Package ‘BNDataGenerator’

February 19, 2015

Version 1.0

Date 2014-12-28

Title Data Generator based on Bayesian Network Model

Author Jae-seong Yoo

Maintainer Jae-seong Yoo <praster1@gmail.com>

Depends R (>= 3.0.0)

Suggests bnlearn

Description Data generator based on Bayesian network model

License GPL (>= 2)

URL http://www.github.com/praster1/BN_Data_Generator

NeedsCompilation no

Repository CRAN

Date/Publication 2014-12-28 10:37:24

R topics documented:

BNDataGenerator-package ............................................. 2
big_letters ................................................................. 2
BN_Data_Generator ..................................................... 3
check_cardinality ....................................................... 4
C_M_WO_WC ................................................................. 4
fromto_to_mat ............................................................ 5
gen_asia ................................................................. 6
is_acyclic ............................................................... 7
is_DAG ................................................................. 8
make_topology ........................................................... 8
mat_to_fromto ........................................................... 9
toss_value ............................................................. 10

Index 11
BNDataGenerator-package

Data Generator based on Bayesian Network Model

Description

Data Generator based on Bayesian Network Model

Details

Package: BN_Data_Generator
Type: Package
Version: 1.0
Date: 2014-12-28
License: GPL (>=2)

Author(s)

Jae-seong Yoo <praster1@gmail.com>

References


See Also

See demo(asia) or demo(topology).

big_letters

Letters built

Description

The 26 or more lower-case letters of the Roman alphabet;

Usage

big_letters(len)
**BN_Data_Generator**

**Arguments**

  len Get or set the length of letters.

**Author(s)**

  Jae-seong Yoo <praster1@gmail.com>

**Examples**

  big_letters(26)
  big_letters(50)
  big_letters(100)
  big_letters(999)

---

**BN_Data_Generator**  
*Data Generator based on Bayesian Network Model*

**Description**

Data Generator based on Bayesian Network Model

**Usage**

  `BN_Data_Generator(arcs_mat, Probs, n, nodename=NULL, cardinality=NULL)`

**Arguments**

  arcs_mat A matrix that determines the arcs.
  Probs The conditional probabilities.
  n Data size.
  nodename The names of each nodes.
  cardinality The cardinality of each nodes.

**Author(s)**

  Jae-seong Yoo <praster1@gmail.com>

**References**


**See Also**

  See demo(asia) or demo(topology).
check_cardinality  A checker of needs of input_Prob follows cardinality

Description
A checker of needs of input_Prob follows cardinality.

Usage
check_cardinality(arcs_matL, nodename=NULLL, cardinality=NULLL)

Arguments
arcs_mat  A matrix that determines the arcs.
nodename  The names of each nodes.
cardinality  The cardinality of each nodes.

Author(s)
Jae-seong Yoo <praster1@gmail.com>

C_M_WO_WC  Correct, Missing, Wrongly Oriented, Wrongly Corrected Arcs

Description
The existence of the known network structures allows us to define three important terms which indicate the performance of the algorithm (in terms of the number of graphical errors in the learnt structure).

Usage
C_M_WO_WC(target_arcs_mat, learnt_arcs_mat)

Arguments
target_arcs_mat  A matrix of known network structure.
learnt_arcs_mat  A matrix of learnt network structure.
Value

C (Correct Arcs)
Edges present in the original network and in the learnt network structure.

M (Missing Arcs)
Edges present in the original network but not in the learnt network structure.

WO (Wrongly Oriented Arcs)
Edges present in the learnt network structure, but having opposite orientation when compared with the corresponding edge in the original network structure.

WC (Wrongly Corrected Arcs)
Edges not present in the original network but included in the learnt network structure.

Author(s)
Jae-seong Yoo <praster1@gmail.com>

References

See Also
See demo(asia) or demo(topology).

fromto_to_mat
Convert from 'fromto' to 'matrix'

Description
Convert from ‘fromto’ to ‘matrix’ that determines the arcs.

Usage
fromto_to_mat(fromto, nodename)

Arguments
fromto A matrix form structured ‘fromto’ that determines the arcs.
nodename The names of each nodes.

Author(s)
Jae-seong Yoo <praster1@gmail.com>
**Description**

Small synthetic data set from Lauritzen and Spiegelhalter (1988) about lung diseases (tuberculosis, lung cancer or bronchitis) and visits to Asia.

**Usage**

gen_asia()

**Value**

- **D** (dyspnoea) A two-level factor with levels yes and no.
- **T** (tuberculosis) A two-level factor with levels yes and no.
- **L** (lung cancer) A two-level factor with levels yes and no.
- **B** (bronchitis) A two-level factor with levels yes and no.
- **A** (visit to Asia) A two-level factor with levels yes and no.
- **S** (smoking) A two-level factor with levels yes and no.
- **X** (chest X-ray) A two-level factor with levels yes and no.
- **E** (tuberculosis versus lung cancer/bronchitis) A two-level factor with levels yes and no.

**Note**

Lauritzen and Spiegelhalter (1988) motivate this example as follows: Shortness-of-breath (dyspnoea) may be due to tuberculosis, lung cancer or bronchitis, or none of them, or more than one of them. A recent visit to Asia increases the chances of tuberculosis, while smoking is known to be a risk factor for both lung cancer and bronchitis. The results of a single chest X-ray do not discriminate between lung cancer and tuberculosis, as neither does the presence or absence of dyspnoea. Standard learning algorithms are not able to recover the true structure of the network because of the presence of a node (E) with conditional probabilities equal to both 0 and 1. Monte Carlo tests seem to behave better than their parametric counterparts.

**Author(s)**

Jae-seong Yoo <praster1@gmail.com>
is_acyclic

References


See Also

See demo(asia).

is_acyclic: Acyclic graphs

Description

This function checks for each node in a DAG whether backtracing arcs leading to it results in an "infinite recursion" error indicating that there actually is a cyclic part in the DAG (which then obviously seems not to be a DAG).

Usage

is_acyclic(arcs_mat)

Arguments

arcs_mat A matrix that determines the arcs.

Value

A list with two elements. acyclic is a boolean indicating whether the DAG is acyclic (=TRUE) or contains a cyclic component (=FALSE). nodewise is a vector containing 1 boolean per node in the DAG, TRUE indicating that backtracing from this node does not lead to a cyclic component, FALSE indicating that backtracing from this node leads to a cyclic component.

Author(s)

Jae-seong Yoo <praster1@gmail.com>

See Also

is_DAG
is_DAG

Directed acyclic graphs

Description
This function tests whether the given graph is a DAG, a directed acyclic graph.

Usage
is_DAG(arcs_mat)

Arguments
arcs_mat A matrix that determines the arcs.

Details
is_dag checks whether there is a directed cycle in the graph. If not, the graph is a DAG.

Value
A logical vector of length one.

Author(s)
Jae-seong Yoo <praster1@gmail.com>

See Also
is_acyclic

make_topo

Bayesian Networks with varying topologies

Description
Bayesian Networks with varying topologies (DAGs) with number of nodes.

Usage
make_topo(nodes, topology, Probs, nodename=NULL, cardinality=NULL)
**Arguments**

- **nodes**: The number of nodes.
- **probs**: The conditional probabilities.
- **nodename**: The names of each node.
- **cardinality**: The cardinality of each node.

**Details**

The volume of the manifold is a geometric characteristic associated with the topology of Bayesian network. Each BN produces a different magnitude of the volume based on the DAG of Bayesian network. Collapse, Line, Star, PseudoLoop, Diamond, Rhombus.

**Author(s)**

Jae-seong Yoo <praster1@gmail.com>

**References**


**See Also**

See demo(topology).

---

**Description**

Convert from 'matrix' to 'fromto' that determines the arcs.

**Usage**

```r
mat_to_fromto(arcs_mat)
```

**Arguments**

- **arcs_mat**: A matrix that determines the arcs that determines the arcs.

**Author(s)**

Jae-seong Yoo <praster1@gmail.com>
toss_value  

Tossing a Cardinality

Description
Sets up a sample space for the experiment of tossing a cardinality repeatedly with the outcomes "Values".

Usage
toss_value(timesL num_of_casesL makespace=false)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>times</td>
<td>Number of tosses.</td>
</tr>
<tr>
<td>num_of_cases</td>
<td>Cardinality.</td>
</tr>
<tr>
<td>makespace</td>
<td>Logical.</td>
</tr>
</tbody>
</table>

Value
A data frame, with an equally likely probs column if makespace is TRUE.

Note
It developed of 'tosscoin' function in prob package.

Author(s)
Jae-seong Yoo <praster1@gmail.com>

Examples
toss_value(1, 3)
toss_value(2, 3)
toss_value(3, 4, makespace = TRUE)
Index

*Topic **Bayesian Network**
  BN_Data_Generator, 3
  BNDatagenator-package, 2
*Topic **letters**
  big_letters, 2

big_letters, 2
BN_Data_Generator, 3
BNDatagenator
  (BNDatagenator-package), 2
BNDatagenator-package, 2
C_M_WO_WC, 4
check_cardinality, 4

fromto_to_mat, 5
gen_asia, 6

is_acyclic, 7, 8
is_DAG, 7, 8

make_topology, 8
mat_to_fromto, 9
toss_value, 10