Package 'randnet'

February 12, 2019

Type Package

Title Random Network Model Selection and Parameter Tuning

Version 0.2

Date 2019-02-10

Author Tianxi Li, Elizaveta Levina, Ji Zhu

Maintainer Tianxi Li <tianxili@umich.edu>

Description Model selection and parameter tuning procedures for a class of random network models. The model selection can be done by a general cross-validation framework called ECV from Li et. al. (2016) <arXiv:1612.04717> . Several other model-based and task-specific methods are also included, such as NCV from Chen and Lei (2016) <arXiv:1411.1715>, likelihood ratio method from Wang and Bickel (2015) <arXiv:1502.02069>, spectral methods from Le and Levina (2015) <arXiv:1507.00827>. Many network analysis methods are also implemented, such as the regularized spectral clustering (Amini et. al. 2013 <doi:10.1214/13-AOS1138>) and its degree corrected version and graphon neighborhood smoothing (Zhang et. al. 2015 <arXiv:1509.08588>).

License GPL (>= 2)

Depends Matrix, entropy, AUC

Imports methods, stats, poweRlaw, RSpectra, irlba

NeedsCompilation no

Repository CRAN

Date/Publication 2019-02-12 20:50:03 UTC

R topics documented:

`randnet-package` *Statistical modeling of random networks with model selection and parameter tuning*

Description

The package provides model fitting and estimation functions for some popular random network models. More importantly, it implements a general cross-validation framework for model selection and parameter tuning called ECV. Several other model selection methods are also included.

Details

Package:	randnet
Type:	Package
Version:	0.2
Date:	2019-02-10
License:	GPL (>= 2)

Author(s)

Tianxi Li, Elizaveta Levina, Ji Zhu

Maintainer: Tianxi Li <tianxili@umich.edu>

References

T. Li, E. Levina, and J. Zhu. Network cross-validation by edge sampling. arXiv preprint arXiv:1612.04717, 2016.

K. Chen and J. Lei. Network cross-validation for determining the number of communities in network data. Journal of the American Statistical Association, 113(521):241-251, 2018.

K. Rohe, S. Chatterjee, and B. Yu. Spectral clustering and the high-dimensional stochastic blockmodel. The Annals of Statistics, pages 1878-1915, 2011.

A. A. Amini, A. Chen, P. J. Bickel, and E. Levina. Pseudo-likelihood methods for community detection in large sparse networks. The Annals of Statistics, 41(4):2097-2122, 2013.

Qin, T. & Rohe, K. Regularized spectral clustering under the degree-corrected stochastic blockmodel Advances in Neural Information Processing Systems, 2013, 3120-3128

J. Lei and A. Rinaldo. Consistency of spectral clustering in stochastic block models. The Annals of Statistics, 43(1):215-237, 2014.

C. M. Le, E. Levina, and R. Vershynin. Concentration and regularization of random graphs. Random Structures & Algorithms, 2017.

S. J. Young and E. R. Scheinerman. Random dot product graph models for social networks. In International Workshop on Algorithms and Models for the Web-Graph, pages 138-149. Springer, 2007.

C. M. Le and E. Levina. Estimating the number of communities in networks by spectral methods. arXiv preprint arXiv:1507.00827, 2015.

Zhang, Y.; Levina, E. & Zhu, J. Estimating network edge probabilities by neighbourhood smoothing Biometrika, Oxford University Press, 2017, 104, 771-783

B. Karrer and M. E. Newman. Stochastic blockmodels and community structure in networks. Physical Review E, 83(1):016107, 2011.

Wang, Y. R. & Bickel, P. J. Likelihood-based model selection for stochastic block models The Annals of Statistics, Institute of Mathematical Statistics, 2017, 45, 500-528

Gao, C.; Ma, Z.; Zhang, A. Y. & Zhou, H. H. Achieving optimal misclassification proportion in stochastic block models The Journal of Machine Learning Research, JMLR. org, 2017, 18, 1980-2024

`BHMC.estimate` *Estimates the number of communities under block models by the spectral methods*

Description

Estimates the number of communities under block models by using the spectral properties of network Beth-Hessian matrix with moment correction.

Usage

```
BHMC.estimate(A, K.max = 15)
```

Arguments

`A`	adjacency matrix of the network
`K.max`	the maximum possible number of communities to check

Details

Note that the method cannot distinguish SBM and DCSBM. But it works under either model.

Value

A list of result

`K`	Estimated K
`values`	eigenvalues of the Beth-Hessian matrix

Author(s)

Tianxi Li, Elizaveta Levina, Ji Zhu
Maintainer: Tianxi Li <`tianxili@umich.edu`>

References

C. M. Le and E. Levina. Estimating the number of communities in networks by spectral methods. arXiv preprint arXiv:1507.00827, 2015.

See Also

`LRBIC`,`ECV.Block`, `NCV.select`

Examples

```
dt <- BlockModel.Gen(30,300,K=3,beta=0.2,rho=0.9,simple=FALSE,power=TRUE)


A <- dt$A


bhmc <- BHMC.estimate(A,15)

bhmc
```

`BlockModel.Gen` *Generates networks from degree corrected stochastic block model*

Description

Generates networks from degree corrected stochastic block model, with various options for node degree distribution

Usage

```
BlockModel.Gen(lambda, n, beta = 0, K = 3, w = rep(1, K),
 Pi = rep(1, K)/K, rho = 0, simple = TRUE, power = TRUE,
alpha = 5, degree.seed = NULL)
```

Arguments

`lambda`	average node degree
`n`	size of network
`beta`	out-in ratio: the ratio of between-block edges over within-block edges
`K`	number of communities
`w`	not effective
`Pi`	a vector of community proportion
`rho`	proportion of small degrees within each community if the degrees are from two point mass disbribution. rho >0 gives degree corrected block model. If rho > 0 and simple=TRUE, then generate the degrees from two point mass distribution, with rho porition of 0.2 values and 1-rho proportion of 1 for degree parameters. If rho=0, generate from SBM.
`simple`	Indicator of wether two point mass degrees are used, if rho > 0. If rho=0, this is not effective
`power`	Whether or not use powerlaw distribution for degrees. If FALSE, generate from theta from U(0.2,1); if TRUE, generate theta from powerlaw. Only effective if rho >0, simple=FALSE.
`alpha`	Shape parameter for powerlaw distribution.
`degree.seed`	Can be a vector of a prespecified values for theta. Then the function will do sampling with replacement from the vector to generate theta. It can be used to control noise level between different configuration settings.

Value

A list of

`A`	the generated network adjacency matrix
`g`	community membership
`P`	probability matrix of the network
`theta`	node degree parameter

Author(s)

Tianxi Li, Elizaveta Levina, Ji Zhu
Maintainer: Tianxi Li <`tianxili@umich.edu`>

References

B. Karrer and M. E. Newman. Stochastic blockmodels and community structure in networks. Physical Review E, 83(1):016107, 2011.

A. A. Amini, A. Chen, P. J. Bickel, and E. Levina. Pseudo-likelihood methods for community detection in large sparse networks. The Annals of Statistics, 41(4):2097-2122, 2013.

T. Li, E. Levina, and J. Zhu. Network cross-validation by edge sampling. arXiv preprint arXiv:1612.04717, 2016.

Examples

```
dt <- BlockModel.Gen(30,300,K=3,beta=0.2,rho=0.9,simple=FALSE,power=TRUE)
```

ConsensusClust *clusters nodes by concensus (majority voting) initialized by regularized spectral clustering*

Description

community detection by concensus (majority voting) initialized by regularized spectral clustering

Usage

```
ConsensusClust(A,K,tau=0.25,lap=TRUE)
```

Arguments

A	adjacency matrix
K	number of communities
tau	reguarlization parameter for regularized spectral clustering. Default value is 0.25. Typically set between 0 and 1. If tau=0, no regularization is applied.
lap	indicator. If TRUE, the Laplacian matrix for initializing clustering. If FALSE, the adjacency matrix will be used.

Details

Community detection algorithm by majority voting algorithm of Gao et. al. (2016). When initialized by regularized spectral clustering, it is shown that the clustering accuracy of this algorithm gives minimax rate under the SBM. However, it can slow compared with spectral clustering.

Value

cluster labels

Author(s)

Tianxi Li, Elizaveta Levina, Ji Zhu

Maintainer: Tianxi Li <tianxili@umich.edu>

References

Gao, C.; Ma, Z.; Zhang, A. Y. & Zhou, H. H. Achieving optimal misclassification proportion in stochastic block models The Journal of Machine Learning Research, JMLR. org, 2017, 18, 1980-2024

See Also

reg.SP

Examples

```
dt <- BlockModel.Gen(15,300,K=3,beta=0.2,rho=0)


A <- dt$A


#cc <- ConsensusClust(A,K=3,lap=TRUE)
# takes about 25 seconds

#NMI(cc,dt$g)
```

DCSBM.estimate *Estimates DCSBM model*

Description

Estimates DCSBM model by given community labels

Usage

```
DCSBM.estimate(A, g)
```

Arguments

A	adjacency matrix
g	vector of community labels for the nodes

Details

Estimation is based on maximum likelihood.

Value

A list object of

`Phat`	estimated probability matrix
`B`	the B matrix with block connection probability, up to a scaling constant
`Psi`	vector of of degree parameter theta, up to a scaling constant

Author(s)

Tianxi Li, Elizaveta Levina, Ji Zhu
Maintainer: Tianxi Li <`tianxili@umich.edu`>

References

B. Karrer and M. E. Newman. Stochastic blockmodels and community structure in networks. Physical Review E, 83(1):016107, 2011.

See Also

`SBM.estimate`

Examples

```

dt <- BlockModel.Gen(30,300,K=3,beta=0.2,rho=0.9,simple=FALSE,power=TRUE)


A <- dt$A


ssc <- reg.SSP(A,K=3,lap=TRUE)

est <- DCSBM.estimate(A,ssc$cluster)
```

`ECV.block` *selecting block models by ECV*

Description

Model selection by ECV for SBM and DCSBM. It can be used to select between the two models or given on model (either SBM or DCSBM) and select K.

Usage

```
ECV.block(A, max.K, cv = NULL, B = 3, holdout.p = 0.1, tau = 0, dc.est = 2, kappa = NULL)
```

Arguments

`A`	adjacency matrix
`max.K`	largest possible K for number of communities
`cv`	cross validation fold. The default value is NULL. We recommend to use the argument B instead, doing indpendent sampling.
`B`	number of replications
`holdout.p`	testing set proportion
`tau`	constant for numerical stability only. Not useful for current version.
`dc.est`	estimation method for DCSBM. By defaulty (dc.est=2), the maximum likelihood is used. If dc.est=1, the method used by Chen and Lei (2016) is used, which is less stable according to our observation.
`kappa`	constant for numerical stability only. Not useful for current version.

Details

The current version is based on a simple matrix completion procedure, as described in the paper. The performance can be improved by better matrix completion method that will be implemented in next version. Moreover, the current implementation is better in computational time but less efficient in memory. Another memory efficient implementation will be added in next version.

Value

`impute.err`	average validaiton imputation error
`l2`	average validation L_2 loss under SBM
`dev`	average validation binomial deviance loss under SBM
`auc`	average validation AUC
`dc.l2`	average validation L_2 loss under DCSBM
`dc.dev`	average validation binomial deviance loss under DCSBM
`sse`	average validation SSE
`l2.model`	selected model by L_2 loss
`dev.model`	selected model by binomial deviance loss
`l2.mat, dc.l2.mat,...`	cross-validation loss matrix for B replications

Author(s)

Tianxi Li, Elizaveta Levina, Ji Zhu
Maintainer: Tianxi Li `<tianxili@umich.edu>`

References

T. Li, E. Levina, and J. Zhu. Network cross-validation by edge sampling. arXiv preprint arXiv:1612.04717, 2016.

See Also

`NCV.select`

Examples

```
dt <- BlockModel.Gen(30,300,K=3,beta=0.2,rho=0.9,simple=FALSE,power=TRUE)


A <- dt$A

ecv <- ECV.block(A,6,B=3)

ecv$l2.model
ecv$dev.model


which.min(ecv$l2)
which.min(ecv$dev)

which.min(ecv$dc.l2)
which.min(ecv$dc.dev)

which.max(ecv$auc)
which.min(ecv$sse)
```

`ECV.nSmooth.lowrank` *selecting tuning parameter for neighborhood smoothing estimation of graphon model*

Description

selecting tuning parameter for neighborhood smoothing estimation of graphon model where the tuning parameter is to control estimation smoothness.

Usage

```
ECV.nSmooth.lowrank(A, h.seq, K, cv = NULL, B = 3, holdout.p = 0.1)
```

Arguments

`A`	adjacency matrix
`h.seq`	a sequence of h values to tune. It is suggested h should be in the order of sqrt(log(n)/n).
`K`	the optimal rank for approximation. Can be obtained by rank selection of ECV.
`cv`	cross-validation fold. Recomend to use replication number B instead.
`B`	independent replication number of random splitting
`holdout.p`	proportion of test sample

Details

The neighborhood smoothing estimation can be slow, so the ECV may take long even for moderately large networks.

Value

a list object with

`err`	average validation error for h.seq
`min.index`	index of the minimum error

Author(s)

Tianxi Li, Elizaveta Levina, Ji Zhu
Maintainer: Tianxi Li <`tianxili@umich.edu`>

References

T. Li, E. Levina, and J. Zhu. Network cross-validation by edge sampling. arXiv preprint arXiv:1612.04717, 2016.

Examples

```
set.seed(500)
N <- 300

U = matrix(1:N,nrow=1) / (N+1)
V = matrix(1:N,nrow=1) / (N+1)

W = (t(U))^2
W = W/3*cos(1/(W + 1e-7)) + 0.15



upper.index <- which(upper.tri(W))

A <- matrix(0,N,N)


rand.ind <- runif(length(upper.index))

edge.index <- upper.index[rand.ind < W[upper.index]]

A[edge.index] <- 1

A <- A + t(A)
diag(A) <- 0
```

```
#ecv.rank <- ECV.Rank(A,10,B=3,weighted=FALSE,mode="undirected")

#K.hat <- ecv.rank$auc.rank ## first estimate a good rank


#h.seq <- sqrt(log(N)/N)*seq(0.5,5,by=0.5)


#ecv.nsmooth <- ECV.nSmooth.lowrank(A,h.seq,K=2,B=3) ## nSmooth can be slow

#h <- h.seq[ecv.nsmooth$min.index]

#What <- nSmooth(A,h=h)

#par(mfrow=c(1,2))
#image(t(W[N:1,]))
#image(t(What[N:1,]))
```

ECV.Rank *estimates optimal low rank model for a network*

Description

estimates the optimal low rank model for a network

Usage

```
ECV.Rank(A, max.K, B = 3, holdout.p = 0.1, weighted = TRUE,mode="directed")
```

Arguments

A	adjacency matrix
max.K	maximum possible rank to check
B	number of replications in ECV
holdout.p	test set proportion
weighted	whether the network is weighted. If TRUE, only sum of squared errors are computed. If FALSE, then treat the network as binary and AUC will be computed along with SSE.
mode	Selectign the mode of "directed" or "undirected" for cross-validation.

Details

AUC is believed to be more accurate in many simulations for binary networks. But the computation of AUC is much slower than SSE, even slower than matrix completion steps.

Note that we do not have to assume the true model is low rank. This function simply finds a best low-rank approximation to the true model.

Value

A list of

`sse.rank`	rank selection by SSE loss
`auc.rank`	rank selection by AUC loss
`auc`	auc sequence for each rank candidate
`sse`	sse sequence for each rank candidate

Author(s)

Tianxi Li, Elizaveta Levina, Ji Zhu
Maintainer: Tianxi Li <`tianxili@umich.edu`>

References

T. Li, E. Levina, and J. Zhu. Network cross-validation by edge sampling. arXiv preprint arXiv:1612.04717, 2016.

See Also

`ECV.block`

Examples

```
dt <- BlockModel.Gen(30,300,K=3,beta=0.2,rho=0.9,simple=FALSE,power=TRUE)


A <- dt$A


ecv.rank <- ECV.Rank(A,6,weighted=FALSE,mode="undirected")

ecv.rank
```

`LRBIC` *selecting number of communities by asymptotic likelihood ratio*

Description

selecting number of communities by asymptotic likelihood ratio based the methdo of Wang and Bickel 2015

Usage

```
LRBIC(A, Kmax, lambda = NULL, model = "both")
```

Arguments

`A`	adjacency matrix
`Kmax`	the largest possible number of communities to check
`lambda`	a tuning parameter. By default, will use the number recommended in the paper.
`model`	selecting K under which model. If set to be "SBM", the calculation will be done under SBM. If set to be "DCSBM", the calculation will be done under DCSBM. The default value is "both" so will give two selections under SBM and DCSBM respectively.

Details

Note that the method cannot distinguish SBM and DCSBM, though different calculation is done under the two models. So it is not valid to compare across models. The theoretical analysis of the method is done under maximum likelhood and variational EM. But as suggested in the paper, we use spectral clustering for community detection before fitting maximum likelhood.

Value

a list of

`SBM.K`	estimated number of communities under SBM
`DCSBM.K`	estimated number of communities under DCSBM
`SBM.BIC`	the BIC values for the K sequence under SBM
`DCSBM.BIC`	the BIC values for the K sequence under DCSBM

Author(s)

Tianxi Li, Elizaveta Levina, Ji Zhu
Maintainer: Tianxi Li <`tianxili@umich.edu`>

References

Wang, Y. R. & Bickel, P. J. Likelihood-based model selection for stochastic block models The Annals of Statistics, Institute of Mathematical Statistics, 2017, 45, 500-528

See Also

`BHMC.estimate`, `ECV.block`, `NCV.select`

Examples

```
dt <- BlockModel.Gen(30,300,K=3,beta=0.2,rho=0.9,simple=FALSE,power=TRUE)


A <- dt$A
```

```
### test LRBIC

lrbic <- LRBIC(A,6,model="both")

lrbic$SBM.K

lrbic$DCSBM.K
```

`NCV.select` *selecting block models by NCV*

Description

selecting block models by NCV of Chen and Lei (2016)

Usage

```
NCV.select(A, max.K, cv = 3)
```

Arguments

`A`	adjacency matrix
`max.K`	largest number of communities to check
`cv`	fold of cross-validation

Details

Spectral clustering is used for fitting the block models

Value

a list of

`dev`	the binomial deviance loss under SBM for each K
`l2`	the L_2 loss under SBM for each K
`dc.dev`	the binomial deviance loss under DCSBM for each K
`dc.l2`	the L_2 loss under DCSBM for each K
`dev.model`	the selected model by deviance loss
`l2.model`	the selected model by L_2 loss
`sbm.l2.mat, sbm.dev.mat,....`	the corresponding matrices of loss for each fold (row) and each K value (column)

Author(s)

Tianxi Li, Elizaveta Levina, Ji Zhu
Maintainer: Tianxi Li <tianxili@umich.edu>

References

Chen, K. & Lei, J. Network cross-validation for determining the number of communities in network data Journal of the American Statistical Association, Taylor & Francis, 2018, 113, 241-251

See Also

ECV.block

Examples

```
dt <- BlockModel.Gen(30,300,K=3,beta=0.2,rho=0.9,simple=FALSE,power=TRUE)


A <- dt$A


ncv <- NCV.select(A,6,3)

ncv$l2.model
ncv$dev.model

which.min(ncv$dev)
which.min(ncv$l2)

which.min(ncv$dc.dev)
which.min(ncv$dc.l2)
```

NMI *calculates normalized mutual information*

Description

calculates normalized mutual information, a metric that is commonly used to compare clustering results

Usage

```
NMI(g1, g2)
```

Arguments

`g1`	a vector of cluster labels
`g2`	a vector of cluster labels (same length as g1)

Value

NMI value

Author(s)

Tianxi Li, Elizaveta Levina, Ji Zhu
Maintainer: Tianxi Li <`tianxili@umich.edu`>

Examples

```
dt <- BlockModel.Gen(30,300,K=3,beta=0.2,rho=0.9,simple=FALSE,power=TRUE)


A <- dt$A


ssc <- reg.SSP(A,K=3,lap=TRUE)

NMI(ssc$cluster,dt$g)
```

`nSmooth` *estimates probabilty matrix by neighborhood smoothing*

Description

estimates probabilty matrix by neighborhood smoothing of Zhang et. al. (2015)

Usage

```
nSmooth(A, h = NULL)
```

Arguments

`A`	adjacency matrix
`h`	quantile value used for smoothing. Recommended to be in the scale of sqrt(log(n)/n) where n is the size of the network. The default value is sqrt(log(n)/n) from the paper.

Details

The method assumes a graphon model where the underlying graphon function is piecewise Lipchitz. However, it may be slow for moderately large networks, though it is one of the fastest methods for graphon models.

Value

the probability matrix

Author(s)

Tianxi Li, Elizaveta Levina, Ji Zhu

Maintainer: Tianxi Li <tianxili@umich.edu>

References

Zhang, Y.; Levina, E. & Zhu, J. Estimating network edge probabilities by neighbourhood smoothing Biometrika, Oxford University Press, 2017, 104, 771-783

Examples

```
N <- 300

U = matrix(1:N,nrow=1) / (N+1)
V = matrix(1:N,nrow=1) / (N+1)

W = (t(U))^2
W = W/3*cos(1/(W + 1e-7)) + 0.15



upper.index <- which(upper.tri(W))

A <- matrix(0,N,N)


rand.ind <- runif(length(upper.index))

edge.index <- upper.index[rand.ind < W[upper.index]]

A[edge.index] <- 1

A <- A + t(A)
diag(A) <- 0

#What <- nSmooth(A)
```

RDPG.Gen *generates random networks from random dot product graph model*

Description

generates random networks from random dot product graph model

Usage

```
RDPG.Gen(n, K, directed = TRUE, avg.d = NULL)
```

Arguments

`n`	size of the network
`K`	dimension of latent space
`directed`	whether the network is directed or not
`avg.d`	average node degree of the network (in expectation)

Details

The network is generated according to special formulation mentioned in ECV paper.

Value

a list of

`A`	the adjacency matrix
`P`	the probability matrix

Author(s)

Tianxi Li, Elizaveta Levina, Ji Zhu
Maintainer: Tianxi Li <`tianxili@umich.edu`>

References

S. J. Young and E. R. Scheinerman. Random dot product graph models for social networks. In International Workshop on Algorithms and Models for the Web-Graph, pages 138-149. Springer, 2007. T. Li, E. Levina, and J. Zhu. Network cross-validation by edge sampling. arXiv preprint arXiv:1612.04717, 2016.

Examples

```
dt <- RDPG.Gen(n=600,K=2,directed=TRUE)

A <- dt$A
```

reg.SP *clusters nodes by regularized spectral clustering*

Description

community detection by regularized spectral clustering

Usage

```
reg.SP(A, K, tau = 1, lap = FALSE)
```

Arguments

`A`	adjacency matrix
`K`	number of communities
`tau`	reguarlization parameter. Default value is one. Typically set between 0 and 1. If tau=0, no regularization is applied.
`lap`	indicator. If TRUE, the Laplacian matrix for clustering. If FALSE, the adjacency matrix will be used.

Details

The regularlization is done by adding a small constant to each element of the adjacency matrix. It is shown by such perturbation helps concentration in sparse networks. It is shown to give consistent clustering under SBM.

Value

a list of

`cluster`	cluster labels
`loss`	the loss of Kmeans algorithm

Author(s)

Tianxi Li, Elizaveta Levina, Ji Zhu

Maintainer: Tianxi Li <tianxili@umich.edu>

References

K. Rohe, S. Chatterjee, and B. Yu. Spectral clustering and the high-dimensional stochastic blockmodel. The Annals of Statistics, pages 1878-1915, 2011.

A. A. Amini, A. Chen, P. J. Bickel, and E. Levina. Pseudo-likelihood methods for community detection in large sparse networks. The Annals of Statistics, 41(4):2097-2122, 2013.

J. Lei and A. Rinaldo. Consistency of spectral clustering in stochastic block models. The Annals of Statistics, 43(1):215-237, 2014.

C. M. Le, E. Levina, and R. Vershynin. Concentration and regularization of random graphs. Random Structures & Algorithms, 2017.

See Also

reg.SP

Examples

```
dt <- BlockModel.Gen(30,300,K=3,beta=0.2,rho=0)


A <- dt$A


sc <- reg.SP(A,K=3,lap=TRUE)


NMI(sc$cluster,dt$g)
```

---

reg.SSP *detects communities by regularized spherical spectral clustering*

Description

community detection by regularized spherical spectral clustering

Usage

```
reg.SSP(A, K, tau = 1, lap = FALSE)
```

Arguments

A	adjacency matrix
K	number of communities
tau	reguarlization parameter. Default value is one. Typically set between 0 and 1. If tau=0, no regularization is applied.
lap	indicator. If TRUE, the Laplacian matrix for clustering. If FALSE, the adjacency matrix will be used.

Details

The regularlization is done by adding a small constant to each element of the adjacency matrix. It is shown by such perturbation helps concentration in sparse networks. The difference from spectral clustering (reg.SP) comes from its extra step to normalize the rows of spectral vectors. It is proved that it gives consistent clustering under DCSBM.

Value

a list of

`cluster`	cluster labels
`loss`	the loss of Kmeans algorithm

Author(s)

Tianxi Li, Elizaveta Levina, Ji Zhu

Maintainer: Tianxi Li <tianxili@umich.edu>

References

T. Qin and K. Rohe. Regularized spectral clustering under the degree-corrected stochastic blockmodel. In Advances in Neural Information Processing Systems, pages 3120-3128, 2013.

J. Lei and A. Rinaldo. Consistency of spectral clustering in stochastic block models. The Annals of Statistics, 43(1):215-237, 2014.

See Also

`reg.SP`

Examples

```
dt <- BlockModel.Gen(30,300,K=3,beta=0.2,rho=0.9,simple=FALSE,power=TRUE)


A <- dt$A

ssc <- reg.SSP(A,K=3,lap=TRUE)

NMI(ssc$cluster,dt$g)
```

`SBM.estimate` *estimates SBM parameters given community labels*

Description

estimates SBM parameters given community labels

Usage

```
SBM.estimate(A, g)
```

Arguments

`A`	adjacency matrix
`g`	a vector of community labels

Details

maximum likelihood is used

Value

a list of

`B`	estimated block connection probability matrix
`Phat`	estimated probability matrix
`g`	community labels

Author(s)

Tianxi Li, Elizaveta Levina, Ji Zhu
Maintainer: Tianxi Li <`tianxili@umich.edu`>

References

B. Karrer and M. E. Newman. Stochastic blockmodels and community structure in networks. Physical Review E, 83(1):016107, 2011.

See Also

`DCSBM.estimate`

Examples

```
dt <- BlockModel.Gen(30,300,K=3,beta=0.2,rho=0)


A <- dt$A


sc <- reg.SP(A,K=3,lap=TRUE)
sbm <- SBM.estimate(A,sc$cluster)
sbm$B
```

Index