Package ‘HCD’

August 24, 2019

Type Package

Title Hierarchical Community Detection by Recursive Partitioning

Version 0.1

Date 2019-08-02

Author Tianxi Li, Lihua Lei, Sharmodeep Bhattacharyya, Purnamrita Sarkar, Peter Bickel, and Elizaveta Levina

Maintainer Tianxi Li <tianxili@virginia.edu>

Description Hierarchical community detection on networks by a recursive spectral partitioning strategy, which is shown to be effective and efficient in Li, Lei, Bhattacharyya, Sarkar, Bickel, and Levina (2018) <arXiv:1810.01509>. The package also includes a data generating function for a binary tree stochastic block model, a special case of stochastic block model that admits hierarchy between communities.

License GPL (>= 2)

Imports Matrix, stats, methods, randnet, RSpectra, irlba, data.tree, data.table,stringr,dendextend

NeedsCompilation no

Repository CRAN

Date/Publication 2019-08-24 14:10:02 UTC

R topics documented:

`HCD-package` *Hierarchical community detection by recursive partitioning*

Description

The package provides the implementation of the recursive partitioning strategy to clustering network nodes in a hierarchical way. It also includes the mechanism of generating networks from a binary tree stochastic block model.

Details

Package:	HCD
Type:	Package
Version:	0.1
Date:	2019-08-02
License:	GPL (>= 2)

Author(s)

Tianxi Li, Lihua Lei, Sharmodeep Bhattacharyya, Purnamrita Sarkar, Peter Bickel, and Elizaveta Levina.

Maintainer: Tianxi Li <tianxili@virginia.edu>

References

Tianxi Li, Lihua Lei, Sharmodeep Bhattacharyya, Purnamrita Sarkar, Peter Bickel, and Elizaveta Levina. Hierarchical community detection by recursive partitioning. arXiv:1810.01509

`BTSBM` *Generates networks from binary tree stochastic block model*

Description

Generates networks from binary tree stochastic block model, with provided sequence of connection probability along the tree

Usage

```
BTSBM(n, d, a.seq, lambda, alpha = NULL, N = 1)
```

Arguments

`n`	number of nodes in the network
`d`	number of layers until leaves (excluding the root)
`a.seq`	the connection probability sequence along the tree, a_r, see details in the paper
`lambda`	average node degree, only used when alpha is not provided
`alpha`	the common scaling of the a_r sequence. So at the end, essentially the a_r sequence is a.seq*alpha
`N`	the number of networks to generate from the same model

Value

A list of objections of

`A.list`	the generated network adjacency matrices
`B`	the connection probability matrix between K communities, where K = 2^d
`label`	the vector of community labels for n nodes
`P`	the connection probability matrix between the n nodes. It is the expectation of adjacency matrices, except on the diagonal
`comm.sim.mat`	the binary string similarity matrix between communities
`node.sim.mat`	the binary string similarity matrix between nodes

Author(s)

Tianxi Li, Lihua Lei, Sharmodeep Bhattacharyya, Purnamrita Sarkar, Peter Bickel, and Elizaveta Levina.

Maintainer: Tianxi Li <tianxili@virginia.edu>

References

Tianxi Li, Lihua Lei, Sharmodeep Bhattacharyya, Purnamrita Sarkar, Peter Bickel, and Elizaveta Levina. Hierarchical community detection by recursive partitioning. arXiv:1810.01509

Examples

```
dt <- BTSBM(n=1600,d=4,a.seq=0.2^seq(0,4),lambda=50)
A <- dt$A.list[[1]]
```

`gen.A.from.P` *generates a network from the given connection probability*

Description

Generates an adjacency matrix from a given probability matrix, according independent Bernoulli – the so-called inhomogeneous Erdos-Renyi model. It is used to generate new networks from a given model.

Usage

```
gen.A.from.P(P, undirected = TRUE)
```

Arguments

`P`	connection probability between nodes
`undirected`	logic value. FALSE (default) if the network is undirected, so the adjacency matrix will be symmetric with only upper diagonal entries being generated as independent Bernoulli.

Value

An adjacency matrix

Author(s)

Tianxi Li, Lihua Lei, Sharmodeep Bhattacharyya, Purnamrita Sarkar, Peter Bickel, and Elizaveta Levina.

Maintainer: Tianxi Li <tianxili@virginia.edu>

References

Tianxi Li, Lihua Lei, Sharmodeep Bhattacharyya, Purnamrita Sarkar, Peter Bickel, and Elizaveta Levina. Hierarchical community detection by recursive partitioning. arXiv:1810.01509

HCD *hierarchical community detection with recursive spectral methods*

Description

Hierarchical community by recursive spectral partitioning. It includes the splitting methods of spectral clustering and sign splitting, as well stopping rules for fixed stopping, non-backtracking matrix checking and edge cross-validation.

Usage

```
HCD(A, method = "SS", stopping = "NB", reg = FALSE, n.min = 25, D = NULL,notree=TRUE)
```

Arguments

`A`	adjacency matrix. Can be standard R matrix or dsCMatrix (or other type in package Matrix)
`method`	splitting method. "SS" (default) for sign splitting, "SC" for spectral clustering
`stopping`	stopping rule. "NB" (default) for non-backtracking matrix spectrum, "ECV" for edge cross-validation, "Fix"for fixed D layers of partitioning (needs D value)
`reg`	logic value on whether regularization is needed. By default it is FALSE.Set it to be TRUE will add reguarlization, which help the performance on sparse networks, but it will make the computation slower.
`n.min`	integer number. The algorithm will stop splitting if the current size is <= 2*n.min.
`D`	the number of layers to partition, if stopping=="Fix".
`notree`	logical value on whether the tree and the corresponding similarity will be computed. If TRUE (default), will not produce the data.tree object or the community similarity matrix. Only the cluster label and the tree path strings will be returned. This typically makes the runing faster.

Details

For stopping rules, ECV is nonparametric rank evaluation by cross-validation, a more generally applicable approach without assuming SBM or its variants. ECV is also applicable for weighted networks.So it is believed to be more robust than NB but less effective if the true model is close to BTSBM. However, the ECV is computationally much more intensive.

Notice that the algorithm does not reply on the assumption of the BTSBM. But the estimated probabiilty matrix from the output is based on the BTSBM.

Value

A list of the following objects:

`labels`	detected community labels of nodes
`ncl`	number of clusters from the algorithm

`cluster.tree` a data.tree object for the binary tree between communities

`P` estimated connection probability matrix between n nodes, according to BTSBM

`node.bin.sim.mat`
binary string similarity between nodes

`comm.bin.sim.mat`
binary string similarity between communities

`tree.path` a list of strings to describe the path from root to each community along the tree

Author(s)

Tianxi Li, Lihua Lei, Sharmodeep Bhattacharyya, Purnamrita Sarkar, Peter Bickel, and Elizaveta Levina.

Maintainer: Tianxi Li <tianxili@virginia.edu>

References

Tianxi Li, Lihua Lei, Sharmodeep Bhattacharyya, Purnamrita Sarkar, Peter Bickel, and Elizaveta Levina. Hierarchical community detection by recursive partitioning. arXiv:1810.01509

Examples

```
dt <- BTSBM(n=1600,d=4,a.seq=0.2^seq(0,4),lambda=50)
A <- dt$A.list[[1]]
# you can try various versions of the algorithm as below: the Fix is fastest and ECV is slowest.
system.time(HCD.result <- HCD(A,method="SC",stopping="Fix",D=4))
```

Index