Package ‘BNSL’

October 12, 2022

Type Package
Title Bayesian Network Structure Learning
Version 0.1.4
Date 2019-1-13
Author Joe Suzuki and Jun Kawahara
Maintainer Joe Suzuki <j-suzuki@sigmath.es.osaka-u.ac.jp>
Depends bnlearn, igraph
Description From a given data frame, this package learns its Bayesian network structure based on a selected score.
License GPL (>= 2)
Imports Rcpp (>= 0.12.0)
LinkingTo Rcpp
NeedsCompilation yes
Repository CRAN
Date/Publication 2019-01-24 10:00:03 UTC

R topics documented:

   BNSL-package ..........................................................  2
   bnsl .................................................................  2
   bnsl_p ...............................................................  4
   cmi .................................................................  5
   FFtable ...............................................................  6
   kruskal ..............................................................  7
   mi .................................................................  8
   mi_matrix ..........................................................  9
   parent.set ........................................................ 10

Index 12
Description

From a given dataframe, this package learns a Bayesian network structure based on a selected score.

Details

Currently, this package estimates mutual information and conditional mutual information, and combines them to construct either a Bayesian network or an undirected forest, any undirected forest can be a Bayesian network by adding appropriate directions.

Author(s)

Joe Suzuki and Jun Kawahara
Maintainer: Joe Suzuki <j-suzuki@sigmath.es.osaka-u.ac.jp>

References

Usage

bnsl(df, tw = 0, proc = 1, s=0, n=0, ss=1)

Arguments

df  a dataframe.
tw  the upper limit of the parent set.
proc  the criterion based on which the BNSL solution is sought. proc=1,2, and 3 indicates that the structure learning is based on Jeffreys [1], MDL [2,3], and BDeu [3]
s  The value computed when obtaining the bound.
n  The number of samples.
ss  The BDeu parameter.

Value

The Bayesian network structure in the bn class of bnlearn.

Author(s)

Joe Suzuki and Jun Kawahara

References


See Also

parent

Examples

library(bnlearn)
bns1(asia)
**bnsl_p**

*Bayesian Network Structure Learning*

---

**Description**

The function outputs the Bayesian network structure given a dataset based on an assumed criterion.

**Usage**

```
bnsl_p(df, psl, tw = 0, proc = 1, s = 0, n = 0, ss = 1)
```

**Arguments**

- `df`: a dataframe.
- `psl`: the list of parent sets.
- `tw`: the upper limit of the parent set.
- `proc`: the criterion based on which the BNSL solution is sought. `proc=1,2, and 3` indicates that the structure learning is based on Jeffreys [1], MDL [2,3], and BDeu [3]
- `s`: The value computed when obtaining the bound.
- `n`: The number of samples.
- `ss`: The BDeu parameter.

**Value**

The Bayesian network structure in the bn class of bnlearn.

**Author(s)**

Joe Suzuki and Jun Kawahara

**References**


**See Also**

parent
Examples

```r
library(bnlearn)
p0 <- parent.set(lizards, 0)
p1 <- parent.set(lizards, 1)
p2 <- parent.set(lizards, 2)
bnsl_p(lizards, list(p0, p1, p2))
```

### Description

A standard estimator of conditional mutual information calculates the maximal likelihood value. However, the estimator takes positive values even the pair follows a distribution of two independent variables. On the other hand, the estimator in this package detects conditional independence as well as consistently estimates the true conditional mutual information value as the length grows based on Jeffrey's prior, Bayesian Dirichlet equivalent uniform (BDeu [1]), and the MDL principle. It also estimates the conditional mutual information value even when one of the pair is continuous (see [2]).

### Usage

```r
cmi(x, y, z, proc=0L)
```

### Arguments

- `x`: a numeric vector.
- `y`: a numeric vector.
- `z`: a numeric vector. `x`, `y` and `z` should have an equal length.
- `proc`: the estimation is based on Jeffrey's prior, the MDL principle, and BDeu for `proc=0,1,2`, respectively. If the argument `proc` is missing, `proc=0 (Jeffreys')` is assumed.

### Value

the estimation of conditional mutual information between the two numeric vectors based on the selected criterion, where the natural logarithm base is assumed.

### Author(s)

Joe Suzuki and Jun Kawahara

### References

See Also

cmi

Examples

n=100

x=c(rbinom(n,1,0.2), rbinom(n,1,0.8))
y=c(rbinom(n,1,0.8), rbinom(n,1,0.2))
z=c(rep(1,n),rep(0,n))
cmi(x,y,z,proc=0); cmi(x,y,z,1); cmi(x,y,z,2)

x=c(rbinom(n,1,0.2), rbinom(n,1,0.8))
u=rbinom(2*n,1,0.1)
y=(x+u)
z=c(rep(1,n),rep(0,n))
cmi(x,y,z); cmi(x,y,z,proc=1); cmi(x,y,z,2)

FFtable

A faster version of fftable

Description

The same procedure as fftable prepared by the R language. The program is written using Rcpp.

Usage

FFtable(df)

Arguments

df a dataframe.

Value

a frequency table of the last column based on the states that are determined by the other columns.

Author(s)

Joe Suzuki and Jun Kawahara

See Also

fftable

Examples

library(bnlearn)
FFtable(asia)
kruskal

| kruskal | Given a weight matrix, generate its maximum weight forest |

Description

The function lists the edges of an forest generated by Kruskal’s algorithm given its weight matrix in which each weight should be symmetric but may be negative. The forest is a spanning tree if the elements of the matrix take positive values.

Usage

```r
kruskal(W)
```

Arguments

- `W` a matrix.

Value

A matrix object of size n x 2 for matrix size n x n in which each row expresses an edge when the vertexes are expressed by 1 through n.

Author(s)

Joe Suzuki and Jun Kawahara

References


Examples

```r
library(igraph)
library(bnlearn)
df=asia
mi.mat=mi_matrix(df)
edge.list=kruskal(mi.mat)
edge.list
g=graph_from_edgelist(edge.list, directed=FALSE)
V(g)$label=colnames(df)
plot(g)
```
mi

Bayesian Estimation of Mutual Information

Description

A standard estimator of mutual information calculates the maximal likelihood value. However, the estimator takes positive values even if the pair follows a distribution of two independent variables. On the other hand, the estimator in this package detects independence as well as consistently estimates the true mutual information value as the length grows based on Jeffreys’s prior, Bayesian Dirichlet equivalent uniform (BDeu [1]), and the MDL principle. It also estimates the mutual information value even when one of the pair is continuous (see [2]).

Usage

mi(x, y, proc=0)

Arguments

x a numeric vector.
y a numeric vector. x and y should have a equal length.
proc the estimation is based on Jeffreys’s prior, the MDL principle, and BDeu for proc=0,1,2, respectively. If one of the two is continuous, proc=10 should be chosen. If the argument proc is missing, proc=0 (Jeffreys) is assumed.

Value

the estimation of mutual information between the two numeric vectors based on the selected criterion, where the natural logarithm base is assumed.

Author(s)

Joe Suzuki and Jun Kawahara

References


See Also

cmi
mi_matrix

Examples

n=100

x=rbinom(n,1,.5); y=rbinom(n,1,.5); mi(x,y)

z=rbinom(n,1,.1); y=(x+z)

mi(x,y); mi(x,y,proc=1); mi(x,y,2)

x=rnorm(n); y=rnorm(n); mi(x,y,proc=10)

x=rnorm(n); z=rnorm(n); y=0.9*x+sqrt(1-0.9^2)*z; mi(x,y,proc=10)

mi_matrix

Generating its Mutual Information Estimations Matrix

Description

The estimators in this package detect independence as well as consistently estimates the true conditional mutual information value as the length grows based on Jeffrey’s prior, Bayesian Dirichlet equivalent uniform (BDeu [1]), and the MDL principle. It also estimates the conditional mutual information value even when one of the pair is continuous (see [2]). Given a data frame each column of which may be either discrete or continuous, this function generates its mutual information estimation matrix.

Usage

mi_matrix(df, proc=0)

Arguments

df a data frame.

proc given two discrete vectors of equal length, the function estimates the mutual information based on Jeffrey’s prior, the MDL principle, and BDeu for proc=0,1,2, respectively. If one of the columns is continuous, proc=10 should be chosen. If the argument proc is missing, proc=0 (Jeffreys’) is assumed.

Value

the estimation of mutual information between the two numeric vectors based on the selected criterion, where the natural logarithm base is assumed.

Author(s)

Joe Suzuki and Jun Kawahara
References


See Also

mi

Examples

library(bnlearn)
mi_matrix(asia)
mi_matrix(asia,proc=1)
mi_matrix(asia,proc=2)
mi_matrix(asia,proc=3)

parent.set Parent Set

Description

This function estimates a parent set of h in each subset w as follows: Suppose we are given a subset w of the p-1 variables excluding h, where p is the number of columns in df. Then, a score is defined for each subset w, where the score expresses how well the subset is likely to be the true parent set of h in w. Currently, a Bayesian score (Jeffreys’ prior) is applied. This function computes the maximum score z and its subset y of w. This function computes y and z for all w, where w and y are expressed by binary sequences of length p, respectively. When the computation is heavy, it can be reduced by specifying the maximum size of w. If tw is zero (default), the tw value is set to p-1. Otherwise, the tw value expresses the maximum size.

Usage

parent.set(df, h, tw=0, proc=1)

Arguments

df a data frame.
h an integer from 0 to p-1, where p is the number of columns in df.
tw an integer from 0 to p-1, where p is the number of columns in df.
proc the parent sets are estimated based on Jeffreys’ (proc=0,1) [1], MDL (proc=2) [2,3], and BDeu (proc=3) [4].
Value

the data frame in which each row consists of the triples \((w, y, z)\): \(w\) is a subset of the \(p-1\) variables excluding \(h\); \(y\) is the parent set for \(w\); and \(z\) is the score of the parent set.

Author(s)

Joe Suzuki and Jun Kawahara

References


See Also

cmi

Examples

library(bnlearn)
df=asia
parent.set(df,7)
parent.set(df,7,1)
parent.set(df,7,2)
Index

* package
  BNSL-package, 2

BNSL (BNSL-package), 2
bnsL, 2
BNSL-package, 2
bnsL_p, 4

cmi, 5

FFtable, 6

kruskal, 7

mi, 8
mi_matrix, 9

parent.set, 10