

Package ‘GreedySBTM’

October 18, 2017

Type Package

Title Greedy Stochastic Block Transition Models

Version 1.0

Date 2017-10-10

Description Performs clustering on the nodes of an undirected binary dynamic network, by maximising the exact integrated complete likelihood. The greedy algorithm used is described in Rastelli, R. (2017) “Exact integrated completed likelihood maximisation in a stochastic block transition model for dynamic networks” <arXiv:1710.03551>.

License GPL-3

Imports Rcpp (>= 0.12.10)

LinkingTo Rcpp, RcppArmadillo

NeedsCompilation yes

Author Riccardo Rastelli [aut, cre]

Maintainer Riccardo Rastelli <riccardo.rastelli@wu.ac.at>

Depends R (>= 2.10)

Repository CRAN

Date/Publication 2017-10-18 13:52:06 UTC

R topics documented:

GreedySBTM-package	2
CollapseLabels	2
GreedyICL	3
GreedyInit	4
GreedyMerge	5
ICLExact	6
reality_mining_1392	7
reality_mining_84	8
SBTMProbs	9

Index	10
--------------	-----------

GreedySBTM-package *Greedy Stochastic Block Transition Models*

Description

Performs clustering on the nodes of an undirected binary dynamic network, by maximising the exact integrated complete likelihood. The greedy algorithm used is described in Rastelli, R. (2017) "Exact integrated completed likelihood maximisation in a stochastic block transition model for dynamic networks" <arXiv:1710.03551>.

Author(s)

Riccardo Rastelli

Maintainer: Riccardo Rastelli <riccardo.rastelli@wu.ac.at>

References

Rastelli, R. (2017) "Exact integrated completed likelihood maximisation in a stochastic block transition model for dynamic networks", <https://arxiv.org/abs/1710.03551>

CollapseLabels *CollapseLabels*

Description

Given a matrix of allocations, it performs a permutation on the group labels so that, if K non-empty groups are present, the labels used are exactly $\{1, \dots, K\}$. Note that the value $z[t, i]=0$ is reserved to the inactive nodes: this transformation leaves such allocations unchanged.

Usage

`CollapseLabels(allocations)`

Arguments

`allocations` A matrix whose rows identify partitions of the same elements. The entries must be positive integers. Entries equal to zero are interpreted as missing values and hence are left unchanged.

Details

The labels $\{1, \dots, G\}$ of the clustering provided are mapped into $\{1, \dots, K\}$ (K is less or equal than G) based on their order of appearance in the matrix provided.

Examples

```
set.seed(12345)
allocations <- matrix(sample(0:25, 25, TRUE), 5, 5)
CollapseLabels(allocations = allocations)
```

 GreedyICL

GreedyICL

Description

Runs the GreedyICL algorithm on the provided network to maximise the exact integrated complete likelihood.

Usage

```
GreedyICL(adj_cube, allocations, max_n_iter = 100, verbose = FALSE)
```

Arguments

adj_cube	A binary array of size $N \times N \times T$ representing the dynamic network. The generic entry in position $[i, j, t]$ is equal to 1 if i interacts with j during the t -th time frame, or to 0 otherwise. Only undirected networks with no self-edges are supported, so each slice of the array must be a symmetric matrix with null elements on the diagonal.
allocations	Initial allocations used by the greedy algorithm. This should be a matrix of size $T \times N$ denoting the cluster membership of each node at each time. Values should be strictly positive integers. The value zero should be used to identify inactive nodes.
max_n_iter	A positive integer denoting the maximum number of iterations for the Greedy-ICL maximisation. Defaults to 100 (i.e. each node is updated up to 100 times).
verbose	TRUE or FALSE, indicating whether a lengthy output should be printed out. Defaults to FALSE

Value

computing_time	Number of seconds required to perform the optimisation.
icl_start	Exact log-ICL value for the initial allocations.
icl_trace	Exact log-ICL values for each of the solutions visited during the optimisation.
icl_end	Exact log-ICL value for the optimal allocations found.
allocations	Optimal allocations.

References

Rastelli, R. (2017) "Exact integrated completed likelihood maximisation in a stochastic block transition model for dynamic networks", <https://arxiv.org/abs/1710.03551>

See Also

[GreedyInit](#), [GreedyMerge](#), [CollapseLabels](#).

Examples

```
### A complete example is provided in the documentation of GreedyMerge.
```

GreedyInit

GreedyInit

Description

Initialises the allocation variables using k-means.

Usage

```
GreedyInit(adj_cube, Kup, list_of_inactive_nodes = NULL)
```

Arguments

adj_cube	A binary array of size $N \times N \times T$ representing the dynamic network. The generic entry in position $[i, j, t]$ is equal to 1 if i interacts with j during the t -th time frame, or to 0 otherwise. Only undirected networks with no self-edges are supported, so each slice of the array must be a symmetric matrix with null elements on the diagonal.
Kup	Number of groups: must be a positive integer between 1 and N .
list_of_inactive_nodes	A matrix whose rows identify all of the inactive nodes in the network. The first element of each row denotes the time frame t , while the second element the node label i : the inactive nodes are identified by all such pairs (t, i) .

Value

The function returns a $T \times N$ allocation matrix, with zeros corresponding to inactive nodes.

See Also

[GreedyICL](#), [GreedyMerge](#), [CollapseLabels](#).

Examples

```
### A complete example is provided in the documentation of GreedyMerge.
```

GreedyMerge

GreedyMerge

Description

Runs the hierarchical clustering algorithm to improve the solution obtained through GreedyICL.

Usage

```
GreedyMerge(adj_cube, allocations, verbose = FALSE)
```

Arguments

adj_cube	A binary array of size $N \times N \times T$ representing the dynamic network. The generic entry in position $[i, j, t]$ is equal to 1 if i interacts with j during the t -th time frame, or to 0 otherwise. Only undirected networks with no self-edges are supported, so each slice of the array must be a symmetric matrix with null elements on the diagonal.
allocations	Initial allocations used by the algorithm. This should be a matrix of size $T \times N$ denoting the cluster membership of each node at each time. Values should be strictly positive integers. The value zero should be used to identify inactive nodes.
verbose	TRUE or FALSE, indicating whether a lengthy output should be printed out. Defaults to FALSE

Value

computing_time	Number of seconds required to run the function.
icl_start	Exact log-ICL value for the initial allocations.
icl_end	Exact log-ICL value for the optimal allocations found.
allocations	Optimal allocations.

References

Rastelli, R. (2017) "Exact integrated completed likelihood maximisation in a stochastic block transition model for dynamic networks", <https://arxiv.org/abs/1710.03551>

See Also

[GreedyICL](#), [CollapseLabels](#).

Examples

```

set.seed(12345)
data(reality_mining_84)
tframes <- dim(reality_mining_84)[3]
N <- dim(reality_mining_84)[1]

### Evaluate matrix containing the degrees of each node at each time
degrees <- apply(X = reality_mining_84, MARGIN = c(1,3), FUN = sum)

### create a binary matrix with ones corresponding to inactive nodes
inactive_nodes <- degrees == 0

### create an edgelist listing all of the ids (time, node_label) of inactive nodes
n_inactive_nodes <- sum(inactive_nodes)
list_of_inactive_nodes <- matrix(NA,n_inactive_nodes,2)
index <- 1
for (t in 1:tframes) for (i in 1:N) if (inactive_nodes[i,t])
{
  list_of_inactive_nodes[index,1] = t
  list_of_inactive_nodes[index,2] = i
  index = index + 1
}

### Find starting allocations using k-means
allocations_init <- GreedyInit(reality_mining_84, 20, list_of_inactive_nodes)

### Run the GreedyICL algorithm (this may take some time)
output_greedy <- GreedyICL(reality_mining_84, allocations_init, max_n_iter = 1)
### max_n_iter is set to 1 to speed up the demonstration:
### please always use max_n_iter = 100 in applications

### Run the hierarchical clustering routine
output_merge <- GreedyMerge(adj_cube = reality_mining_84, allocations = output_greedy$allocations)
str(output_merge)

```

ICLExact

ICLExact

Description

Evaluates the exact log integrated complete likelihood for a given allocation matrix.

Usage

```
ICLExact(adj_cube, allocations, verbose = FALSE)
```

Arguments

adj_cube	A binary array of size $N \times N \times T$ representing the dynamic network. The generic entry in position $[i, j, t]$ is equal to 1 if i interacts with j during the t -th time frame, or to 0 otherwise. Only undirected networks with no self-edges are supported, so each slice of the array must be a symmetric matrix with null elements on the diagonal.
allocations	This should be a matrix of size $T \times N$ denoting the cluster membership of each node at each time. Values should be strictly positive integers. The value zero should be used to identify inactive nodes.
verbose	TRUE or FALSE, indicating whether a lengthy output should be printed out. Defaults to FALSE

Value

computing_time	Number of seconds required for the evaluation.
prior_value	Contribution to the exact log-ICL given by the marginal prior on the allocations.
likelihood_value	Contribution to the exact log-ICL given by the marginal likelihood.
icl_value	Exact log-ICL value for the allocation matrix given.

Examples

```
set.seed(12345)
data(reality_mining_84)
tframes <- dim(reality_mining_84)[3]
N <- dim(reality_mining_84)[1]
allocations <- matrix(sample(1:10, tframes*N, TRUE), tframes, N)
ICLExact(adj_cube = reality_mining_84, allocations = allocations, verbose = FALSE)
```

reality_mining_1392 *Reality Mining dataset*

Description

Human contact data among 96 students of the Massachusetts Institute of Technology (MIT), collected by the Reality Mining experiment performed in 2004 as part of the Reality Commons project. The dataset is an adaptation of the proximity dataset extracted and made public by KONECT (url provided below).

Usage

```
data(reality_mining_1392)
```

Format

The object is a binary adjacency cube of size [96x96x1392]. The time frames correspond to intervals of 4 hours. The entry $[i, j, t]$ is equal to 1 if students i and j were physically close to each other in the time interval $[t-1, t]$. The dataset spans across all 9 months of the study, i.e. from 14 Sept 2004 to 5 May 2005.

Source

<http://realitycommons.media.mit.edu/realitymining.html>

<http://konect.uni-koblenz.de/networks/mit>

References

Nathan Eagle and Alex (Sandy) Pentland. Reality Mining: Sensing complex social systems. *Personal Ubiquitous Computing*, 10(4):255–268, 2006.

reality_mining_84 *Reality Mining small dataset*

Description

Human contact data among 96 students of the Massachusetts Institute of Technology (MIT), collected by the Reality Mining experiment performed in 2004 as part of the Reality Commons project. The dataset is a subset and an adaptation of the proximity dataset extracted and made public by KONECT (url provided below).

Usage

```
data(reality_mining_84)
```

Format

The object is a binary adjacency cube of size [96x96x84]. The time frames correspond to intervals of 4 hours. The entry $[i, j, t]$ is equal to 1 if students i and j were physically close to each other in the time interval $[t-1, t]$. The dataset contains only the first two weeks of interactions. The full dataset is available in this package under the name `reality_mining_1392`.

Source

<http://realitycommons.media.mit.edu/realitymining.html>

<http://konect.uni-koblenz.de/networks/mit>

References

Nathan Eagle and Alex (Sandy) Pentland. Reality Mining: Sensing complex social systems. *Personal Ubiquitous Computing*, 10(4):255–268, 2006.

SBTMProbs

*SBTM parameter estimators***Description**

Evaluates estimates for the model parameters of a Stochastic Block Transition Model. This function can summarise the results obtained through the [GreedyICL](#) and [GreedyMerge](#) algorithms.

Usage

```
SBTMProbs(adj_cube, allocations)
```

Arguments

adj_cube	A binary array of size $N \times N \times T$ representing the dynamic network. The generic entry in position $[i, j, t]$ is equal to 1 if i interacts with j during the t -th time frame, or to 0 otherwise. Only undirected networks with no self-edges are supported, so each slice of the array must be a symmetric matrix with null elements on the diagonal.
allocations	A matrix of size $T \times N$ denoting the cluster membership of each node at each time. Values should be strictly positive integers. The value zero should be used to identify the inactive nodes.

Value

Pi	A matrix of size $(K+1) \times (K+1)$. The generic entry $[g, h]$ denotes the estimated probability that a node moves from group $g-1$ to group $h-1$. The labels are shifted because the allocations take values from 0 to K , with zeros corresponding to inactive nodes. The first row and first column denote the transition probabilities for the group of inactive nodes.
Theta	Connection probability matrix of size $K \times K$ for the SBM part of the model. In this case the group of inactive nodes is omitted. The entry in $[g, h]$ is the estimated probability that an edge between a node in group g and a node in group h is realised. Note that this is only for the case where no information is available regarding the presence or absence of the same edge in the previous time-frame.
P	Edge creation probability matrix of size $K \times K$ for the SBTM part of the model. As in Theta, the group of inactive nodes is omitted. The entry in $[g, h]$ is the probability that an edge between a node in group g and a node in group h changes its value from 0 to 1.
Q	Edge destruction probability matrix of size $K \times K$ for the SBTM part of the model. As in Theta, the group of inactive nodes is omitted. The entry in $[g, h]$ is the probability that an edge between a node in group g and a node in group h changes its value from 1 to 0.

See Also

[GreedyICL](#), [GreedyMerge](#).

Index

*Topic **datasets**

reality_mining_1392, [7](#)

reality_mining_84, [8](#)

*Topic **package**

GreedySBTM-package, [2](#)

CollapseLabels, [2](#), [4](#), [5](#)

GreedyICL, [3](#), [4](#), [5](#), [9](#)

GreedyInit, [4](#), [4](#)

GreedyMerge, [4](#), [5](#), [9](#)

GreedySBTM (GreedySBTM-package), [2](#)

GreedySBTM-package, [2](#)

ICLExact, [6](#)

reality_mining_1392, [7](#)

reality_mining_84, [8](#)

SBTMProbs, [9](#)