

Package ‘clrdag’

March 31, 2019

Type Package

Title Likelihood ratio tests of a large directed acyclic graph

Version 0.19.03

Date 2019-03-30

Author Chunlin Li, Xiaotong Shen, and Wei Pan

Maintainer Chunlin Li <li000007@umn.edu>

Depends R (>= 3.5.3)

Imports Rcpp (>= 1.0.1)

LinkingTo Rcpp, RcppArmadillo

Suggests mvtnorm

Description The 'clrdag' package provides R functions for constrained likelihood ratio tests of a large directed acyclic graph. Documentation about 'clrdag' is provided by the vignette included in this package and via the paper by Li, Shen, and Pan (2019).

License GPL (>= 2)

BugReports <https://github.umn.edu/li000007/clrdag/issues>

NeedsCompilation yes

R topics documented:

clrdag	1
Index	4

clrdag	<i>cmleDAG</i>
--------	----------------

Description

A function computes the MLE/likelihood ratio of a Gaussian directed acyclic graph with specified constraints.

Usage

```
cmleDAG(X, A = NULL, Lambda = NULL, D = NULL, tau, mu, rho, tol_abs = 1e-04, tol_rel = 1e-04, dc_max_iter
```

Arguments

X	An n by p data matrix, where n is the number of observations and p is the dimension.
A, Lambda	Initial estimate. A is a p by p adjacency matrix, Lambda is a p by p dual matrix in acyclicity condition. A must be a DAG! If A is NULL (default), the initial estimate is provided automatically (Be careful!).
D	A p by p matrix indicating hypothesized edges. For the entries equal to 1, no sparse penalty is imposed.
tau	A positive real number. tau is the threshold parameter in TLP.
mu	A positive real number. mu is the sparsity parameter.
rho	A positive real number. rho is the ADMM dual parameter.
tol_abs, tol_rel	Positive real. The absolute and relative tolerance.
dc_max_iter, admm_max_iter	Positive integer. The maximum iteration number of DC and ADMM.
test_path	Logical. If TRUE, the path test is used.
trace_obj	Logical. If TRUE, the objective values are printed after each iteration.

Author(s)

Chunlin Li

References

Li, C., Shen, X., and Pan, W. (2019). Likelihood ratio tests of a large directed acyclic graph. Submitted.

Examples

```
library(mvtnorm)
##
## Example 1: random graph
##
set.seed(2019)
p<-50
n<-1000
## random graph: randomly generate adjacency matrix A, A lower triangular
sparsity <- 2/p
A <- matrix(rbinom(p*p,1,sparsity)*sign(runif(p*p,min=-1,max=1)),p,p)
A[upper.tri(A, diag = TRUE)] <- 0
Sigma <- solve(diag(p) - A)
Sigma <- Sigma %*% t(Sigma)
X <- rmvnorm(n,mean=rep(0,p), sigma=Sigma, method="chol")
out <- cmleDAG(X=X,tau=0.3,mu=1,rho=1.2,trace_obj=FALSE) # compute the MLE
```

```
B <- out$A
B <- ifelse(abs(B)>0.3,1,0)
B == abs(A)
##
## Example 2: hub graph
##
set.seed(2019)
p<-50
n<-1000
## hub graph: randomly generate adjacency matrix A, A lower triangular
A <- matrix(0,p,p)
A[,1] <- sign(runif(p,min=-1,max=1))
A[1,1] <- 0
Sigma <- solve(diag(p) - A)
Sigma <- Sigma %*% t(Sigma)
X <- rmvnorm(n,mean=rep(0,p), sigma=Sigma, method="chol")
out <- cmleDAG(X=X,tau=0.3,mu=1,rho=1.2,trace_obj=FALSE) # compute the MLE
B <- out$A
B <- ifelse(abs(B)>0.3,1,0)
B == abs(A)
```

Index

*Topic **Directed acyclic graph**

clrdag, [1](#)

*Topic **Likelihood ratio test**

clrdag, [1](#)

*Topic **MLE**

clrdag, [1](#)

clrdag, [1](#)

cmleDAG (clrdag), [1](#)