Package ‘scorelingam’

December 5, 2022

Type Package

Title What the package does (short line)

Version 1.0

Date 2022-12-04

Author Gabriel Ruiz

Maintainer Gabriel Ruiz

Description Implements the sorting procedure outlined in Ruiz et. al (2022+).

URL `https://github.com/gabriel-ruiz/scorelingam`, `https://openreview.net/forum?id=4pCjIGIjrt`

BugReports `https://github.com/gabriel-ruiz/scorelingam/issues`

Imports Rcpp (>= 1.0.9)

LinkingTo Rcpp, RcppArmadillo

RoxygenNote 7.2.0

R topics documented:

`scorelingam-package` *ScoreLiNGAM*

Description

Implements the sorting procedure outlined in Ruiz et. al (2022+).

Details

This section should provide a more detailed overview of how to use the package, including the most important functions.

Author(s)

Gabriel Ruiz

References

G Ruiz, OH Madrid-Padilla, Q Zhou. "Sequentially learning the topological ordering of directed acyclic graphs with likelihood ratio scores." Transactions of Machine Learning Research, 2022+. [Link]

Examples

```
See the examples at https://github.com/gabriel-ruiz/scorelingam.
```

`checkSortingErrors` *Sorting Error.*

Description

Sorting Error.

Usage

```
checkSortingErrors(estOrder, A)
```

Arguments

`A:`	The true weighted adjacency matrix.

Value

Returns proportion of ancestors sorted after descendants if argument M is a DAG adjacency matrix, then returns proportion of parents sorted after children.

corrMat *Quickly estimate a Pearson correlation matrix using RcppArmadillo.*

Description

Quickly estimate a Pearson correlation matrix using RcppArmadillo.

Usage

```
corrMat(X)
```

Arguments

`X`	an n by p data matrix which is used to estimate the correlation matrix.

Value

a p by p estimate of the correlation matrix.

dlap *laplace density of x*

Description

laplace density of x

Usage

```
dlap(x, center = mean(x), shape = mean(abs(x - center)), log = F)
```

generateData *Generate Linear (laplace, logistic, or scaled-t error family) Structural Causal Model Data*

Description

Generate Linear (laplace, logistic, or scaled-t error family) Structural Causal Model Data

Usage

```
generateData(
  B,
  n = 1000,
  family = "laplace",
  scale = rep(1, ncol(B)),
  perm = topoSort(as.matrix(B)),
  df = 5
)
```

Arguments

`B`	a p by p weighted adjacency matrix for an acyclic linear SEM.
`n`	the desired sample size. Default: n=1000.
`family`	Currently allow 't', 'laplace' (default), or 'logistic'.
`scale`	length p vector s.t. $\mathrm{var}[\epsilon_j]$ =scale[j]$\times\mathrm{var}[\epsilon_0]$ with $\epsilon_0 \sim \mathrm{family}(1)$ (the chosen family with scale parameter 1). Default: scale=rep(1,ncol(B)).
`df`	degrees of freedom for scaled-t-distributed noise. Default: df=5.

Value

n by p matrix object.

---

getParents — *Get parent sets given Markov blanket and topological ordering*

Description

Get parent sets given Markov blanket and topological ordering

Usage

```
getParents(mb, ordering)
```

Arguments

`mb`	a length p list object whose j-th entry gives the Markov blanket of node j. The j-th entry can also be some other set that defines the possible support for the parent set.
`ordering`	a length p vector which gives the specified topological ordering of the nodes in the underlying DAG.

Value

a list object of length p; entry j denotes the parent set of node j in the underlying DAG.

`getWeights` *Get weighted adjancency matrix for linear SEM. Assumes that X is zero-centered, which can be done via X=scale(Xoriginal,center=T,scale=F). Estimation is done via Ordinary Least Squares (OLS) regression.*

Description

Get weighted adjancency matrix for linear SEM. Assumes that X is zero-centered, which can be done via X=scale(Xoriginal,center=T,scale=F). Estimation is done via Ordinary Least Squares (OLS) regression.

Usage

```
getWeights(X, pa)
```

Arguments

`X`	an n by p data matrix which is used to estimate the weighted adjacency matrix.
`pa`	a length p list object whose j-th entry gives the support for node j's parent set.

Value

A p by p weighted adjacency matrix for the linear SEM.

`moralize` *Moralize a DAG. Obtain Markov Blanket, the union of all children, parents, co-parents.*

Description

Moralize a DAG. Obtain Markov Blanket, the union of all children, parents, co-parents.

Usage

```
moralize(B)
```

Arguments

`B`	the p by p weighted adjacency matrix. Assumes B[j,k]!=0 if $j \in PA_k$.

Value

a list of length p. Each entry j corresponds to the Markov Blanket of node j according to adjacency matrix B.

randomWeightedAdjacencyMatrix

Generate a random weighted adjacency matrix with num.roots number of root nodes. Each non-root node will have between pa.min and pa.max number of parents. The parent weight (in absolute value) is between pa.wt.min and pa.wt.max. The parent weight is positive with probability prob.pos (else, negative).

Description

Generate a random weighted adjacency matrix with num.roots number of root nodes. Each non-root node will have between pa.min and pa.max number of parents. The parent weight (in absolute value) is between pa.wt.min and pa.wt.max. The parent weight is positive with probability prob.pos (else, negative).

Usage

```
randomWeightedAdjacencyMatrix(
  p,
  num.roots = p - 1,
  pa.min = 1,
  pa.max = 1,
  pa.wt.min = 1,
  pa.wt.max = 1,
  prob.pos = 0.5,
  perm = sample(x = p, size = p, replace = F)
)
```

Arguments

`p`	number of nodes in underlying DAG.
`num.roots`	number of root nodes in underlying DAG. Default: num.roots=p-1.
`pa.min`	minimum number of parents a node can have. Default: pa.min=1.
`pa.max`	maximum number of parents a node can have. Default: pa.max=1.
`pa.wt.min`	the minimum (in absolute value) coefficient value for the parent to a child in the linear SEM. Default: pa.wt.min=1.
`pa.wt.max`	the maximum (in absolute value) coefficient value for the parent to a child in the linear SEM. Default: pa.wt.max=1.
`prob.pos`	the probability of a positive coefficient value for the parent to a child in the the linear SEM. Default: prob.pos=0.5.
`perm`	a topological ordering for underlying DAG. Default is random.

Value

a list with two elements: perm (the topological ordering used) and B (the weighted adjacency matrix).

rlap *laplace r.v.*

Description

laplace r.v.

Usage

```
rlap(n = 1, mu = 0, b = 1)
```

scorelingam *The ScoreLiNGAM sorting procedure.*

Description

The ScoreLiNGAM sorting procedure.

Usage

```
scorelingam(X, mb, numUpdates = 5L, family = "laplace", df = 10)
```

Arguments

`X`	an n by p data matrix which is used to estimate the permutation.
`mb`	a length p list object whose j-th entry gives the Markov blanket of node j (or its superset).
`numUpdates`	the number of updates to give while sorting. Default: numUpdates=5.
`family`	Currently allows 't', 'laplace' (default), or 'logistic'.
`df`	degrees of freedom for scaled-t-distributed noise. Default: df=10.

Value

A length p vector with the estimated permutation.

`skeleton`	*Skeleton for an undirected graph based on Markov Blanket (or neighborhood estimates) of each node.*

Description

Skeleton for an undirected graph based on Markov Blanket (or neighborhood estimates) of each node.

Usage

```
skeleton(mb)
```

Arguments

`mb`	the Markov blanket or neighborhood estiamtes. A list of length p.

Value

a p by p adjacency matrix for an undirected graph. 0 indicates no undirected edge in the graph, while 1 indicates an directed edge in the graph.

Index