Package: SELF (via r-universe)

August 22, 2024

Type Package

Title A Structural Equation Embedded Likelihood Framework for Causal Discovery

Version 0.1.1

Author Ruichu Cai [ths, aut], Jie Qiao [aut, cre], Zhenjie Zhang [ths, aut], Zhifeng Hao [ths, aut]

Maintainer Jie Qiao <qiaojie2004@vip.qq.com>

Description Provides the SELF criteria to learn causal structure. Please cite ``Ruichu Cai, Jie Qiao, Zhenjie Zhang, Zhifeng Hao. SELF: Structural Equational Embedded Likelihood Framework for Causal Discovery. AAAI. 2018."

License GPL-2

LazyData true

Imports data.table (>= 1.10.4), xgboost (>= 0.6-4), Rcpp (>= 0.12.10), CompareCausalNetworks (>= 0.1.0), bnlearn (>= 4.1.1)

LinkingTo Rcpp

RoxygenNote 6.0.1

Encoding UTF-8

NeedsCompilation yes

Repository CRAN

Date/Publication 2017-11-22 13:24:22 UTC

Contents

`SELF-package` *SELF: A Structural Equation Embedded Likelihood Framework for Causal Discovery*

Description

Provides the SELF criteria to learn causal structure. Please cite "Ruichu Cai, Jie Qiao, Zhenjie Zhang, Zhifeng Hao. SELF: Structural Equational Embedded Likelihood Framework for Causal Discovery. AAAI. 2018."

Author(s)

Maintainer: Jie Qiao <qiaojie2004@vip.qq.com>

Authors:

- Ruichu Cai <cairuichu@gmail.com> [thesis advisor]
- Zhenjie Zhang <zhenjie@adsc.com.sg> [thesis advisor]
- Zhifeng Hao <zfhao@gdut.edu.cn> [thesis advisor]

`fhc` *Fast Hill-Climbing*

Description

The function for the causal structure learning.

Usage

```
fhc(D, G = NULL, min_increase = 0.01, score_type = "bic", file = "",
  verbose = TRUE, save_model = FALSE, bw = "nrd0", booster = "gbtree",
  gamma = 10, nrounds = 30, ...)
```

Arguments

`D`	Input Data.
`G`	An initial graph for hill climbing. Default: empty graph.
`min_increase`	Minimum score increase for faster convergence.
`score_type`	You can choose "bic","log","aic" score to learn the causal struture. Default: bic
`file`	Specifies the output folder and its path to save the model at each iteration.
`verbose`	Show the progress bar for each iteration.
`save_model`	Save the meta data during the iteration so that you can easily restore progress and evaluate the model during iteration.

bw	the smoothing bandwidth which is the parameter of the function stats::density(Kernel stats::density Estimation)
booster	Choose the regression method, it could be "lm", "gbtree" and "gblinear". The "lm" and "gblinear" is the linear regression methods and "gbtree" is the nonlinear regression method. Default: gbtree
gamma	The parameter in xgboost: minimum loss reduction required to make a further partition on a leaf node of the tree. the larger, the more conservative the algorithm will be.
nrounds	the maximum number of trees for xgboost.Default:30.
...	other parameters for xgboost.see also: help(xgboost)

Value

The adjacency matrix of the casual structure.

Examples

```
## Not run:
#x->y->z
set.seed(0)
x=rnorm(4000)
y=x^2+runif(4000,-1,1)*0.1
z=y^2+runif(4000,-1,1)*0.1
data=data.frame(x,y,z)
fhc(data,gamma=10,booster = "gbtree")

#x->y->z linear data
set.seed(0)
x=rnorm(4000)
y=3*x+runif(4000,-1,1)*0.1
z=3*y+runif(4000,-1,1)*0.1
data=data.frame(x,y,z)
fhc(data,booster = "lm")

#randomGraph with linear data

set.seed(0)
G=randomGraph(dim=10,indegree=1.5)
data=synthetic_data_linear(G=G,sample_num=4000)
fitG=fhc(data,booster = "lm")
indicators(fitG,G)

## End(Not run)
```

indicators — *Calculate the f1,precision,recall score of the graph*

Description

Calculate the f1,precision,recall score of the graph

Usage

```
indicators(pred, real)
```

Arguments

pred	Predicted graph
real	Real graph

Value

f1,precision,recall score.

Examples

```
pred<-matrix(c(0,0,0,0,1,0,1,1,0),nrow=3,ncol=3)
real<-matrix(c(0,0,0,0,1,0,1,0,0),nrow=3,ncol=3)
indicators(pred,real)
```

mmpcAnm — *mmpc algorithm with additive noise model*

Description

The nonlinear data comparison algorithm. We use the mmpc algorithm to learn a causal skeleton and use ANM to recognize the direction

Usage

```
mmpcAnm(data)
```

Arguments

data	The data

randomGraph *Generate a random graph*

Description

Generate a random graph based on the given dimension size and average indegree

Usage

```
randomGraph(dim, indegree, maxite = 10000)
```

Arguments

dim	The dimension of the random graph
indegree	The average indegree of random graph for each nodes
maxite	The maximum iterations to find the random graph

Value

Return a random graph

Examples

```
randomGraph(dim=10,indegree=1)
```

synthetic_data_linear *synthetic linear data base on the graph*

Description

Synthetic linear data base on the graph. The noises are sampled from the super-gaussian distribution. The coefficients are sample from U(-1,-0.5),U(0.5,1)

Usage

```
synthetic_data_linear(G, sample_num, ratio = 1, return_noise = FALSE)
```

Arguments

G	An adjacency matrix.
sample_num	The number of samples
ratio	The noise ratio It will grow or shrink the value of the noise
return_noise	Whether return the noise of each nodes for further analysis.

Value

Return a synthetic data

Examples

```
G<-matrix(c(0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0),nrow = 4,ncol = 4)
data=synthetic_data_linear(G,100)
```

---

synthetic_data_nonlinear

synthetic nonlinear data base on the graph

Description

synthetic nonlinear data base on the graph. The data generation mechanism is y=scale(a1b1x^2+a2b2x^3+a3b3x^4+a4b4sin(

Usage

```
synthetic_data_nonlinear(G, sample_num, ratio = 1, return_noise = FALSE)
```

Arguments

`G`	An adjacency matrix.
`sample_num`	The number of samples
`ratio`	The noise ratio. It will grow or shrink the value of the noise.
`return_noise`	Whether return the noise of each nodes for further analysis.

Value

Return a synthetic data

Examples

```
G<-matrix(c(0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0),nrow = 4,ncol = 4)
data=synthetic_data_nonlinear(G,100)
```

Index