Package ‘mDAG’

August 20, 2019

Type Package
Title Inferring Causal Network from Mixed Observational Data Using a Directed Acyclic Graph
Version 1.2.2
Date 2019-8-20
Author Wujuan Zhong, Li Dong, Quefeng Li, Xiaojing Zheng
Maintainer Wujuan Zhong <zhongwujuan@gmail.com>
Description Learning a mixed directed acyclic graph based on both continuous and categorical data.
LazyData true
Encoding UTF-8
License GPL (>= 2)
Imports Rcpp (>= 0.12.14), pcalg, mgm, bnlearn, methods, nnet
Depends R (>= 2.10), logistf
LinkingTo Rcpp, RcppArmadillo
RoxygenNote 6.1.1
NeedsCompilation yes
Repository CRAN
Date/Publication 2019-08-20 15:20:03 UTC

R topics documented:

  example_data ................................................................. 2
  mDAG ................................................................. 2

Index 5
example_data  

Example data

Description

An example data, including 5 variables (4 continuous variables and 1 binary variable) and 100 samples.

Usage

data(example_data)

mDAG  

Inferring Causal Network from Mixed Observational Data Using a Directed Acyclic Graph

Description

This function learns a mixed directed acyclic graph based on both continuous and categorical data.

Usage

mDAG(data, type, level, SNP = rep(0, ncol(data)), lambdaGam = 0.25, 
    ruleReg = "OR", threshold = "LW", weights = rep(1, nrow(data)), 
    alpha = 0.05, nperm = 10000)

Arguments

data  
A n x p matrix. Each row is a sample; each column is a variable.

type  
A string vector of length p, indicating the type of variable for each column in data. 'g' for Gaussian, 'c' for categorical.

level  
A vector of length p, indicating the number of categories of each variable. For continuous variables, set it to 1.

SNP  
A vector of length p, indicating which variable is a SNP.

lambdaGam  
Hyperparameter $\gamma$ in the EBIC if lambdaSel = 'EBIC'. Defaults is lambdaGam = 0.25.

ruleReg  
Default is 'OR'. Rule used to combine two estimates from nodewise regression (one from regressing A on B and the other from B on A). ruleReg = 'AND' requires both estimates to be nonzero in order to set the edge to be present. ruleReg = 'OR' requires at least one estimate to be nonzero in order to set the edge to be present.
threshold Default is ‘LW’. A threshold below which the combined estimates from node-wise regression are put to zero. threshold = ‘LW’ refers to the threshold in Loh and Wainwright (2012). threshold = ‘HW’ refers to the threshold in Haslbeck and Waldorp (2016). If threshold = ‘none’ no thresholding is applied.

weights A vector of length n, indicating weights for observations.

alpha Significance level for permutation test of conditional independence. Default is 0.05.

nperm The number of permutations in the permutation test of conditional independence. Default is 10000.

Value

A list of the following components:

- arcs: A two-column matrix, indicating arcs of the DAG.
- nodes: A list. Each element is named after a node and contains the following elements.
  - nbr: a string vector indicating the neighbourhood of the node.
  - parents: a string vector indicating the parents of the node.
  - children: a string vector indicating the children of the node.
- skeleton: A p x p adjacency matrix. If there is an edge from node i to node j, its (i, j) th entry = 1; otherwise = 0.

Author(s)

Wujuan Zhong, Li Dong, Quefeng Li, Xiaojing Zheng

References


Examples

```r
# load package
library(mDAG)
type=c("g","g","g","g","c")
level=c(1,1,1,1,2)
# To save time for running example, we set nperm as 150.
# Use default nperm=10000 to generate a more reliable DAG for your own data.
dag=mDAG(data=example_data, type=type, level=level, nperm=150)
print(dag$skeleton)
# draw the DAG
# library(bnlearn)
# bnlearn:::graphviz.backend(nodes=names(dag$nodes),arcs=dag$arcs,shape="rectangle")
```
Index

example_data, 2

mDAG, 2