Package: Bestie (via r-universe)

September 8, 2024

Type Package

Title Bayesian Estimation of Intervention Effects

Version 0.1.5

Author Jack Kuipers [aut,cre] and Giusi Moffa [aut]

Maintainer Jack Kuipers <`jack.kuipers@bsse.ethz.ch`>

Description An implementation of intervention effect estimation for DAGs (directed acyclic graphs) learned from binary or continuous data. First, parameters are estimated or sampled for the DAG and then interventions on each node (variable) are propagated through the network (do-calculus). Both exact computation (for continuous data or for binary data up to around 20 variables) and Monte Carlo schemes (for larger binary networks) are implemented.

License GPL-3

Imports BiDAG (>= 2.0.0), Rcpp (>= 1.0.3), mvtnorm (>= 1.1.0)

LinkingTo Rcpp

Encoding UTF-8

RoxygenNote 7.1.2

NeedsCompilation yes

Date/Publication 2022-04-28 13:10:02 UTC

Repository https://jackkuipers.r-universe.dev

RemoteUrl https://github.com/cran/Bestie

RemoteRef HEAD

RemoteSha 1c4f01c0ed501ae77aec4051cd031c7fe0d42ef2

Contents

`DAGintervention` *Exact estimation of intervention effects for a single DAG or a chain of sampled DAGs*

Description

`DAGintervention` takes a DAG or a sampled chain of DAGs (for example from the `partitionMCMC` function of the BiDAG package) and computes the intervention effect of each node on all others. For binary data, this is performed by exhaustively examining all possible binary states. This is exponentially complex in the number of variables which should therefore be limited to around 20 or fewer. For more variables there is a Monte Carlo version `DAGinterventionMC` instead. For continuous data, the intervention estimation is performed by extracting the edge coefficients from their posterior distribution and using matrix inversion following arXiv:2010.00684. User-defined scores are also supported as long as the DAG parameters are analogous to the BDe/BGe cases, see `DAGparameters`.

Usage

```
DAGintervention(incidences, dataParams, sample = TRUE, unrollDBN = TRUE)
```

Arguments

`incidences`	a single adjacency matrix of a list of adjacency matrices of sampled DAGs, with entry [i,j] equal to 1 when a directed edge exists from node i to node j
`dataParams`	the data and parameters used to learn the DAGs derived from the `scoreparameters` function of the BiDAG package
`sample`	logical indicating whether to sample the parameters of each node from the posterior (TRUE, default) or to take the expectation (FALSE)
`unrollDBN`	logical indicating whether to unroll a DBN to a full DAG over all time slices (TRUE, default) or to use the compact representation (FALSE)

Value

a single matrix or a list of matrices containing the full set of intervention effects for each input DAG. Entry [i,j] is the downstream effect on node j of intervening on node i (the difference observed at node j when setting node i to 1 and 0)

See Also

`scoreparameters`

Examples

```
scoreParam <- BiDAG::scoreparameters("bde", BiDAG::Asia)
causalmat <- DAGintervention(BiDAG::Asiamat, scoreParam)
```

`DAGinterventionMC` *Monte Carlo estimation of intervention effects for a DAG or chain of sampled DAGs*

Description

`DAGinterventionMC` takes a DAG or a sampled chain of DAGs (for example from the `partitionMCMC` function of the BiDAG package) and computes, for binary data, a Monte Carlo estimate of the intervention effect of each node on all others by simulating data from the DAG. By default each node is intervened upon and the downstream effects estimated by further sampling. A faster but less robust and accurate version is also offered which reweights a single simulated dataset.

Usage

```
DAGinterventionMC(
  incidences,
  dataParams,
  sampleSize,
  sample = TRUE,
  fixNode = TRUE,
  reducedVarianceSampling = TRUE,
  unrollDBN = TRUE
)
```

Arguments

`incidences`	a single adjacency matrix of a list of adjacency matrices of sampled DAGs, with entry [i,j] equal to 1 when a directed edge exists from node i to node j
`dataParams`	the data and parameters used to learn the DAGs derived from the `scoreparameters` function of the BiDAG package
`sampleSize`	the number of Monte Carlo samples to draw
`sample`	logical indicating whether to sample the parameters of each node from the posterior (TRUE, default) or to take the expectation (FALSE)
`fixNode`	logical indicating whether to intervene on each node (TRUE, default) and resample downstream nodes or to sample once and reweight the sample (FALSE)
`reducedVarianceSampling`	logical indicating whether to perform Bernoulli samping for each node (FALSE) or to sample from a distribution with the same mean and lower variance (TRUE, default)
`unrollDBN`	logical indicating whether to unroll a DBN to a full DAG over all time slices (TRUE, default) or to use the compact representation (FALSE)

Value

a single matrix or a list of matrices containing the full set of intervention effects for each input DAG. Entry [i,j] is the downstream effect on node j of intervening on node i (the difference observed at node j when setting node i to 1 and 0)

See Also

scoreparameters

Examples

```
scoreParam <- BiDAG::scoreparameters("bde", BiDAG::Asia)
causalmatMC <- DAGinterventionMC(BiDAG::Asiamat, scoreParam, 1e4)
```

`DAGparameters` *Augment a DAG with parameters*

Description

`DAGparameters` takes a DAG and augments it with parameters. For binary data these are the parameters of the posterior beta distributions and its mean. For continuous data, these are parameters of the posterior distributions of the edge coefficients from arXiv:2010.00684. There is support for user-defined augmentation, with the caveat that it must match the output format of either the binary or continuous cases.

Usage

```
DAGparameters(incidence, dataParams, unrollDBN = TRUE)
```

Arguments

`incidence`	a single adjacency matrix with entry [i,j] equal to 1 when a directed edge exists from node i to node j
`dataParams`	the data and parameters used to learn the DAGs derived from the `scoreparameters` function of the BiDAG package
`unrollDBN`	logical indicating whether to unroll a DBN to a full DAG over all time slices (TRUE, default) or to use the compact representation (FALSE)

Value

the DAG and a list of parameters for each node given its parents

Examples

```
scoreParam <- BiDAG::scoreparameters("bde", BiDAG::Asia)
AsiaParam <- DAGparameters(BiDAG::Asiamat, scoreParam)
```

Index