of times the algorithm is repeated (by default Nbrepet=2)
rho	Hyperparameter of the non-NIG method (by default rho=1)
tau	Hyperparameter of the non-NIG method (by default tau=1)
a	Hyperparameter of the NIG method (by default a=0)
b	Hyperparameter of the NIG method (by default b=1)
c	Hyperparameter of the NIG method (by default c=1)
d	Hyperparameter of the NIG method (by default d=1)
n0	Hyperparameter (by default n0=1)
eta0	Hyperparameter (by default eta0=1)
zeta0	Hyperparameter (by default zeta0=1)
alpha	Level of significance of the multiple testing procedure (by default alpha=0.1)
Qup	Maximal number of cluster (by default Qup =10)
nbCores	Nb of cores to be used during calculations (by default nbCores=parallel::detectCores())
nbOfZ	Nb of initialization (by default nbOfZ=12)
sigma0	standard deviation under the null hypothesis (by default sigma0=1)

sigma1	standard deviation under the alternative hypothesis in the non-NIG method (by default sigma1=1)
percentageOfPerturbation	perturbation during initialization (by default percentageOfPerturbation=0.3)
verbatim	print information messages

Value

A	the adjacency matrix of the inferred graph
Z	the inferred clustering
theta	the parameters of the noisySBM at the end
Q	the number of clusters at the end

#' @examples main_noisySBM_GGM(GGMtest$dataMatrix,Meth="Ren",NIG=TRUE,Qup=10,nbOfZ=1)

See Also

main_noisySBM

`matrixToVec` *matrixToVec*

Description

matrixToVec

Usage

```
matrixToVec(X)
```

Arguments

X	a SYMETRIC matrix

Value

a vector contenting the coefficient of the upper triangle of the matrix X from left to right and from top to bottom.

NSBMtest *NoisySBM for test*

Description

Example of NoisySBM data

Usage

```
NSBMtest
```

Format

`dataMatrix` A square matrix containing the observation of the graph

`theta` True NSBM parameters

`latentZ` True latent clustering

`latentAdj` True latent adjacency matrix

Examples

```
main_noisySBM(NSBMtest$dataMatrix,NIG=TRUE,Qup=10,nbOfZ=1,nbCores=1)

#Note : These data were created using the following instructions
p=50
Q=6
pi=c(1/6,1/6,1/6,1/6,1/6,1/6)
w=c(0.811,0.001,0.001,0.001,0.001,0.001,0.811,0.011,0.001,0.001,0.001,
0.811,0.001,0.001,0.001,0.811,0.001,0.001,0.811,0.011,0.811)
theta=list(pi=pi,w=w,nu0=c(0,1))
theta$nu <- array(0, dim = c(Q*(Q+1)/2, 2))
theta$nu[,1] <- rep(2,21)
theta$nu[,2] <- rep(2,21)
NSBMtest=rnsbm(p,theta)
```

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

---

rnsbm *return a random NSBM*

Description

return a random NSBM

Usage

```
rnsbm(p, theta, modelFamily = "Gauss")
```

Arguments

p	(interger) number of node in the network
theta	=(pi;w;nu0;nu) parameter of the model
modelFamily	the distribution family of the noise under the null hypothesis, which can be "Gauss" (Gaussian), "Gamma", or "Poisson", by default it's 'Gauss'

Value

X the noisy matrix

theta

latentZ the latent clustering

latentA the latent adjacency matrix latent variables we strat by sampling the latent variable Z which is the vector containing the family of each nodes adjacency matrix then we sample the adjacency matrix, conditionally to Z the coordinate of A follow a binomial of a parameter contain in theta$w noisy observations under the null we create a matrix (n,n) X and we initialize all its entry (half of them is undirected) with a sampling of the law under the null then for each entry where A is none zero we sample it according to the law under the alternative

vecToMatrix *vecToMatrix*

Description

vecToMatrix

Usage

```
vecToMatrix(X, p)
```

Arguments

X	a vector
p	(integer) the dimension of the square matrix returned by the function be careful the length of the vector X must be equal to p(p+1)/2

Value

a p by p symetric matrix whose upper triangle coefficients from left to right and from top to bottom are the entries of the vector X

Index