Package 'cibn'

January 15, 2021

Type Package

Title Causal Independence Bayesian Networks

Version 0.0

Date 2021-01-14

Author Alessandro Magrini

Maintainer Alessandro Magrini <alessandro.magrini@unifi.it>

Description Elicitation, estimation and inference functionalities for Bayesian networks under the causal independence assumption.

Depends R (>= 3.5.0), graph, gRbase, gRain

License GPL-2

NeedsCompilation no

R topics documented:

`cibn-package` *Causal Independence Bayesian Networks*

Description

Elicitation, estimation and inference functionalities for Bayesian networks under the causal independence assumption.

Details

Package:	cibn
Type:	Package
Version:	0.0
Date:	2021-01-07
License:	GPL-2

Causal independence Bayesian networks (Magrini, 2021) are Bayesian networks with non-interacting parent variables (causal independence assumption). They allow three exaustive types of variables (graded, double-graded and multi-valued nominal variables) and admit the Causal Independence Decomposition (CID), which increases efficiency of elicitation, estimation and inference. Causal interactions can be added upon need. The main functions of the package are:

- new.cibn, to create a new network based on prior knowledge;
- `update.cibn`, to update an existing network based on possibly incomplete data (not still implemented but available soon);
- query.cibn, to perform exact inference in a network through an interface to the `gRain` package;
- sample.cibn, to draw a random sample from a network.

Author(s)

Alessandro Magrini <alessandro.magrini@unifi.it>

References

A. Magrini (2021). Efficient decomposition of Bayesian networks with non-graded variables. To be appeared on *International Journal of Statistics and Probability*, 10(2).

`as.grain` *Conversion into* `grain` *class*

Description

Convert an object of class `cibn` into an object of class `grain`.

Usage

```
as.grain(x)
```

Arguments

`x`	An object of class `cibn`.

Value

An object of class `grain`.

See Also

new.cibn.

Examples

```
data(bankrisk_code)
bankrisk_bn <- new.cibn(bankrisk_code)
#
G <- as.grain(bankrisk_bn)
G
```

`as.graphNEL` *Conversion into* `graphNEL` *class*

Description

Convert an object of class `cibn` into an object of class `graphNEL`.

Usage

```
as.graphNEL(x, full=FALSE)
```

Arguments

`x`	An object of class `cibn`.
`full`	Logical value indicating whether the full DAG (i.e., augmented with latent causes and auxiliary nodes implied by the CID) should be considered. If `FALSE` (the default), the DAG before the CID is considered.

Value

An object of class `graphNEL`.

See Also

new.cibn.

Examples

```
data(bankrisk_code)
bankrisk_bn <- new.cibn(bankrisk_code)
#
G <- as.graphNEL(bankrisk_bn)
G
```

bankrisk_code *Model code for a Bayesian network*

Description

Model code for a Bayesian network to infer risk attitude of bank customers.

Usage

```
data(bankrisk_code)
```

Format

A character string.

dSepCheck *Conditional independence check*

Description

Check conditional independence between two variables based on the d-separation criterion.

Usage

```
dSepCheck(x, var1, var2, given = NULL)
```

Arguments

x	An object of class cibn.
var1	The name of the first variable.
var2	The name of the second variable.
given	A vector containing the names of conditioning variables. If NULL, marginal independence is checked.

Details

The d-separation is a necessary and sufficient condition for conditional independence. See Pearl (2000), page 16 and following).

Value

Logical

Note

The result is unchanged if arguments var1 and var2 are switched.

Dependence is a necessary but not sufficient condition for causation: see the discussion in Pearl (2000).

References

J. Pearl (2000). Causality: models, reasoning, and inference. Cambridge University Press. Cambridge, UK. ISBN: 978-0-521-89560-6

See Also

new.cibn.

Examples

```
data(bankrisk_code)
bankrisk_bn <- new.cibn(bankrisk_code)
#
dSepCheck(bankrisk_bn, var1="Age", var2="Edu")
dSepCheck(bankrisk_bn, var1="Portf", var2="Edu", given="Risk")
dSepCheck(bankrisk_bn, var1="Portf", var2="Edu", given=c("Risk","Life"))
```

---

getCPT *Functionalities for causal independence Bayesian networks*

Description

Obtain variable names, types, description fields, sample spaces, parent sets and CPTs for a causal independence Bayesian network.

Usage

```
getVariables(x)
getTypes(x)
getDescription(x)
getStates(x)
getParSets(x, full=FALSE)
getCPT(x, variables=NULL)
```

Arguments

x	An object of class cibn.
full	Only for function getParSets: logical value indicating whether the full DAG (i.e., augmented with latent causes and auxiliary nodes implied by the CID) should be considered. If FALSE (the default), the DAG before the CID is considered.
variables	Only for function getCPT: vector of character strings indicating the name of the variables for which the CPT should be computed. If NULL (the default), the CPT of all the variables in the Bayesian network will be computed.

Details

Function getVariables returns the variable names, function getTypes returns the variable types, function getDescription returns the description fields, function getStates returns the sample spaces, function getParSets returns the parent sets, and function getCPT computes one or more CPTs of interest.

See Also

new.cibn.

Examples

```
data(bankrisk_code)
bankrisk_bn <- new.cibn(bankrisk_code)
#
getVariables(bankrisk_bn)
getTypes(bankrisk_bn)
getStates(bankrisk_bn)
getDescription(bankrisk_bn)
#
getParSets(bankrisk_bn)
getParSets(bankrisk_bn, full=TRUE)  ## parent sets of the full DAG
#
getCPT(bankrisk_bn)  ## CPTs of all variables
getCPT(bankrisk_bn, variables=c("Portf","Life"))
```

`new.cibn` *Create a causal independence Bayesian network*

Description

Create a causal independence Bayesian network based on prior knowledge on the DAG and on CID parameters.

Usage

```
new.cibn(model.code = NULL, path = NULL, maximal = TRUE)
```

Arguments

`model.code`	The model code. See details below.
`path`	The path to the model code in text format. See details below.
`maximal`	Logical value indicating whether the maximal CID should be applied. Default is `TRUE`.

Details

For each variable, one command `variable` and one command `model` must be specified. Further details will be added soon, see the examples below.

The name of a variable must begin with a capital letter and cannot include special characters excepting '_'. The name of a state cannot include special characters excepting '_', and cannot begin with 'LAMBDA' or 'AUX'.

S3 methods `print`, `summary` and `plot` are available for class `cibn`.

References

A. Magrini (2021). Efficient decomposition of Bayesian networks with non-graded variables. To be appeared on *International Journal of Statistics and Probability*, 10(2).

See Also

query.cibn; sample.cibn.

Examples

```
## A simple Bayesian network to infer risk attitude of bank customers
#
#   Variables:
#    - 'Age': age in years, double-graded variable
#        with sample space: (18_30, 31_50, 51_);
#    - 'Edu': education level, double-graded variable
#        with sample space: (primary_or_less, secondary, tertiary);
#    - 'Marital': marital status, graded variable
#        with sample space: (single, convivent);
#    - 'Parent': parentship, graded variable
#        with sample space: (no, yes);
#    - 'Risk': risk attitude, double-graded variable
#        with sample space: (low, normal, high);
#    - 'Portf': type of portfolio, double-graded variable
#        with sample space: (money_market, mixed, stock_market);
#    - 'Life': life insurance, multi-valued nominal variable
#        with sample space: (long_term, short_term, none).
#
#   Edges in the DAG:
#    - 'Age' -> 'Marital'
#    - 'Age' -> 'Parent'
#    - 'Age' -> 'Risk'
#    - 'Edu' -> 'Risk'
#    - 'Marital' -> 'Risk'
#    - 'Parent' -> 'Risk'
#    - 'Risk' -> 'Portf'
#    - 'Risk' -> 'Life'
#
#   Causal interactions:
#    - between 'Marital' and 'Parent' in determining 'Risk'
#

# load model code
data(bankrisk_code)

# create the network
bankrisk_bn <- new.cibn(bankrisk_code)
bankrisk_bn <- new.cibn(bankrisk_code, maximal=FALSE)  ## disable maximal CID

# summary
summary(bankrisk_bn)
```

`plot.cibn` *Graphic for the DAG of a causal independence Bayesian network*

Description

Obtain the graphic for the DAG of a causal independence Bayesian network.

Usage

```
## S3 method for class 'cibn'
plot(x, full=FALSE, ...)
```

Arguments

x	An object of class cibn.
full	Logical value indicating whether the full DAG (i.e., augmented with latent causes and auxiliary nodes implied by the CID) should be displayed. If FALSE (the default), the DAG before the CID is displayed.
...	Further graphical parameters.

See Also

new.cibn.

Examples

```
data(bankrisk_code)
bankrisk_bn <- new.cibn(bankrisk_code)
#
plot(bankrisk_bn, attrs=list(edge=list(arrowsize=0.5)))
plot(bankrisk_bn, attrs=list(edge=list(arrowsize=0.5)), full=TRUE)  ## full DAG
```

---

query.cibn *Inference in a causal independence Bayesian network*

Description

Perform exact inference in a causal independence Bayesian network through the joint tree algorithm (interface to the gRain package)

Usage

```
query.cibn(x, target=NULL, evidence=NULL, type="marginal")
```

Arguments

x	An object of class cibn.
target	The name of the target variable. If NULL (the default), all the variables in the Bayesian network will be set as targets.
evidence	A named list with each component indicating the evidence on a specific variable in the form of a vector of state names. See the examples below.
type	A character string indicating the type of inference: "marginal", "joint" or "conditional". Default is "marginal".

Details

This function is an interface to function querygrain in the gRain package.

Value

A list with one component for each variable in the Bayesian network, indicating its probability distribution given the evidence.

References

S. L. Lauritzen and D. J. Spiegelhalter (1988). Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, Series B (Methodological)*, 50(2): 157-224. DOI: 10.1023/A:1008935617754.

See Also

new.cibn; sample.cibn.

Examples

```
data(bankrisk_code)
bankrisk_bn <- new.cibn(bankrisk_code)
#
getStates(bankrisk_bn)  ## see the sample spaces
#
query.cibn(bankrisk_bn, target="Risk", evidence=list(Age="31_50",Portf="mixed"))
query.cibn(bankrisk_bn, target="Risk", evidence=list(Age="31_50",Portf="money_market"))
query.cibn(bankrisk_bn, target="Risk", evidence=list(Age="31_50",Portf="stock_market"))
```

---

sample.cibn *Sampling from a causal independence Bayesian network*

Description

Draw a random sample from a causal independence Bayesian network

Usage

```
sample.cibn(x, nsam, seed=NULL)
```

Arguments

x	An object of class cibn.
nsam	The number of sample units.
seed	The seed for the random number generator. If NULL (the default), it is chosen randomly.

Value

An object of class data.frame.

See Also

new.cibn; query.cibn.

Examples

```
data(bankrisk_code)
bankrisk_bn <- new.cibn(bankrisk_code)
#
bankrisk_sam <- sample.cibn(bankrisk_bn, nsam=100)
head(bankrisk_sam)
summary(bankrisk_sam)
```

Index