Package: omphc (via r-universe)

November 7, 2026

Type Package

Title Bayesian Network Learning Using the OMP Algorithm

Version 1.0

Maintainer Michail Tsagris <mtsagris@uoc.gr>

Depends R (>= 4.0)

Imports bnlearn, pchc, Rcpp, Rfast, stats

LinkingTo Rcpp, RcppArmadillo

Description Bayesian network learning using the orthogonal matching pursuit to perform variable selection and then application of the Hill-Climbing greedy search or the TABU search to orient the edges. Some relevant papers are Tsagris M. and Adam C. (2026) <doi:10.13140/RG.2.2.26278.28489>, Tsagris M. (2021) <doi:10.1007/s10614-020-10065-7>, Tsagris M. (2022) <doi:10.3390/math10152604>, Tsagris et al. (2022) <doi:10.1109/TCBB.2020.3029952>, Tsamardinos I., Brown L.E. and Aliferis C.F. (2006) <doi:10.1007/s10994-006-6889-7>, Mallat S. G. and Zhang Z. (1993) <doi:10.1109/78.258082>.

License GPL (>= 2)

Encoding UTF-8

NeedsCompilation yes

Author Michail Tsagris [aut, cre], Christos Adam [aut] (ORCID: <https://orcid.org/0009-0003-3244-7034>)

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

Date/Publication 2026-11-07 16:57:27 UTC

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

RemoteRef HEAD

RemoteSha 3c36a6c952f040321c255ea7d66cd237a61ffcb2

Contents

omphc-package 2
Adjacency matrix of a Bayesian network 3
omp.network 4
omphc 5
Plot of a Bayesian network 7
rbn 8
rbn2 9
update.skel 11

Index **13**

`omphc-package` *Bayesian Network Learning Using the OMP Algorithm*

Description

Bayesian network learning using the orthogonal matching pursuit to perform variable selection and then application of the Hill-Climbing greedy search or the TABU search to orient the edges. Some relevant papers are Tsagris M. and Adam C. (2026) <doi:10.13140/RG.2.2.26278.28489>, Tsagris M. (2021) <doi:10.1007/s10614-020-10065-7>, Tsagris M. (2022) <doi:10.3390/math10152604>, Tsagris et al. (2022) <doi:10.1109/TCBB.2020.3029952>, Tsamardinos I., Brown E.L. and Aliferis F.C. (2006) <doi:10.1007/s10994-006-6889-7> and Mallat S. G. and Zhang Z. (1993) <doi:10.1109/78.258082>.

Details

Package:	omphc
Type:	Package
Version:	1.0
Date:	2026-10-23
License:	GPL-2

Maintainers

Michail Tsagris <mtsagris@uoc.gr>.

Author(s)

Michail Tsagris <mtsagris@uoc.gr>.

References

Spirtes P., Glymour C. and Scheines R. (2001). Causation, Prediction, and Search. The MIT Press, Cambridge, MA, USA, 3nd edition.

`Adjacency matrix of a Bayesian network`
Adjacency matrix of a Bayesian network

Description

Adjacency matrix of a Bayesian network.

Usage

```
bnmat(dag)
```

Arguments

`dag`	A BN object, an object of class "bn".

Details

The function is called from the "bnlearn" package which invokes the "Rgraphviz" package from Bioconductor and you need to install it first.

Value

Adjacency matrix of a Bayesian network is extracted.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

See Also

omphc

Examples

```
x <- matrix( rnorm(200 * 10, 1, 10), nrow = 200 )
a <- omphc(x)
bnmat(a$dag)
```

omp.network *skeleton of a Bayesian network produced by the OMP algorithm*

Description

skeleton of a Bayesian network produced by the OMP algorithm.

Usage

```
omp.network(x, R = NULL, method = "pvalue", tol = 0.05)
```

Arguments

x	A numerical matrix with the variables. If you have a data.frame (i.e. categorical data) turn them into a matrix. Note, that for the categorical case data, the numbers must start from 0. No missing data are allowed.
R	If the correlation matrix is available, pass it here.
method	The method to proceed with the OMP, p-value based ("pvalue"), BIC based ("BIC"), or adjusted R-square ("ar2") based.
tol	The tolerance value to terminate the OMP. The significance level, a value for the BIC difference, or a percentage for the adjusted R-square.

Details

As in FEDHC the first phase consists of a variable selection procedure, the OMP algorithm. Then, hill-climbing or TABU search is applied to direct the edges.

Value

A list including:

ini.stat	The test statistics of the univariate associations.
ini.pvalue	The initial p-values univariate associations.
pvalue	A matrix with the logarithm of the p-values of the updated associations. This final p-value is the maximum p-value among the two p-values in the end.
runtime	The duration of the algorithm.
ntests	The number of tests conducted during each k.
G	The adjancency matrix. A value of 1 in G[i, j] appears in G[j, i] also, indicating that i and j have an edge between them.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Tsamardinos I., Brown E.L. and Aliferis F.C. (2006). The max-min hill-climbing Bayesian network structure learning algorithm. Machine Learning, 65(1): 31–78.

See Also

omphc

Examples

```
m <- rbn2(1000, p = 50, nei = 5)
G <- m$G
x <- m$x
a <- omp.network(x, tol = 0.1)
```

---

omphc *The OMP-based Bayesian network learning algorithm*

Description

The OMP-based Bayesian network learning algorithm.

Usage

```
omphc(x, skel = NULL, R = NULL, method = "pvalue", tol = 0.05, algo = "hc",
tabu = 10, restart = 10, score = "bic-g", blacklist = NULL, whitelist = NULL)
```

Arguments

x	A numerical matrix with the variables. No missing data are allowed.
skel	If you have the output of the skeleton phase, the output from the function omp.network plug it here. This can save time.
R	If the correlation matrix is available, pass it here.
method	If the skeleton is not available, the method to proceed with the OMP, p-value based ("pvalue"), BIC based ("BIC"), or adjusted R-square ("ar2") based.
tol	The tolerance value to terminate the OMP. The significance level, a value for the BIC difference, or a percentage for the adjusted R-square.
algo	The scoring algorithm, either "hc" or "tabu"
tabu	An integer, the length of the tabu list used in the tabu function.
restart	An integer, the number of random restarts.
score	A character string, the label of the network score to be used in the algorithm. If none is specified, the default score is the Bayesian Information Criterion for continuous data sets. The available score for continuous variables are: "bic-g" (default), "loglik-g", "aic-g", "bic-g" or "bge".

blacklist	A data frame with two columns (optionally labeled "from" and "to"), containing a set of arcs not to be included in the graph.
whitelist	A data frame with two columns (optionally labeled "from" and "to"), containing a set of arcs to be included in the graph.

Details

The OMPHC or OMPTABU algorithm is implemented. The OMP algorithm (Tsagris et al., 2002, Mallat S. G. and Zhang Z. (1993)), is implemented during the skeleton identification phase. Next, the Hill Climbing greedy search or the Tabu search is employed to score the network.

Value

A list including:

ini	A list including the output of the omp.network function.
dag	A "bn" class output. A list including the outcome of the Hill-Climbing or the Tabu search phase. See the package "bnlearn" for more details.
scoring	The score value.
runtime	The duration of the algorithm.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Tsagris M., Papadovasilakis Z., Lakiotaki K. and Tsamardinos I. (2022). The γ-OMP Algorithm for Feature Selection With Application to Gene Expression Data. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 19(2): 1214–1224.

Tsamardinos I., Brown E.L. and Aliferis F.C. (2006). The max-min hill-climbing Bayesian network structure learning algorithm. Machine Learning, 65(1): 31–78.

Mallat S. G. and Zhang Z. (1993). Matching pursuits with time-frequency dictionaries. IEEE Transactions on Signal Processing, 41(12): 3397–3415.

See Also

omp.network

Examples

```
m <- rbn2(1000, p = 50, nei = 5)
G <- m$G
x <- m$x
a <- omphc(x, tol = 0.01)
```

`Plot of a Bayesian network`

Plot of a Bayesian network

Description

Plot of a Bayesian network.

Usage

```
bnplot(dag, shape = "ellipse", main = NULL, sub = NULL, highlight = NULL)
```

Arguments

`dag`	A BN object, an object of class "bn".
`shape`	A character string defining the shape of the nodes, "ellipse" (default value), "circle" or "rectangle".
`main`	The main title of the graph displayed on the top.
`sub`	The subtitle of the graph displayed at the bottom.
`highlight`	A list with options specifying which nodes to plot with different colours. You can also check the package bnlearn or the package Rgraphviz for more information on this, or simply check the example below.

Details

The function is called from the "bnlearn" package which invokes the "Rgraphviz" package from Bioconductor and you need to install it first.

Value

The Bayesian network is visualised.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

See Also

`omphc`

Examples

```
# simulate a dataset with continuous data
x <- matrix( rnorm(100 * 15, 1, 5), nrow = 100 )
a <- omphc(x)
```

rbn *Random values simulation from a Bayesian network*

Description

Random values simulation from a Bayesian network.

Usage

```
rbn(n, dag, x)
```

Arguments

n	The number of observations to generate.
dag	A "bn" object. See the examples for more information.
x	The data used to fit the Bayesian network in a data.frame format.

Details

This information is taken directly from the R package "bnlearn". This function implements forward/logic sampling: values for the root nodes are sampled from their (un-conditional) distribution, then those of their children conditional on the respective parent sets. Thisis done iteratively until values have been sampled for all nodes.If "dagobj" contains NA parameter estimates (because of unobserved discrete parents configurations in the data the parameters were learned from), rbn will produce observations that contain NAs when thoseparents configurations appear in the simulated samples.

Value

A data frame with the same structure (column names and data types) of the argument "data".

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Korb K. and Nicholson A.E. (2010).Bayesian Artificial Intelligence. Chapman & Hall/CRC, 2nd edition.

See Also

omphc

Examples

```
# simulate a dataset with continuous data
x <- matrix( rnorm(200 * 20, 1, 10), nrow = 200 )
a <- omphc(x)
sim <- rbn(100, dag = a$dag, x = x)
```

---

rbn2 *Continuous data simulation from a DAG.*

Description

Contunuous data simulation from a DAG.

Usage

```
rbn2(n, G = NULL, p, nei, low = 0.1, up = 1)
rbn3(n, p, s, a = 0, m, G = NULL, seed = FALSE)
```

Arguments

n	A number indicating the sample size.
p	A number indicating the number of nodes (or vectices, or variables).
nei	The average number of neighbours.
s	A number in $(0, 1)$. This defines somehow the sparseness of the model. It is the probability that a node has an edge.
a	A number in $(0, 1)$. The defines the percentage of outliers to be included in the simulated data. If $a = 0$, no outliers are generated.
m	A vector equal to the number of nodes. This is the mean vector of the normal distribution from which the data are to be generated. This is used only when $a > 0$ so as to define the mena vector of the multivariate normal from which the outliers will be generated.
G	If you already have an an adjacency matrix in mind, plug it in here, otherwise, leave it NULL.
seed	If seed is TRUE, the simulated data will always be the same.
low	Every child will be a function of some parents. The beta coefficients of the parents will be drawn uniformly from two numbers, low and up. See details for more information on this.
up	Every child will be a function of some parents. The beta coefficients of the parents will be drawn uniformly from two numbers, low and up. See details for more information on this.

Details

In the case where no adjacency matrix is given, an $p \times p$ matrix with zeros everywhere is created. Every element below the diagonal is is replaced by random values from a Bernoulli distribution with probability of success equal to s. This is the matrix B. Every value of 1 is replaced by a uniform value in $0.1, 1$. This final matrix is called A. The data are generated from a multivariate normal distribution with a zero mean vector and covariance matrix equal to $(\mathbf{I}_p - A)^{-1}(\mathbf{I}_p - A)$, where $\mathbf{I}_p$ is the $p \times p$ identiy matrix. If a is greater than zero, the outliers are generated from a multivariate normal with the same covariance matrix and mean vector the one specified by the user, the argument "m". The flexibility of the outliers is that you cna specifiy outliers in some variables only or in all of them. For example, m = c(0,0,5) introduces outliers in the third variable only, whereas m = c(5,5,5) introduces outliers in all variables. The user is free to decide on the type of outliers to include in the data.

For the "rdag2", this is a different way of simulating data from DAGs. The first variable is normally generated. Every other variable can be a function of some previous ones. Suppose now that the i-th variable is a child of 4 previous variables. We need for coefficients b_j to multiply the 4 variables and then generate the i-th variable from a normal with mean $\sum_{j=1} b_j X_j$ and variance 1. The b_j will be either positive or negative values with equal probability. Their absolute values ranges between "low" and "up". The code is accessible and you can see in detail what is going on. In addition, every generated data, are standardised to avoid numerical overflow.

Value

A list including:

`nout`	The number of outliers.
`G`	The adcacency matrix used. For the "rdag" if G[i, j] = 2, then G[j, i] = 3 and this means that there is an arrow from j to i. For the "rdag2" the entries are either G[i, j] = G[j, i] = 0 (no edge) or G[i, j] = 1 and G[j, i] = 0 (indicating i -> j).
`A`	The matrix with the with the uniform values in the interval $0.1, 1$. This is returned only by "rdag".
`x`	The simulated data.

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

References

Tsagris M. (2019). Bayesian network learning with the PC algorithm: an improved and correct variation. Applied Artificial Intelligence, 33(2): 101–123.

Tsagris M., Borboudakis G., Lagani V. and Tsamardinos I. (2018). Constraint-based Causal Discovery with Mixed Data. International Journal of Data Science and Analytics, 6: 19–30.

Spirtes P., Glymour C. and Scheines R. (2001). Causation, Prediction, and Search. The MIT Press, Cambridge, MA, USA, 3nd edition.

Colombo Diego, and Marloes H. Maathuis (2014). Order-independent constraint-based causal structure learning. Journal of Machine Learning Research, 15(1): 3741–3782.

See Also

rbn, omphc

Examples

```
x <- rbn3(100, 20, 0.2)$x
a <- omphc(x)
```

---

update.skel *Update the skeleton of the Bayesian network*

Description

The OMP-based Bayesian network learning algorithm.

Usage

```
update.skel(skel, method = "pvalue", tol = 0.01)
```

Arguments

skel	The output of the function omp.network.
method	The method used to construct the skeleton, p-value based ("pvalue"), BIC based ("BIC"), or adjusted R-square ("ar2") based.
tol	The tolerance value to update the skeleton. The significance level, a value for the BIC difference, or a percentage for the adjusted R-square. It has to be smaller than the tolerance value initially used.

Details

Suppose you have run the skeleton of the OMPHC with a tolerance value equal to 0.05 using the "pvalue" method. Now, you wish to update the skeleton for tol=0.01. Instead of running the omp.network again with tol=0.01, you can update the G matrix with this function. The function will change the G matrix, and the Rho list.

Value

The same output as omp.network .

Author(s)

Michail Tsagris.

R implementation and documentation: Michail Tsagris <mtsagris@uoc.gr>.

See Also

omp.network

Examples

```
m <- rbn2(1000, p = 50, nei = 5)
G <- m$G
x <- m$x
a1 <- omp.network(x, tol = 0.1)
a2 <- omp.network(x, tol = 0.01)
b <- update.skel(a1, method = "pvalue", tol = 0.01)
```

Index

Adjacency matrix of a Bayesian network, 3

bnmat *(*Adjacency matrix of a Bayesian network*)*, 3
bnplot *(*Plot of a Bayesian network*)*, 7

omp.network, 4, *5, 6*, *11*
omphc, *3*, 5, *5*, *7, 8*, *11*
omphc-package, 2

Plot of a Bayesian network, 7

rbn, 8, *11*
rbn2, 9
rbn3 *(*rbn2*)*, 9

update.skel, 11