

Package ‘rCausalMGM’

September 20, 2022

Type Package

Title What the Package Does Using Title Case

Version 1.0

Date 2019-12-12

Author Your Name

Maintainer Takis Benos <benos@pitt.edu>

Description More details about what the package does. See
<[http://cran.r-project.org/doc/manuals/r-release/R-exts.html#](http://cran.r-project.org/doc/manuals/r-release/R-exts.html#The-DESCRIPTION-file)
[The-DESCRIPTION-file](#)> for details on how to write this
part.

License GPL (>= 2)

Imports Rcpp (>= 1.0.3)

LinkingTo Rcpp, RcppArmadillo, BH, RcppThread

SystemRequirements C++14

RoxygenNote 7.2.1

Encoding UTF-8

R topics documented:

adjMat2Graph	2
bootstrap	3
cfcf	4
cpcStable	5
createCPDAG	6
createMoral	6
data.n100.p25	7
data.n1000.p100	7
fci50	8
fciMax	9
fciStable	10
loadGraph	11
mgm	11

mgmPath	12
orientationSHD	12
pc50	13
pcMax	14
pcStable	15
print.graph	16
printGraph	16
saveGraph	17
SHD	17
skeletonSHD	18
steps	18
Index	20

adjMat2Graph	<i>Convert an adjacency matrix into a graph</i>
--------------	---

Description

Convert an adjacency matrix into a graph

Usage

adjMat2Graph(adj, nodes, directed = FALSE)

Arguments

- adj The adjacency matrix, NxN
- nodes The names of the nodes, length N
- directed TRUE if the graph should be directed, default FALSE

Value

The graph representation of the adjacency list

Examples

```
mat <- matrix(sample(c(0,1), 16, replace=TRUE), nrow=4)
nodes <- c("X1", "X2", "X3", "X4")
g <- rCausalMGM::adjMat2Graph(mat, nodes, directed=TRUE)
```

bootstrap	<i>Runs bootstrapping for a selected causal discovery algorithm on the dataset.</i>
-----------	---

Description

Runs bootstrapping for a selected causal discovery algorithm on the dataset.

Usage

```
bootstrap(
  df,
  algorithm = as.character(c("mgm-pc50", "mgm", "pc", "cpc", "pcm", "pc50", "fci",
    "cfci", "fcim", "mgm-pc", "mgm-cpc", "mgm-pcm", "mgm-fci", "mgm-cfci", "mgm-fcim",
    "mgm-fci50")),
  ensemble = as.character(c("highest", "majority")),
  lambda = as.numeric(c(0.2, 0.2, 0.2)),
  alpha = 0.05,
  numBoots = 20L,
  threads = -1L,
  rank = FALSE,
  verbose = FALSE
)
```

Arguments

df	The dataframe
algorithm	string indicating the name of the causal discovery algorithm to bootstrap. Causal discovery algorithms can be run alone or with mgm to learn an initial graph. Options include "mgm", "pc", "cpc", "pcm", "pc50", "fci", "cfci", "fcim", "mgm-pc", "mgm-cpc", "mgm-pcm", "mgm-pc50", "mgm-fci", "mgm-cfci", "mgm-fcim", "mgm-fci50." The default value is set to "mgm-pc50."
ensemble	Method for constructing an ensemble graph from bootstrapped graphs. Options include "highest", which orients edges according to the orientation with the highest bootstrap probability, or "majority", which only orients edges if they have an orientation with a bootstrap probability > 0.5. Otherwise, the adjacency is included but the edge is left unoriented. The default value is "highest."
lambda	A vector of three lambda values - the first for continuous-continuous interaction, the second for continuous-discrete, and the third for discrete-discrete. Defaults to c(0.2, 0.2, 0.2). If a single value is provided, all three values in the vector will be set to that value.
alpha	The p value below which results are considered significant. Defaults to 0.05.
numBoots	The number of bootstrap samples to run. Defaults to 20.
threads	The number of consumer threads to create during multi-threaded steps. If -1, defaults to number of available processors.

rank	Whether or not to use rank-based associations as opposed to linear
verbose	Whether or not to output additional information. Defaults to FALSE.

Value

The calculated search graph with a table of edge stabilities

Examples

```
data("data.n100.p25")
g.boot <- rCausalMGM::bootstrap(data.n100.p25)
```

cfc	<i>Runs the causal algorithm CFCI-Stable on a dataset</i>
-----	---

Description

Runs the causal algorithm CFCI-Stable on a dataset

Usage

```
cfc(
  df,
  initialGraph = NULL,
  alpha = 0.05,
  threads = -1L,
  fdr = FALSE,
  rank = FALSE,
  verbose = FALSE
)
```

Arguments

df	The dataframe
initialGraph	An initial undirected graph to use as a starting point. If NULL, a full graph will be used. Defaults to NULL.
alpha	The p value below which results are considered significant. Defaults to 0.05.
threads	The number of consumer threads to create during multi-threaded steps. If -1, defaults to number of available processors.
fdr	Whether or not to run with FDR correction for the adjacencies.
rank	Whether or not to use rank-based associations as opposed to linear
verbose	Whether or not to output additional information. Defaults to FALSE.

Value

The calculated search graph

Examples

```
data("data.n100.p25")
ig <- rCausalMGM::mgm(data.n100.p25)
g <- rCausalMGM::cfci(data.n100.p25, initialGraph = ig)
```

cpcStable

*Runs the causal algorithm CPC-Stable on a dataset***Description**

Runs the causal algorithm CPC-Stable on a dataset

Usage

```
cpcStable(
  df,
  initialGraph = NULL,
  alpha = 0.05,
  threads = -1L,
  fdr = FALSE,
  rank = FALSE,
  verbose = FALSE
)
```

Arguments

df	The dataframe
initialGraph	An initial undirected graph to use as a starting point. If NULL, a full graph will be used. Defaults to NULL.
alpha	The p value below which results are considered significant. Defaults to 0.05.
threads	The number of consumer threads to create during multi-threaded steps. If -1, defaults to number of available processors.
fdr	Whether or not to run with FDR correction for the adjacencies.
rank	Whether or not to use rank-based associations as opposed to linear
verbose	Whether or not to output additional information. Defaults to FALSE.

Value

The calculated search graph

Examples

```
data("data.n100.p25")
ig <- rCausalMGM::mgm(data.n100.p25)
g <- rCausalMGM::cpcStable(data.n100.p25, initialGraph = ig)
```

createCPDAG	<i>Create the completed partially directed acyclic graph (CPDAG) for the input directed acyclic graph (DAG). The CPDAG represents the Markov equivalence class of the true causal DAG. The PC algorithms are only identifiable up to the Markov equivalence class, so assessments of causal structure recovery should be compared to the CPDAG rather than the causal DAG.</i>
-------------	--

Description

Create the completed partially directed acyclic graph (CPDAG) for the input directed acyclic graph (DAG). The CPDAG represents the Markov equivalence class of the true causal DAG. The PC algorithms are only identifiable up to the Markov equivalence class, so assessments of causal structure recovery should be compared to the CPDAG rather than the causal DAG.

Usage

```
createCPDAG(graph)
```

Arguments

graph	The graph object used to generate the CPDAG. Should be the ground-truth causal DAG
-------	--

Value

The CPDAG corresponding to the input DAG

Examples

```
data(dag_n10000_p10)
cpdag <- rCausalMGM::createCPDAG(dag_n10000_p10)
```

createMoral	<i>Create the moral graph for the input directed acyclic graph (DAG). The moral graph is the equivalent undirected representation corresponding to the input DAG. The MGM algorithm learns the undirected moral graph for a corresponding causal DAG, so assessments of structure recovery should be compared to the moral graph rather than the causal DAG.</i>
-------------	--

Description

Create the moral graph for the input directed acyclic graph (DAG). The moral graph is the equivalent undirected representation corresponding to the input DAG. The MGM algorithm learns the undirected moral graph for a corresponding causal DAG, so assessments of structure recovery should be compared to the moral graph rather than the causal DAG.

Usage

```
createMoral(graph)
```

Arguments

graph The graph object used to generate the moral graph. Should be the ground-truth causal DAG

Value

The moral graph corresponding to the input DAG

Examples

```
data(dag_n10000_p10)
moral <- rCausalMGM::createMoral(dag_n10000_p10)
```

data.n100.p25	<i>A small simulated dataset with 25 variables (13 discrete, 12 continuous) and 100 samples. Generated using the true graph graph.n100.p25.txt</i>
---------------	--

Description

A small simulated dataset with 25 variables (13 discrete, 12 continuous) and 100 samples. Generated using the true graph graph.n100.p25.txt

Author(s)

Tyler Lovelace <tyl15@pitt.edu>

data.n1000.p100	<i>A large simulated dataset with 100 variables (50 discrete, 50 continuous) and 1000 samples. Generated using the true graph graph.n1000.p100.txt</i>
-----------------	--

Description

A large simulated dataset with 100 variables (50 discrete, 50 continuous) and 1000 samples. Generated using the true graph graph.n1000.p100.txt

Author(s)

Tyler Lovelace <tyl15@pitt.edu>

fci50

*Runs the causal algorithm FCI50 Stable on a dataset***Description**

Runs the causal algorithm FCI50 Stable on a dataset

Usage

```
fci50(
  df,
  initialGraph = NULL,
  alpha = 0.05,
  threads = -1L,
  fdr = FALSE,
  rank = FALSE,
  verbose = FALSE
)
```

Arguments

df	The dataframe
initialGraph	An initial undirected graph to use as a starting point. If NULL, a full graph will be used. Defaults to NULL.
alpha	The p value below which results are considered significant. Defaults to 0.05.
threads	The number of consumer threads to create during multi-threaded steps. If -1, defaults to number of available processors.
fdr	Whether or not to run with FDR correction for the adjacencies.
rank	Whether or not to use rank-based associations as opposed to linear
verbose	Whether or not to output additional information. Defaults to FALSE.

Value

The calculated search graph

Examples

```
data("data.n100.p25")
ig <- rCausalMGM::mgm(data.n100.p25)
g <- rCausalMGM::fci50(data.n100.p25, initialGraph = ig)
```


fciMax

*Runs the causal algorithm FCI-Max on a dataset***Description**

Runs the causal algorithm FCI-Max on a dataset

Usage

```
fciMax(
  df,
  initialGraph = NULL,
  alpha = 0.05,
  threads = -1L,
  fdr = FALSE,
  rank = FALSE,
  verbose = FALSE
)
```

Arguments

df	The dataframe
initialGraph	An initial undirected graph to use as a starting point. If NULL, a full graph will be used. Defaults to NULL.
alpha	The p value below which results are considered significant. Defaults to 0.05.
threads	The number of consumer threads to create during multi-threaded steps. If -1, defaults to number of available processors.
fdr	Whether or not to run with FDR correction for the adjacencies.
rank	Whether or not to use rank-based associations as opposed to linear
verbose	Whether or not to output additional information. Defaults to FALSE.

Value

The calculated search graph

Examples

```
data("data.n100.p25")
ig <- rCausalMGM::mgm(data.n100.p25)
g <- rCausalMGM::fciMax(data.n100.p25, initialGraph = ig)
```

fciStable

*Runs the causal algorithm FCI-Stable on a dataset***Description**

Runs the causal algorithm FCI-Stable on a dataset

Usage

```
fciStable(
  df,
  initialGraph = NULL,
  alpha = 0.05,
  threads = -1L,
  fdr = FALSE,
  rank = FALSE,
  verbose = FALSE
)
```

Arguments

df	The dataframe
initialGraph	An initial undirected graph to use as a starting point. If NULL, a full graph will be used. Defaults to NULL.
alpha	The p value below which results are considered significant. Defaults to 0.05.
threads	The number of consumer threads to create during multi-threaded steps. If -1, defaults to number of available processors.
fdr	Whether or not to run with FDR correction for the adjacencies.
rank	Whether or not to use rank-based associations as opposed to linear
verbose	Whether or not to output additional information. Defaults to FALSE.

Value

The calculated search graph

Examples

```
data("data.n100.p25")
ig <- rCausalMGM::mgm(data.n100.p25)
g <- rCausalMGM::fciStable(data.n100.p25, initialGraph = ig)
```

loadGraph	<i>Load a graph from a file</i>
-----------	---------------------------------

Description

Load a graph from a file

Usage

```
loadGraph(filename)
```

Arguments

filename	The graph file
----------	----------------

Value

The graph as a List object, which can be passed into search functions

mgm	<i>Calculate the MGM graph on a dataset</i>
-----	---

Description

Calculate the MGM graph on a dataset

Usage

```
mgm(df, lambda = as.numeric(c(0.2, 0.2, 0.2)), rank = FALSE, verbose = FALSE)
```

Arguments

df	The dataframe
lambda	A vector of three lambda values - the first for continuous-continuous interaction, the second for continuous-discrete, and the third for discrete-discrete. Defaults to c(0.2, 0.2, 0.2). If a single value is provided, all three values in the vector will be set to that value.
rank	Whether or not to use rank-based associations as opposed to linear
verbose	Whether or not to output additional information. Defaults to FALSE.

Value

The calculated MGM graph

Examples

```
data("data.n100.p25")
g <- rCausalMGM::mgm(data.n100.p25)
```

mgmPath	<i>Calculate the solution path for an MGM graph on a dataset</i>
---------	--

Description

Calculate the solution path for an MGM graph on a dataset

Usage

```
mgmPath(df, lambdas = NULL, nLambda = 30L, rank = FALSE, verbose = FALSE)
```

Arguments

df	The dataframe
lambdas	A range of lambda values used to calculate a solution path for MGM. If NULL, lambdas is set to nLambda logarithmically spaced values from $10 \cdot \sqrt{\log_{10}(p)/n}$ to $\sqrt{\log_{10}(p)/n}$. Defaults to NULL.
nLambda	The number of lambda values to fit an MGM for when lambdas is NULL
rank	Whether or not to use rank-based associations as opposed to linear
verbose	Whether or not to output additional information. Defaults to FALSE.

Value

The calculated MGM graph

Examples

```
data("data.n100.p25")
g <- rCausalMGM::mgmPath(data.n100.p25)
```

orientationSHD	<i>Calculate the orientation Structural Hamming Distance (SHD) between two graphs. This only counts the incorrect edge endpoints for edges present in both graphs, and does not consider differences in the graph skeleton. Each different endpoint adds 0.5 to the orientation SHD (i.e. A o-> B vs. A -> B). Thus, a completely misoriented edge adds 1 to the orientation SHD (i.e. A o-o B vs. A <- B).</i>
----------------	--

Description

Calculate the orientation Structural Hamming Distance (SHD) between two graphs. This only counts the incorrect edge endpoints for edges present in both graphs, and does not consider differences in the graph skeleton. Each different endpoint adds 0.5 to the orientation SHD (i.e. A o-> B vs. A -> B). Thus, a completely misoriented edge adds 1 to the orientation SHD (i.e. A o-o B vs. A <- B).

Usage

```
orientationSHD(graph1, graph2)
```

Arguments

graph1	A graph object
graph2	A graph object

Value

The skeleton SHD between the two graph objects

Examples

```
data("data.n100.p25")
g <- rCausalMGM::mgm(data.n100.p25)
rCausalMGM::printGraph(g)
```

pc50	<i>Runs the causal algorithm PC50 on a dataset</i>
------	--

Description

Runs the causal algorithm PC50 on a dataset

Usage

```
pc50(
  df,
  initialGraph = NULL,
  alpha = 0.05,
  threads = -1L,
  fdr = FALSE,
  rank = FALSE,
  verbose = FALSE
)
```

Arguments

df	The dataframe
initialGraph	An initial undirected graph to use as a starting point. If NULL, a full graph will be used. Defaults to NULL.
alpha	The p value below which results are considered significant. Defaults to 0.05.
threads	The number of consumer threads to create during multi-threaded steps. If -1, defaults to number of available processors.
fdr	Whether or not to run with FDR correction for the adjacencies.
rank	Whether or not to use rank-based associations as opposed to linear
verbose	Whether or not to output additional information. Defaults to FALSE.

Value

The calculated search graph

Examples

```
data("data.n100.p25")
ig <- rCausalMGM::mgm(data.n100.p25)
g <- rCausalMGM::pc50(data.n100.p25, initialGraph = ig)
```

pcMax

Runs the causal algorithm PC-Max on a dataset

Description

Runs the causal algorithm PC-Max on a dataset

Usage

```
pcMax(
  df,
  initialGraph = NULL,
  alpha = 0.05,
  threads = -1L,
  fdr = FALSE,
  rank = FALSE,
  verbose = FALSE
)
```

Arguments

df	The dataframe
initialGraph	An initial undirected graph to use as a starting point. If NULL, a full graph will be used. Defaults to NULL.
alpha	The p value below which results are considered significant. Defaults to 0.05.
threads	The number of consumer threads to create during multi-threaded steps. If -1, defaults to number of available processors.
fdr	Whether or not to run with FDR correction for the adjacencies.
rank	Whether or not to use rank-based associations as opposed to linear
verbose	Whether or not to output additional information. Defaults to FALSE.

Value

The calculated search graph

Examples

```
data("data.n100.p25")
ig <- rCausalMGM::mgm(data.n100.p25)
g <- rCausalMGM::pcMax(data.n100.p25, initialGraph = ig)
```

pcStable

*Runs the causal algorithm PC-Stable on a dataset***Description**

Runs the causal algorithm PC-Stable on a dataset

Usage

```
pcStable(
  df,
  initialGraph = NULL,
  alpha = 0.05,
  threads = -1L,
  fdr = FALSE,
  rank = FALSE,
  verbose = FALSE
)
```

Arguments

df	The dataframe
initialGraph	An initial undirected graph to use as a starting point. If NULL, a full graph will be used. Defaults to NULL.
alpha	The p value below which results are considered significant. Defaults to 0.05.
threads	The number of consumer threads to create during multi-threaded steps. If -1, defaults to number of available processors.
fdr	Whether or not to run with FDR correction for the adjacencies.
rank	Whether or not to use rank-based associations as opposed to linear
verbose	Whether or not to output additional information. Defaults to FALSE.

Value

The calculated search graph

Examples

```
data("data.n100.p25")
ig <- rCausalMGM::mgm(data.n100.p25)
g <- rCausalMGM::pcStable(data.n100.p25, initialGraph = ig)
```

print.graph	<i>A print override function for the graph</i>
-------------	--

Description

A print override function for the graph

Usage

```
## S3 method for class 'graph'  
print(x, ...)
```

Arguments

x	The graph object
---	------------------

printGraph	<i>Display a graph object as text</i>
------------	---------------------------------------

Description

Display a graph object as text

Usage

```
printGraph(graph)
```

Arguments

graph	The graph object
-------	------------------

Examples

```
data("data.n100.p25")  
g <- rCausalMGM::mgm(data.n100.p25)  
rCausalMGM::printGraph(g)
```

saveGraph	<i>Save a graph to a file</i>
-----------	-------------------------------

Description

Save a graph to a file

Usage

```
saveGraph(list, filename)
```

Arguments

list	The graph object
filename	The graph file

Examples

```
data("data.n100.p25")
g <- rCausalMGM::mgm(data.n100.p25)
rCausalMGM::saveGraph(g, "graphs/mgm_graph.txt")
```

SHD	<i>Calculate the Structural Hamming Distance (SHD) between two graphs. This is the sum of the skeleton SHD and the orientation SHD.</i>
-----	---

Description

Calculate the Structural Hamming Distance (SHD) between two graphs. This is the sum of the skeleton SHD and the orientation SHD.

Usage

```
SHD(graph1, graph2)
```

Arguments

graph1	A graph object
graph2	A graph object

Value

The SHD between the two graph objects

Examples

```
data("data.n100.p25")
g <- rCausalMGM::mgm(data.n100.p25)
rCausalMGM::printGraph(g)
```

skeletonSHD	<i>Calculate the skeleton Structural Hamming Distance (SHD) between two graphs. This only counts the missing and added edges, and does not consider edge orientation</i>
-------------	--

Description

Calculate the skeleton Structural Hamming Distance (SHD) between two graphs. This only counts the missing and added edges, and does not consider edge orientation

Usage

```
skeletonSHD(graph1, graph2)
```

Arguments

graph1	A graph object
graph2	A graph object

Value

The skeleton SHD between the two graph objects

Examples

```
data("data.n100.p25")
g <- rCausalMGM::mgm(data.n100.p25)
rCausalMGM::printGraph(g)
```

steps	<i>Calculates the optimal lambda values for the MGM algorithm using StEPS and run the algorithm using those values. Optimal values are printed</i>
-------	--

Description

Calculates the optimal lambda values for the MGM algorithm using StEPS and run the algorithm using those values. Optimal values are printed

Usage

```
steps(
  df,
  lambdas = NULL,
  nLambda = 20L,
  g = 0.05,
  numSub = 20L,
  subSize = -1L,
  leaveOneOut = FALSE,
  computeStabs = FALSE,
  threads = -1L,
  rank = FALSE,
  verbose = FALSE
)
```

Arguments

df	The dataframe
lambdas	A range of lambda values assessed for stability by the StEPS algorithm. If NULL, lambdas is set to nLambda logarithmically spaced values from $10 \cdot \sqrt{\log_{10}(p)/n}$ to $\sqrt{\log_{10}(p)/n}$. Defaults to NULL.
nLambda	The number of lambda values to fit an MGM for when lambdas is NULL
g	The gamma parameter for STEPS. Defaults to 0.05
numSub	The number of subsets to split the data into. Defaults to 20
subSize	The size of the subsamples used for STEPS. If the value is -1, the size of the subsamples is set to $\text{floor}(10 \cdot \sqrt{n})$. If the value is in the range (0,1), the size of the subsamples is set to $\text{floor}(\text{subSize} * n)$. Otherwise, if subSize is in the range [1,n), the size of the subsamples is set to subSize. Defaults to -1.
leaveOneOut	If TRUE, performs leave-one-out subsampling. Defaults to FALSE.
computeStabs	If TRUE, stability values are calculated. Defaults to FALSE.
threads	The number of consumer threads to create during multi-threaded steps. If -1, defaults to number of available processors.
rank	Whether or not to use rank-based associations as opposed to linear
verbose	Whether or not to output additional information. Defaults to FALSE.

Value

The calculated MGM graph

Examples

```
data("data.n100.p25")
g <- rCausalMGM::steps(data.n100.p25)
```

Index

* **data**

data.n100.p25, [7](#)

data.n1000.p100, [7](#)

adjMat2Graph, [2](#)

bootstrap, [3](#)

cfci, [4](#)

cpcStable, [5](#)

createCPDAG, [6](#)

createMoral, [6](#)

data.n100.p25, [7](#)

data.n1000.p100, [7](#)

fci50, [8](#)

fciMax, [9](#)

fciStable, [10](#)

loadGraph, [11](#)

mgm, [11](#)

mgmPath, [12](#)

orientationSHD, [12](#)

pc50, [13](#)

pcMax, [14](#)

pcStable, [15](#)

print.graph, [16](#)

printGraph, [16](#)

saveGraph, [17](#)

SHD, [17](#)

skeletonSHD, [18](#)

steps, [18](#)