Package: pchc (via r-universe)

September 9, 2024

Type Package

Title Bayesian Network Learning with the PCHC and Related Algorithms

Version 1.2

Date 2023-09-06

Author Michail Tsagris [aut, cre]

Maintainer Michail Tsagris <mtsagris@uoc.gr>

Depends R (>= 4.0)

Imports bigstatsr, bnlearn, dcov, foreach, doParallel, parallel, Rfast, Rfast2, robustbase, stats

Suggests bigreadr, Rgraphviz

Description Bayesian network learning using the PCHC algorithm. PCHC stands for PC Hill-Climbing, a new hybrid algorithm that uses PC to construct the skeleton of the BN and then applies the Hill-Climbing greedy search. More algorithms and variants have been added, such as MMHC, FEDHC, and the Tabu search variants, PCTABU, MMTABU and FEDTABU. The relevant papers are: a) Tsagris M. (2021). A new scalable Bayesian network learning algorithm with applications to economics. Computational Economics, 57(1): 341-367. <doi:10.1007/s10614-020-10065-7>. b) Tsagris M. (2022). The FEDHC Bayesian Network Learning Algorithm. Mathematics 2022, 10(15): 2604. <doi:10.3390/math10152604>.

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2023-09-06 16:20:02 UTC

Contents

---

pchc-package *Bayesian Network Learning with the PCHC and Related Algorithms*

Description

The original version of this package was to learn Bayesian networks with the PCHC algorithm. PCHC stands for PC Hill-Climbing. It is a new hybrid algorithm that used PC to construct the skeleton of the BN and then utilizes the Hill-Climbing greedy search. The package has been expanded to include the MMHC and the FEDHC algortihms. It further includes the PCTABU, MMTABU and FEDTABU algortihms which are pretty much similar. Instead of the Hill Climbing greey search the Tabu search is employed.

Details

Package:	pchc
Type:	Package
Version:	1.2
Date:	2023-09-06
License:	GPL-2

Maintainers

Michail Tsagris <mtsagris@uoc.gr>.

Author(s)

Michail Tsagris <mtsagris@uoc.gr>.

References

Tsagris M. (2022). The FEDHC Bayesian Network Learning Algorithm. Mathematics 2022, 10(15): 2604.

Tsagris M. (2021). A new scalable Bayesian network learning algorithm with applications to economics. Computational Economics 57(1): 341-367.

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

Tsamardinos I., Borboudakis G. (2010) Permutation Testing Improves Bayesian Network Learning. In Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2010. 322-337.

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.

Tsagris M. (2017). Conditional independence test for categorical data using Poisson log-linear model. Journal of Data Science, 15(2):347-356.

Borboudakis G. and Tsamardinos I. (2019). Forward-backward selection with early dropping. Journal of Machine Learning Research, 20(8): 1-39.

`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

pchc, pc.skel

Examples

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

All pairwise G-square and chi-square tests of indepedence

All pairwise G-square and chi-square tests of indepedence

Description

All pairwise G-square and chi-square tests of indepedence.

Usage

```
g2test_univariate(x, dc)
g2test_univariate_perm(x, dc, B)
chi2test_univariate(x, dc)
```

Arguments

`x`	A numerical matrix with the data. **The minimum must be 0, otherwise the function can crash or will produce wrong results**. The data must be consecutive numbers.
`dc`	A numerical value equal to the number of variables (or columns of the data matrix) indicating the number of distinct, unique values (or levels) of each variable. Make sure you give the correct numbers here, otherwise the degrees of freedom will be wrong.
`B`	The number of permutations. The permutations test is slower than without permutations and should be used with small sample sizes or when the contigency tables have zeros. When there are few variables, R's "chisq.test" function is faster, but as the number of variables increase the time difference with R's procedure becomes larger and larger.

Details

The function does all the pairwise G^2 test of independence and gives the position inside the matrix. The user must build the associations matrix now, similarly to the correlation matrix. See the examples of how to do that. The p-value is not returned, we live this to the user. See the examples of how to obtain it.

Value

A list including:

`statistic`	The G^2 or X^2 test statistic for each pair of variables.
`pvalue`	The p-value of the test statistic for each pair of variables.
`x`	The row or variable of the data.
`y`	The column or variable of the data.
`df`	The degrees of freedom of each test.

Author(s)

Michail Tsagris.

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

References

Tsagris M. (2017). Conditional independence test for categorical data using Poisson log-linear model. Journal of Data Science, 15(2):347-356.

Tsamardinos, I. and Borboudakis, G. (2010). Permutation testing improves Bayesian network learning. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases (pp. 322-337). Springer Berlin Heidelberg

See Also

`g2test`, `cat.tests`

Examples

```
nvalues <- 3
nvars <- 10
nsamples <- 1000
x<- matrix( sample( 0:(nvalues - 1), nvars * nsamples, replace = TRUE ), nsamples, nvars )
dc <- rep(nvalues, nvars)
system.time( g2test_univariate(x, dc) )
a <- g2test_univariate(x, dc)
```

`Bootstrap versions of the skeleton of a Bayesian network`

Bootstrap versions of the skeleton of a Bayesian network

Description

Bootstrap versions of the skeleton of a Bayesian network.

Usage

```
pchc.skel.boot(x, method = "pearson", alpha = 0.05, B = 200)
fedhc.skel.boot(x, method = "pearson", alpha = 0.05, B = 200)
mmhc.skel.boot(x, max_k = 3, method = "pearson", alpha = 0.05, B = 200)
```

Arguments

`x`	A matrix with the variables. The user must know if they are continuous or if they are categorical. **If you have categorical data though, the user must transform the data.frame into a matrix. In addition, the numerical matrix must have values starting from 0. For example, 0, 1, 2, instead of "A", "B" and "C"**.
`max_k`	The maximum conditioning set to use in the conditional indepedence test (see Details). Integer, default value is 3.
`method`	If you have continuous data, this "pearson". If you have categorical data though, this must be "cat". In this case, make sure the minimum value of each variable is zero. The function "g2Test" in the R package **Rfast** and the relevant functions work that way.
`alpha`	The significance level (suitable values in (0, 1)) for assessing the p-values. The default value is 0.05.
`B`	The number of bootstrap resamples to draw. The algorithm is performed in each bootstrap sample. In the end, the adjacency matrix on the observed data is returned, along with another adjacency matrix produced by the bootstrap. The latter one contains values from 0 to 1 indicating the proportion of times an edge between two nodes was present.

Value

A list including:

`G`	The observed 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.
`Gboot`	The bootstrapped 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.
`runtime`	The duration of the algorithm.

Author(s)

Michail Tsagris.

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

References

A new scalable Bayesian network learning algorithm with applications to economics. Computational Economics, 57(1): 341-367.

Tsagris M. (2022). The FEDHC Bayesian Network Learning Algorithm. Mathematics 2022, 10(15), 2604.

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

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.

Borboudakis G. and Tsamardinos I. (2019). Forward-backward selection with early dropping. Journal of Machine Learning Research, 20(8): 1-39.

See Also

`pchc.skel`, `fedhc.skel`, `mmhc.skel`, `bn.skel.utils`

Examples

```
x <- pchc::rbn2(500, p = 20, nei = 3)$x
a <- pchc::pchc.skel.boot(x, alpha = 0.05)
```

`Bootstrapping the FEDHC and FEDTABU Bayesian network learning algorithms`

Bootstrapping the FEDHC and FEDTABU Bayesian network learning algorithms

---

Description

Bootstrapping the FEDHC and FEDTABU Bayesian network learning algorithms.

Usage

```
fedhc.boot(x, method = "pearson", alpha = 0.05, ini.stat = NULL, R = NULL,
restart = 10, score = "bic-g", blacklist = NULL, whitelist = NULL, B = 200, ncores = 1)

fedtabu.boot(x, method = "pearson", alpha = 0.05, ini.stat = NULL, R = NULL,
tabu = 10, score = "bic-g", blacklist = NULL, whitelist = NULL, B = 200, ncores = 1)
```

Arguments

x	A numerical matrix with the variables. If you have a data.frame (i.e. categorical data) turn them into a matrix using data.frame.to_matrix. Note, that for the categorical case data, the numbers must start from 0. No missing data are allowed.
method	If you have continuous data, you can choose either "pearson" or "spearman". If you have categorical data though, this must be "cat". In this case, make sure the minimum value of each variable is zero. The g2test and the relevant functions work that way.
alpha	The significance level for assessing the p-values.
ini.stat	If the initial test statistics (univariate associations) are available, pass them through this parameter.
R	If the correlation matrix is available, pass it here.
restart	An integer, the number of random restarts.
tabu	An integer, the length of the tabu list used in the tabu function.
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 both discrete and continuous data sets. The available score for continuous variables are: "bic-g" (default), "loglik-g", "aic-g", "bic-g" or "bge". The available score categorical variables are: "bde", "loglik" or "bic".
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.
B	The number of bootstrap resamples to draw. The algorithm is performed in each bootstrap sample. In the end, the adjacency matrix on the observed data is returned, along with another adjacency matrix produced by the bootstrap. The latter one contains values from 0 to 1 indicating the proportion of times an edge between two nodes was present.
ncores	The number of cores to use, in case of parallel computing.

Details

The FEDHC algorithm is implemented. The FBED algortihm (Borboudakis and Tsamardinos, 2019), without the backward phase, 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:

`mod`	A list including the output of the `pchc` or the `pctabu` function.
`Gboot`	The bootstrapped adjancency matrix of the Bayesian network.
`runtime`	The duration of the algorithm.

Author(s)

Michail Tsagris.

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

References

Tsagris M. (2022). The FEDHC Bayesian Network Learning Algorithm. Mathematics 2022, 10(15): 2604.

Borboudakis G. and Tsamardinos I. (2019). Forward-backward selection with early dropping. Journal of Machine Learning Research, 20(8): 1-39.

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

`pchc`, `mmhc`, `fedhc`, `fedhc.skel`

Examples

```
# simulate a dataset with continuous data
x <- matrix( rnorm(200 * 20, 1, 10), nrow = 200 )
a <- fedhc.boot(x, B = 50)
```

`Bootstrapping the MMHC and MMTABU Bayesian network learning algorithms`

Bootstrapping the MMHC and MMTABU Bayesian network learning algorithms

---

Description

Bootstrapping the MMHC and MMTABU Bayesian network learning algorithms.

Usage

```
mmhc.boot(x, method = "pearson", max_k = 3, alpha = 0.05, ini.stat = NULL,
R = NULL, restart = 10, score = "bic-g", blacklist = NULL, whitelist = NULL,
B = 200, ncores = 1)

mmtabu.boot(x, method = "pearson", max_k = 3, alpha = 0.05, ini.stat = NULL,
R = NULL, tabu = 10, score = "bic-g", blacklist = NULL, whitelist = NULL,
B = 200, ncores = 1)
```

Arguments

x	A numerical matrix with the variables. If you have a data.frame (i.e. categorical data) turn them into a matrix using data.frame.to_matrix. Note, that for the categorical case data, the numbers must start from 0. No missing data are allowed.
method	If you have continuous data, this "pearson". If you have categorical data though, this must be "cat". In this case, make sure the minimum value of each variable is zero. The function "g2Test" in the R package **Rfast** and the relevant functions work that way.
max_k	The maximum conditioning set to use in the conditional indepedence test (see Details). Integer, default value is 3
alpha	The significance level for assessing the p-values.
ini.stat	If the initial test statistics (univariate associations) are available, pass them through this parameter.
R	If the correlation matrix is available, pass it here.
restart	An integer, the number of random restarts.
tabu	An integer, the length of the tabu list used in the tabu function.
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 both discrete and continuous data sets. The available score for continuous variables are: "bic-g" (default), "loglik-g", "aic-g", "bic-g" or "bge". The available score categorical variables are: "bde", "loglik" or "bic".
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.
B	The number of bootstrap resamples to draw. The algorithm is performed in each bootstrap sample. In the end, the adjacency matrix on the observed data is returned, along with another adjacency matrix produced by the bootstrap. The latter one contains values from 0 to 1 indicating the proportion of times an edge between two nodes was present.
ncores	The number of cores to use, in case of parallel computing.

Details

The MMHC algorithm is implemented without performing the backward elimination during the skeleton identification phase. The MMHC as described in Tsamardinos et al. (2006) employs the MMPC algorithm during the skeleton construction phase and the Tabu search in the scoring phase. In this package, the mmhc function employs the Hill Climbing greedy search in the scoring phase while the mmtabu employs the Tabu search.

Value

A list including:

`mod`	A list including the output of the `mmhc` or the `mmtabu` function.
`Gboot`	The bootstrapped adjancency matrix of the Bayesian network.
`runtime`	The duration of the algorithm.

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.

Tsagris M. (2021). A new scalable Bayesian network learning algorithm with applications to economics. Computational Economics, 57(1):341-367.

See Also

`fedhc`, `pchc`, `mmhc.skel`, `mmhc`

Examples

```
# simulate a dataset with continuous data
x <- matrix( rnorm(200 * 20, 1, 10), nrow = 200 )
a <- mmhc.boot(x, B = 50)
```

`Bootstrapping the PCHC and PCTABU Bayesian network learning algorithms`

Bootstrapping the PCHC and PCTABU Bayesian network learning algorithms

---

Description

Bootstrapping the PCHC and PCTABU Bayesian network learning algorithms.

Usage

```
pchc.boot(x, method = "pearson", alpha = 0.05, ini.stat = NULL,
R = NULL, restart = 10, score = "bic-g", blacklist = NULL, whitelist = NULL,
B = 200, ncores = 1)

pctabu.boot(x, method = "pearson", alpha = 0.05, ini.stat = NULL,
R = NULL, tabu = 10, score = "bic-g", blacklist = NULL, whitelist = NULL,
B = 200, ncores = 1)
```

Arguments

x	A numerical matrix with the variables. If you have a data.frame (i.e. categorical data) turn them into a matrix using data.frame.to_matrix. Note, that for the categorical case data, the numbers must start from 0. No missing data are allowed.
method	If you have continuous data, you can choose either "pearson" or "spearman". If you have categorical data though, this must be "cat". In this case, make sure the minimum value of each variable is zero. The g2test and the relevant functions work that way.
alpha	The significance level for assessing the p-values.
ini.stat	If the initial test statistics (univariate associations) are available, pass them through this parameter.
R	If the correlation matrix is available, pass it here.
restart	An integer, the number of random restarts.
tabu	An integer, the length of the tabu list used in the tabu function.
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 both discrete and continuous data sets. The available score for continuous variables are: "bic-g" (default), "loglik-g", "aic-g", "bic-g" or "bge". The available score categorical variables are: "bde", "loglik" or "bic".
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.
B	The number of bootstrap resamples to draw. The algorithm is performed in each bootstrap sample. In the end, the adjacency matrix on the observed data is returned, along with another adjacency matrix produced by the bootstrap. The latter one contains values from 0 to 1 indicating the proportion of times an edge between two nodes was present.
ncores	The number of cores to use, in case of parallel computing.

Details

The PC algorithm as proposed by Spirtes et al. (2001) is first implemented followed by a scoring phase, such as hill climbing or tabu search. The PCHC was proposed by Tsagris (2021), while the PCTABU algorithm is the same but instead of the hill climbing scoring phase, the tabu search is employed.

Value

A list including:

`mod`	A list including the output of the `pchc` or the `pctabu` function.
`Gboot`	The bootstrapped adjancency matrix of the Bayesian network.
`runtime`	The duration of the algorithm.

Author(s)

Michail Tsagris.

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

References

Tsagris M. (2021). A new scalable Bayesian network learning algorithm with applications to economics. Computational Economics, 57(1):341-367.

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

Tsamardinos I. and Borboudakis G. (2010) Permutation Testing Improves Bayesian Network Learning. In Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2010, 322-337.

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

`fedhc`, `mmhc`, `pchc`, `pchc.skel`

Examples

```
# simulate a dataset with continuous data
x <- matrix( rnorm(200 * 20, 1, 10), nrow = 200 )
a <- pchc.boot(x, B = 50)
```

`Check whether a directed graph is acyclic`

Check whether a directed graph is acyclic

Description

Check whether a directed graph is acyclic.

Usage

```
is.dag(dag)
```

Arguments

dag	A square matrix representing a directed graph which contains either 0 or 1, where G[i, j] = 1, means there is an arrow from node i to node j.

Details

The topological sort is performed. If it cannot be performed, NAs are returned. Hence, the functions checks for NAs.

Value

A logical value, TRUE if the matrix represents a DAG and FALSE otherwise.

Author(s)

Michail Tsagris.

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

References

Chickering, D.M. (1995). A transformational characterization of equivalent Bayesian network structures. Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence, Montreal, Canada, 87-98.

See Also

pchc, fedhc, mmhc

Examples

```
G <- pchc::rbn3(100, 20, 0.3)$G
pchc::is.dag(G)  ## TRUE
```

Chi-square and G-square tests of (unconditional) indepdence

Chi-square and G-square tests of (unconditional) indepdence

Description

Chi-square and G-square tests of (unconditional) indepdence.

Usage

```
cat.tests(x, y, logged = FALSE)
```

Arguments

x	A numerical vector or a factor variable with data. The data must be consecutive numbers.
y	A numerical vector or a factor variable with data. The data must be consecutive numbers.
logged	Should the p-values be returned (FALSE) or their logarithm (TRUE)?

Details

The function calculates the test statistic of the X^2 and the G^2 tests of unconditional independence between x and y. x and y need not be numerical vectors like in g2Test. This function is more close to the spirit of MASS' loglm function which calculates both statistics using Poisson log-linear models (Tsagris, 2017).

Value

A matrix with two rows. In each row the X2 or G2 test statistic, its p-value and the degrees of freedom are returned.

Author(s)

Michail Tsagris.

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

References

Tsagris M. (2021). A new scalable Bayesian network learning algorithm with applications to economics. Computational Economics 57(1): 341-367.

Tsagris M. (2017). Conditional independence test for categorical data using Poisson log-linear model. Journal of Data Science, 15(2): 347-356.

See Also

g2test, cortest, pc.skel

Examples

```
x <- rbinom(100, 3, 0.5)
y <- rbinom(100, 2, 0.5)
cat.tests(x, y)
```

Continuous data simulation from a DAG

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.

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`, `pchc`, `fedhc`, `mmhc`

Examples

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

Correlation between pairs of variables

Correlation between pairs of variables

Description

Correlations between pairs of variables.

Usage

```
corpairs(x, y, rho = NULL, logged = FALSE, parallel = FALSE)
```

Arguments

x	A matrix with real valued data.
y	A matrix with real valued data whose dimensions match those of x.
rho	This can be a vector of assumed correlations (equal to the number of variables or the columns of x or y) to be tested. If this is not the case, leave it NULL and only the correlations will be returned.
logged	Should the p-values be returned (FALSE) or their logarithm (TRUE)? This is taken into account only if "rho" is a vector.
parallel	Should parallel implentations take place in C++? The default value is FALSE.

Details

The paired correlations are calculated. For each column of the matrices x and y the correlation between them is calculated.

Value

A vector of correlations in the case of "rho" being NULL, or a matrix with two extra columns, the test statistic and the (logged) p-value.

Author(s)

Michail Tsagris.

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

References

Lambert Diane (1992). Zero-Inflated Poisson Regression, with an Application to Defects in Manufacturing. Technometrics. 34(1):1-14.

Johnson Norman L., Kotz Samuel and Kemp Adrienne W. (1992). Univariate Discrete Distributions (2nd ed.). Wiley

Cohen, A. Clifford (1960). Estimating parameters in a conditional Poisson distribution. Biometrics. 16:203-211.

Johnson, Norman L. Kemp, Adrianne W. Kotz, Samuel (2005). Univariate Discrete Distributions (third edition). Hoboken, NJ: Wiley-Interscience.

See Also

`correls`, `cortest`, `pcor`

Examples

```
x <- matrix( rnorm(100 * 100), ncol = 100)
y <- matrix( rnorm(100 * 100), ncol = 100)
system.time( a <- corpairs(x, y) )
```

`Correlation matrix for FBM class matrices (big matrices)`
Correlation matrix for FBM class matrices (big matrices)

Description

Correlation matrix for FBM class matrices (big matrices).

Usage

```
big_cor(x)
```

Arguments

`x`	An FBM class matrix.

Details

The function accepts a Filebacked Big Matrix (FBM) class matrix and returns the correlation matrix. Check you matrix for possible NA values. For more information see the "bigmemory" and "bigstatsr" packages.

Value

The correlation matrix of the big data x.

Author(s)

Michail Tsagris.

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

See Also

big_read, fedhc.skel, mmhc.skel

Examples

```
require(bigstatsr, quietly = TRUE)
x <- matrix( runif(100 * 50, 1, 100), ncol = 50 )
x <- bigstatsr::as_FBM(x)
a <- pchc::big_cor(x)
```

Correlation significance testing using Fisher's z-transformation

Correlation significance testing using Fisher's z-transformation

---

Description

Correlation significance testing using Fisher's z-transformation.

Usage

```
cortest(y, x, rho = 0, a = 0.05 )
```

Arguments

y	A numerical vector.
x	A numerical vector.
rho	The value of the hypothesised correlation to be used in the hypothesis testing.
a	The significance level used for the confidence intervals.

Details

The function uses the built-in function "cor" which is very fast, then computes a confidence interval and produces a p-value for the hypothesis test.

Value

A vector with 5 numbers; the correlation, the p-value for the hypothesis test that each of them is equal to "rho", the test statistic and the $a/2%$ lower and upper confidence limits.

Author(s)

Michail Tsagris.

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

References

Tsagris M. (2021). A new scalable Bayesian network learning algorithm with applications to economics. Computational Economics 57(1): 341-367.

See Also

pcor, correls, corpairs

Examples

```
x <- rcauchy(60)
y <- rnorm(60)
cortest(y, x)
```

Correlations *Correlation between a vector and a set of variables*

Description

Correlation between a vector and a set of variables.

Usage

```
correls(y, x, type = "pearson", rho = 0, a = 0.05)
```

Arguments

y	A numerical vector.
x	A matrix with the data.
type	The type of correlation you want. "pearson" and "spearman" are the two supported types because their standard error is easily calculated. For the "groupcorrels" you can also put "kendall" because no hypothesis test is performed in that function.
rho	The value of the hypothesised correlation to be used in the hypothesis testing.
a	The significance level used for the confidence intervals.

Details

The functions uses the built-in function "cor" which is very fast and then includes confidence intervals and produces a p-value for the hypothesis test.

Value

A matrix with 5 column; the correlation, the p-value for the hypothesis test that each of them is eaqual to "rho", the test statistic and the $a/2\%$ lower and upper confidence limits.

Author(s)

Michail Tsagris.

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

See Also

corpairs, cortest, pcor

Examples

```
x <- matrix( rnorm(100 * 50 ), ncol = 50)
y <- rnorm(100)
r <- cor(y, x)  ## correlation of y with each of the xs
b <- correls(y, x)
```

Estimation of the percentage of null p-values

Estimation of the percentage of null p-values

Description

Estimation of the percentage of null p-values.

Usage

```
pi0est(p, lambda = seq(0.05, 0.95, by = 0.01), dof = 3)
```

Arguments

p	A vector of p-values.
lambda	A vector of values of the tuning parameter lambda.
dof	Number of degrees of freedom to use when estimating pi_0 with smoothing splines.

Details

The estimated proporiton of null p-values is estimated the algorithm by Storey and Tibshirani (2003).

Value

The estimated proportion of non significant (null) p-values. In the paper Storey and Tibshirani mention that the estimate of pi0 is with lambda=1, but in their R code they use the highest value of lambda and thus we do the same here.

Author(s)

Michail Tsagris.

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

References

Storey J.D. and Tibshirani R. (2003). Statistical significance for genome-wide experiments. Proceedings of the National Academy of Sciences, 100: 9440-9445.

See Also

conf.edge.lower, bn.skel.utils, mmhc.skel

Examples

```
A <- pchc::rbn2(1000, p = 20, nei = 3)
x <- A$x
mod <- pchc::mmhc.skel(x, alpha = 0.05 )
pval <- exp(mod$pvalue)
pval <- lower.tri(pval)
pchc::pi0est(pval)
```

G-square and Chi-square test of conditional indepdence

G-square test of conditional indepdence

Description

G-square test of conditional indepdence with and without permutations.

Usage

```
g2test(x, indx, indy, indz, dc)
chi2test(x, indx, indy, indz, dc)
g2test_perm(x, indx, indy, indz, dc, B)
```

Arguments

x	A numerical matrix with the data. **The minimum must be 0, otherwise the function can crash or will produce wrong results**. The data must be consecutive numbers.
indx	A number between 1 and the number of columns of data. This indicates which variable to take.
indy	A number between 1 and the number of columns of data (other than x). This indicates the other variable whose independence with x is to be tested.
indz	A vector with the indices of the variables to condition upon. It must be non zero and between 1 and the number of variables. If you want unconditional independence test see g2test_univariate, chi2test_univariate and g2test_univariate_perm. If there is an overlap between x, y and cs you will get 0 as the value of the test statistic.
dc	A numerical value equal to the number of variables (or columns of the data matrix) indicating the number of distinct, unique values (or levels) of each variable. Make sure you give the correct numbers here, otherwise the degrees of freedom will be wrong.
B	The number of permutations. The permutations test is slower than without permutations and should be used with small sample sizes or when the contigency tables have zeros. When there are few variables, R's "chisq.test" function is faster, but as the number of variables increase the time difference with R's procedure becomes larger and larger.

Details

The functions calculates the test statistic of the G^2 or the X^2 test of conditional independence between x and y conditional on a set of variable(s) cs.

Value

A list including:

statistic	The G^2 or chi^2 test statistic.
df	The degrees of freedom of the test statistic.
x	The row or variable of the data.
y	The column or variable of the data.

Author(s)

Michail Tsagris.

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

References

Tsagris M. (2021). A new scalable Bayesian network learning algorithm with applications to economics. Computational Economics 57(1): 341-367.

Tsamardinos, I., & Borboudakis, G. (2010). Permutation testing improves Bayesian network learning. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases (pp. 322-337). Springer Berlin Heidelberg

See Also

`cat.tests`, `g2test_univariate`

Examples

```
nvalues <- 2
nvars <- 5
nsamples <- 5000
data <- matrix( sample( 0:(nvalues - 1), nvars * nsamples, replace = TRUE ), nsamples, nvars )
dc <- rep(nvalues, nvars)

g2test( data, 1, 2, 3, c(3, 3, 3) )
g2test_perm( data, 1, 2, 3, c(3, 3, 3), 1000 )
```

`Lower limit of the confidence of an edge`

Lower limit of the confidence of an edge

Description

Lower limit of the confidence of an edge.

Usage

```
conf.edge.lower(p)
```

Arguments

p	A numerical vector with the proportion of times an edge was found in the bootstrapped PC algorithm or the confidence of the edge returned by `bn.skel.utils2`.

Details

After having performed PC algorithm many times in the bootstrap samples (using `mmhc.skel.boot` for example) you get a symmetric matrix with the proportion of times an edge was discovered. Take the lower (or upper) triangular elements of that matrix and pass them as input in this function. This will tell you the minimum proportion required to be confident that an edge is trully significant.

Value

The estimated cutoff limit above which an edge can be deemed significant.

Author(s)

Michail Tsagris.

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

References

Scutari M. and Nagarajan R. (2013). Identifying significant edges in graphical models of molecular networks. Artifficial Intelligence in Medicine, 57: 207-217.

See Also

pchc.skel.boot, mmhc.skel.boot, fedhc.skel.boot

Examples

```
y <- pchc::rbn2(200, p = 30, nei = 3)
x <- y$x
g <- pchc::pchc.skel.boot(x, B = 100)$Gboot
a <- g[ lower.tri(g) ]
pchc::conf.edge.lower(a)
```

Markov blanket of a node in a Bayesian network

Markov blanket of a node in a Bayesian network

Description

Markov blanket of a node in a Bayesian network.

Usage

```
mb(bn, node)
```

Arguments

bn	This can either be a bn object or the adjacency matrix.
node	A vector with one number indicating the node or variable whose Markov blanket is to be returned.

Details

The Markov blanket of a variable (node) is the set of its parents, children and spouses.

Value

`parents`	The parents of the node of interest.
`children`	The children of the node of interest.
`spouses`	The spouses of the node of interest. These are the other parents of the children of the node of interest.
`markov.blanket`	The Markov blanket of the node of interest. The collection of all the previous.

Author(s)

Michail Tsagris.

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

See Also

`pchc`, `fedhc`, `mmhc`, `bnplot`

Examples

```
y <- pchc::rbn3(1000, 10, 0.3)
tru <- y$G
x <- y$x
mod <- pchc(x)
pchc::bnplot(mod$dag)
G <- pchc::bnmat(mod$dag)
pchc::mb(G, 6)
```

Outliers free data via the reweighted MCD

Outliers free data via the reweighted MCD

Description

Outliers free data via the reweighted MCD.

Usage

```
rmcd(x, alpha = NULL)
```

Arguments

`x`	A numerical matrix with the variables. If you have a data.frame (i.e. categorical data) turn them into a matrix using `data.frame.to_matrix`.
`alpha`	A number controlling the size of the subsets over which the determinant is minimized; roughly alpha*n observations are used for computing the determinant. Values between 0.5 and 1 are allowed.

Details

The FEDHC algorithm.

Value

A list including:

`poia`	A vector with the indices of the vectors that were removed.
`x`	The outlier free data.

Author(s)

Michail Tsagris.

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

References

Rousseeuw P. J. and Leroy A. M. (1987) Robust Regression and Outlier Detection. Wiley.

Rousseeuw P. J. and van Driessen K. (1999) A fast algorithm for the minimum covariance determinant estimator. Technometrics 41: 212-223.

Pison G., Van Aelst S., and Willems G. (2002) Small Sample Corrections for LTS and MCD, Metrika 55: 111-123.

Hubert M., Rousseeuw P. J. and Verdonck, T. (2012) A deterministic algorithm for robust location and scatter. Journal of Computational and Graphical Statistics 21: 618-637.

Cerioli A. (2010). Multivariate outlier detection with high-breakdown estimators.Journal of the American Statistical Association 105(489): 147-156.

Cerchiello P. and Giudici P. (2016). Big data analysis for financial risk management. Journal of Big Data 3(1): 18.

See Also

`fedhc.skel`, `pchc.skel`, `mmhc.skel`

Examples

```
x <- matrix( rnorm(200 * 20), nrow = 200 )
x1 <- matrix( rnorm(10 * 20, 10), nrow = 10 )
x <- rbind(x, x1)
a <- pchc::rmcd(x)
a$poia
```

`Partial correlation between two continuous variables`
Partial correlation

Description

Partial correlation between two continuous variables when a correlation matrix is given.

Usage

```
pcor(R, indx, indy, indz, n)
```

Arguments

`R`	A correlation or covariance matrix.
`indx`	The index of the first variable whose conditional correlation is to estimated.
`indy`	The index of the second variable whose conditional correlation is to estimated.
`indz`	The index of the conditioning variables.
`n`	The sample size of the data from which the correlation matrix was computed.

Details

Given a correlation or a covariance matrix the function will caclulate the partial correlation between variables indx and indy conditioning on variable(s) indz and will return the logarithm of the p-value.

Value

A numeric vector containing the partial correlation and logged p-value for the test of no partial correlation.

Author(s)

Michail Tsagris.

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

See Also

`cor2pcor`, `cortest`, `correls`

Examples

```
y <- as.matrix( iris[, 1:2] )
z <- cbind(1, iris[, 3] )
er <- resid( .lm.fit(z, y) )
r <- cor(er)[1, 2]
z <- 0.5 * log( (1 + r) / (1 - r) ) * sqrt( 150 - 1 - 3 )
log(2) + pt( abs(z), 150 - 1 - 3, lower.tail = FALSE, log.p = TRUE )
```

```
r <- cor(iris[, 1:3])
pcor(r, 1,2, 3, 150)
```

`Partial correlation matrix from correlation or covariance matrix`
Partial correlation matrix from correlation or covariance matrix

---

Description

Partial correlation matrix from correlation or covariance matrix.

Usage

```
cor2pcor(R)
```

Arguments

`R`	A correlation or covariance matrix.

Details

Given a correlation or covariance matrix the function will caclulate the pairwise partial correlation conditional on all other variables.

Value

A matrix where each entry is the partial correlation matrix between each pair of variables conditional on all other variables.

Author(s)

Michail Tsagris.

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

See Also

`pcor`, `cortest`, `correls`

Examples

```
x <- as.matrix(iris[, 1:4])
R <- cor(x)
cor2pcor(R)
pcor(R, 1, 2, 3:4, n = 150)
```

`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

`pchc`, `pc.skel`

Examples

```
if (require("Rgraphviz") ) {
# simulate a dataset with continuous data
x <- matrix( rnorm(100 * 15, 1, 5), nrow = 100 )
colnames(x) <- paste("X", 1:15, sep = "")
nam <- colnames(x)
a <- pchc(x)
bnplot(a$dag)
bnplot( a$dag, highlight = list(nodes = nam[c(2, 3)],
col = "tomato", fill = "orange") )
}
```

`Random values simulation from a Bayesian network`

Random values simulation from a Bayesian network

Description

Random values simulation from a Bayesian network.

Usage

```
rbn(n, dagobj, x)
```

Arguments

`n`	The number of observations to generate.
`dagobj`	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

pchc

Examples

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

`Read big data or a big.matrix object`

Read big data or a big.matrix object

Description

Read big data or a big.matrix object.

Usage

```
big_read(big_path, select, header = TRUE, sep = ",")
```

Arguments

`big_path`	The path (including the name) where the big.matrix object is.
`select`	Indices of columns to read (sorted). The length of select will be the number of columns of the resulting filebacked Big Matrix.
`header`	If there are column names, then this should be TRUE.
`sep`	A field delimiter, for example ";" or "," (comma separated). See also `read.csv` for more information.

Details

The data (matrix) which will be read and compressed into a big.matrix object must be of type "numeric". I tested it and it works with "integer" as well. But, in general, bear in mind that only matrices will be read. I have not tested with data.frame for example. However, in the help page of "bigmemory" this is mentioned: Any non-numeric entry will be ignored and replaced with NA, so reading something that traditionally would be a data.frame won't cause an error. A warning is issued. In all cases, the big.matrix is turned into a Filebacked Big Matrix (FBM) of type 'double' the object size is alwasy 680 bytes! If the initial dataset has row names these will be ignored and a column with NAs will apear. So check your final FBM matrix. For more information see the "bigmemory" and "bigstatsr" packages.

Value

A Filebacked Big Matrix (FBM) matrix.

Author(s)

Michail Tsagris.

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

See Also

big_cor, fedhc.skel, mmhc.skel

Examples

```
x <- matrix( runif(100 * 5, 1, 100), ncol = 5 )
```

ROC and AUC *ROC and AUC*

Description

Receiver operating curve and area under the curve.

Usage

```
auc(group, preds, roc = FALSE, cutoffs = NULL)
```

Arguments

group	A numerical vector with the predicted values of each group as 0 and 1.
preds	The predicted values of each group.
roc	If you want the ROC to appear set it to TRUE.
cutoffs	If you provide a vector with decreasing numbers from 1 to 0 that will be used for the ROC, otherwise, the values from 1 to 0 with a step equal to -0.01 will be used.

Details

The ara under the curve is returned. The user has the option of getting the receiver operating curve as well.

Value

A list including:

`cutoffs`	The cutoff values.
`sensitivity`	The sensitivity values for each cutoff value.
`specificity`	The specificity value for each cutoff value.
`youden`	The pair of of 1- specificity and sensitivity where the Youden's J appears on the graph and the Youden index which is defined as the maximum value of sensitivity - specificity + 1.
`auc`	The area under the curve, plus a circle with the point where Youden's J is located. If "roc" is set to FALSE, this is the only item in the list to be returned.

Author(s)

Michail Tsagris.

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

See Also

`bn.skel.utils`, `conf.edge.lower`, `mmhc.skel`

Examples

```
g <- rbinom(150, 1, 0.6)
f <- rnorm(150)
pchc::auc(g, f, roc = FALSE)
```

Skeleton of the FEDHC algorithm

The skeleton of a Bayesian network produced by the FEDHC algorithm

---

Description

The skeleton of a Bayesian network produced by the FEDHC algorithm.

Usage

```
fedhc.skel(x, method = "pearson", alpha = 0.05, robust = FALSE,
ini.stat = NULL, R = NULL, parallel = FALSE)
```

Arguments

x	A numerical matrix with the variables. If you have a data.frame (i.e. categorical data) turn them into a matrix using data.frame.to_matrix. Note, that for the categorical case data, the numbers must start from 0. No missing data are allowed.
method	If you have continuous data, this "pearson". If you have categorical data though, this must be "cat". In this case, make sure the minimum value of each variable is zero. The function "g2Test" in the R package **Rfast** and the relevant functions work that way.
alpha	The significance level (suitable values in (0, 1)) for assessing the p-values. Default value is 0.05.
robust	Do you want outliers to be removed prior to applying the PCHC algorithm? If yes, set this to TRUE to utilise the MCD.
ini.stat	If the initial test statistics (univariate associations) are available, pass them through this parameter.
R	If the correlation matrix is available, pass it here.
parallel	Set this to TRUE if you have millions of observations. In that instance it can reduce the computaitonal time by 1/3.

Details

Similar to MMHC and PCHC the first phase consists of a variable selection procedure, the FBED algortihm (Borboudakis and Tsamardinos, 2019).

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

Tsagris M. (2022). The FEDHC Bayesian Network Learning Algorithm. Mathematics, 10(25): 2604.

Borboudakis G. and Tsamardinos I. (2019). Forward-backward selection with early dropping. Journal of Machine Learning Research, 20(8): 1-39.

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

`pchc.skel`, `mmhc.skel`, `fedhc`, `fedhc.skel.boot`, `dcor.fedhc.skel`

Examples

```
# simulate a dataset with continuous data
x <- matrix( rnorm(200 * 50, 1, 10), nrow = 200 )
a <- fedhc.skel(x)
```

`Skeleton of the FEDHC algorithm using the distance correlation`

The skeleton of a Bayesian network produced by the FEDHC algorithm using the distance correlation

---

Description

The skeleton of a Bayesian network produced by the FEDHC algorithm using the distance correlation.

Usage

```
dcor.fedhc.skel(x, alpha = 0.05, ini.stat = NULL, R = NULL)
```

Arguments

`x`	A numerical matrix with the variables. If you have a data.frame (i.e. categorical data) turn them into a matrix using `data.frame.to_matrix`. Note, that for the categorical case data, the numbers must start from 0. No missing data are allowed.
`alpha`	The significance level (suitable values in (0, 1)) for assessing the p-values. Default value is 0.05.
`ini.stat`	If the initial test statistics (univariate associations) are available, pass them through this parameter.
`R`	If the correlation matrix is available, pass it here.

Details

As in FEDHC the first phase consists of a variable selection procedure, the FBED algortihm (Borboudakis and Tsamardinos, 2019) which is performed though by utilizing the distance correlation (Szekely et al., 2007, Szekely and Rizzo 2014, Huo and Szekely, 2016).

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

Tsagris M. (2022). The FEDHC Bayesian Network Learning Algorithm. Mathematics, 10(25): 2604.

Szekely G.J., Rizzo M.L. and Bakirov N.K. (2007). Measuring and Testing Independence by Correlation of Distances. Annals of Statistics, 35(6):2769-2794.

Szekely G.J. and Rizzo M. L. (2014). Partial distance correlation with methods for dissimilarities. Annals of Statistics, 42(6), 2382-2412.

Huo X. and Szekely G.J. (2016). Fast computing for distance covariance. Technometrics, 58(4), 435-447.

See Also

`fedhc.skel`, `fedhc.skel.boot`

Examples

```
# simulate a dataset with continuous data
x <- matrix( rnorm(500 * 30, 1, 10), nrow = 500 )
a <- dcor.fedhc.skel(x)
```

`Skeleton of the MMHC algorithm`
The skeleton of a Bayesian network learned with the MMHC algorithm

Description

The skeleton of a Bayesian network learned with the MMHC algorithm.

Usage

```
mmhc.skel(x, method = "pearson", max_k = 3, alpha = 0.05,
robust = FALSE, ini.stat = NULL, R = NULL, parallel = FALSE)
```

Arguments

`x`	A numerical matrix with the variables. If you have a data.frame (i.e. categorical data) turn them into a matrix using `data.frame.to_matrix`. Note, that for the categorical case data, the numbers must start from 0. No missing data are allowed.
`method`	If you have continuous data, this "pearson". If you have categorical data though, this must be "cat". In this case, make sure the minimum value of each variable is zero. The function "g2Test" in the R package **Rfast** and the relevant functions work that way.
`max_k`	The maximum conditioning set to use in the conditional indepedence test (see Details). Integer, default value is 3.
`alpha`	The significance level (suitable values in (0, 1)) for assessing the p-values. Default value is 0.05.
`robust`	Do you want outliers to be removed prior to applying the PCHC algorithm? If yes, set this to TRUE to utilise the MCD.
`ini.stat`	If the initial test statistics (univariate associations) are available, pass them through this parameter.
`R`	If the correlation matrix is available, pass it here.
`parallel`	Set this to TRUE if you have millions of observations. In that instance it can reduce the computaitonal time by 1/3.

Details

The max_k option: the maximum size of the conditioning set to use in the conditioning independence test. Larger values provide more accurate results, at the cost of higher computational times. When the sample size is small (e.g., < 50 observations) the max_k parameter should be 3 for example, otherwise the conditional independence test may not be able to provide reliable results.

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., Aliferis, C. F. and Statnikov, A. (2003). Time and sample efficient discovery of Markov blankets and direct causal relations. In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 673-678). ACM.

Brown, L. E., Tsamardinos, I. and Aliferis, C. F. (2004). A novel algorithm for scalable and accurate Bayesian network learning. Medinfo, 711-715.

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

`pchc.skel`, `mmhc.skel`, `mmhc`, `mmhc.skel.boot`

Examples

```
# simulate a dataset with continuous data
x <- matrix( rnorm(300 * 30, 1, 100), nrow = 300 )
a <- mmhc.skel(x)
```

`Skeleton of the PC algorithm`

The skeleton of a Bayesian network learned with the PC algorithm

Description

The skeleton of a Bayesian network learned with the PC algorithm.

Usage

```
pchc.skel(x, method = "pearson", alpha = 0.05,
robust = FALSE, ini.stat = NULL, R = NULL)
```

Arguments

`x`	A numerical matrix with the variables. If you have a data.frame (i.e. categorical data) turn them into a matrix using `data.frame.to_matrix`. Note, that for the categorical case data, the numbers must start from 0. No missing data are allowed.
`method`	If you have continuous data, this "pearson". If you have categorical data though, this must be "cat". In this case, make sure the minimum value of each variable is zero. The function "g2Test" in the R package **Rfast** and the relevant functions work that way.
`alpha`	The significance leve for assessing the p-values.
`robust`	Do you want outliers to be removed prior to applying the PCHC algorithm? If yes, set this to TRUE to utilise the MCD.
`ini.stat`	If the initial test statistics (univariate associations) are available, pass them through this parameter.
`R`	If the correlation matrix is available, pass it here.

Details

The PC algorithm as proposed by Spirtes et al. (2000) is implemented. The variables must be either continuous or categorical, only. The skeleton of the PC algorithm is order independent, since we are using the third heuristic (Spirte et al., 2000, pg. 90). At every stage of the algorithm use the pairs which are least statistically associated. The conditioning set consists of variables which are most statistically associated with each other of the pair of variables.

For example, for the pair (X, Y) there can be two conditioning sets for example (Z1, Z2) and (W1, W2). All p-values and test statistics and degrees of freedom have been computed at the first step of the algorithm. Take the p-values between (Z1, Z2) and (X, Y) and between (Z1, Z2) and (X, Y). The conditioning set with the minimum p-value is used first. If the minimum p-values are the same, use the second lowest p-value. If the unlikely, but not impossible, event of all p-values being the same, the test statistic divided by the degrees of freedom is used as a means of choosing which conditioning set is to be used first.

If two or more p-values are below the machine epsilon (.Machine$double.eps which is equal to 2.220446e-16), all of them are set to 0. To make the comparison or the ordering feasible we use the logarithm of p-value. Hence, the logarithm of the p-values is always calculated and used.

In the case of the G^2 test of independence (for categorical data) with no permutations, we have incorporated a rule of thumb. If the number of samples is at least 5 times the number of the parameters to be estimated, the test is performed, otherwise, independence is not rejected according to Tsamardinos et al. (2006). We have modified it so that it calculates the p-value using permutations.

Value

A list including:

`stat`	The test statistics of the univariate associations.
`ini.pvalue`	The initial p-values univariate associations.
`pvalue`	The logarithm of the p-values of the univariate associations.
`runtime`	The duration of the algorithm.
`kappa`	The maximum value of k, the maximum cardinality of the conditioning set at which the algorithm stopped.
`n.tests`	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.
`sepset`	A list with the separating sets for every value of k.

Author(s)

Michail Tsagris.

R implementation and documentation: 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.

Tsagris M. (2021). A new scalable Bayesian network learning algorithm with applications to economics. Computational Economics, 57(1):341-367.

Tsamardinos I. and Borboudakis G. (2010) Permutation Testing Improves Bayesian Network Learning. In Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2010. 322-337.

See Also

`fedhc.skel`, `mmhc.skel`, `pchc`, `pchc.skel.boot`

Examples

```
# simulate a dataset with continuous data
x <- matrix( rnorm(300 * 30, 1, 100), nrow = 300 )
a <- pchc::pchc.skel(x)
```

`The FEDHC and FEDTABU Bayesian network learning algorithms`
The FEDHC and FEDTABU Bayesian network learning algorithms

Description

The FEDHC and FEDTABU Bayesian network learning algorithms.

Usage

```
fedhc(x, method = "pearson", alpha = 0.05, robust = FALSE, skel = NULL,
ini.stat = NULL, R = NULL, restart = 10, score = "bic-g", blacklist = NULL,
whitelist = NULL)

fedtabu(x, method = "pearson", alpha = 0.05, robust = FALSE, skel = NULL,
ini.stat = NULL, R = NULL, tabu = 10, score = "bic-g", blacklist = NULL,
whitelist = NULL)
```

Arguments

`x`	A numerical matrix with the variables. If you have a data.frame (i.e. categorical data) turn them into a matrix using `data.frame.to_matrix`. Note, that for the categorical case data, the numbers must start from 0. No missing data are allowed.
`method`	If you have continuous data, you can choose either "pearson" or "spearman". If you have categorical data though, this must be "cat". In this case, make sure the minimum value of each variable is zero. The `g2test` and the relevant functions work that way.
`alpha`	The significance level for assessing the p-values.
`robust`	Do you want outliers to be removed prior to applying the FEDHC algorithm? If yes, set this to TRUE to utilise the MCD.
`skel`	If you have the output of the skeleton phase, the output from the function `fedhc.skel` plug it here. This can save time.
`ini.stat`	If the initial test statistics (univariate associations) are available, pass them through this parameter.
`R`	If the correlation matrix is available, pass it here.
`restart`	An integer, the number of random restarts.
`tabu`	An integer, the length of the tabu list used in the tabu function.
`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 both discrete and continuous data sets. The available score for continuous variables are: "bic-g" (default), "loglik-g", "aic-g", "bic-g" or "bge". The available score categorical variables are: "bde", "loglik" or "bic".

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 FEDHC algorithm is implemented. The FBED algortihm (Borboudakis and Tsamardinos, 2019), without the backward phase, 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 fedhc.skel 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. (2022). The FEDHC Bayesian Network Learning Algorithm. Mathematics 2022, 10(15): 2604.

Borboudakis G. and Tsamardinos I. (2019). Forward-backward selection with early dropping. Journal of Machine Learning Research, 20(8): 1-39.

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

pchc, mmhc, fedhc.skel, fedhc.boot

Examples

```
# simulate a dataset with continuous data
x <- matrix( rnorm(300 * 30, 1, 10), nrow = 300 )
a <- fedhc(x)
```

`The MMHC and MMTABU Bayesian network learning algorithms`
The MMHC and MMTABU Bayesian network learning algorithms

Description

The MMHC and MMTABU Bayesian network learning algorithms.

Usage

```
mmhc(x, method = "pearson", max_k = 3, alpha = 0.05, robust = FALSE,
skel = NULL, ini.stat = NULL, R = NULL, restart = 10, score = "bic-g",
blacklist = NULL, whitelist = NULL)

mmtabu(x, method = "pearson", max_k = 3, alpha = 0.05, robust = FALSE,
skel = NULL, ini.stat = NULL, R = NULL, tabu = 10, score = "bic-g",
blacklist = NULL, whitelist = NULL)
```

Arguments

`x`	A numerical matrix with the variables. If you have a data.frame (i.e. categorical data) turn them into a matrix using `data.frame.to_matrix`. Note, that for the categorical case data, the numbers must start from 0. No missing data are allowed.
`method`	If you have continuous data, this "pearson". If you have categorical data though, this must be "cat". In this case, make sure the minimum value of each variable is zero. The function "g2Test" in the R package **Rfast** and the relevant functions work that way.
`max_k`	The maximum conditioning set to use in the conditional indepedence test (see Details). Integer, default value is 3
`alpha`	The significance level for assessing the p-values.
`robust`	Do you want outliers to be removed prior to applying the MMHC algorithm? If yes, set this to TRUE to utilise the MCD.
`skel`	If you have the output of the skeleton phase, the output from the function `mmhc.skel` plug it here. This can save time.
`ini.stat`	If the initial test statistics (univariate associations) are available, pass them through this parameter.
`R`	If the correlation matrix is available, pass it here.
`restart`	An integer, the number of random restarts.
`tabu`	An integer, the length of the tabu list used in the tabu function.
`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 both discrete and continuous data sets. The available score for continuous variables are: "bic-g" (default), "loglik-g", "aic-g", "bic-g" or "bge". The available score categorical variables are: "bde", "loglik" or "bic".

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 MMHC algorithm is implemented without performing the backward elimination during the skeleton identification phase. The MMHC as described in Tsamardinos et al. (2006) employs the MMPC algorithm during the skeleton construction phase and the Tabu search in the scoring phase. In this package, the mmhc function employs the Hill Climbing greedy search in the scoring phase while the mmtabu employs the Tabu search.

Value

A list including:

ini	A list including the output of the mmhc.skel 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

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.

Tsagris M. (2021). A new scalable Bayesian network learning algorithm with applications to economics. Computational Economics, 57(1):341-367.

See Also

fedhc, pchc, mmhc.skel, mmhc.boot

Examples

```
# simulate a dataset with continuous data
x <- matrix( rnorm(300 * 30, 1, 10), nrow = 300 )
a <- mmhc(x)
```

`The PCHC and PCTABU Bayesian network learning algorithms`
The PCHC and PCTABU Bayesian network learning algorithms

Description

The PCHC and PCTABU Bayesian network learning algorithms.

Usage

```
pchc(x, method = "pearson", alpha = 0.05, robust = FALSE, skel = NULL,
ini.stat = NULL, R = NULL, restart = 10, score = "bic-g", blacklist = NULL,
whitelist = NULL)

pctabu(x, method = "pearson", alpha = 0.05, robust = FALSE, skel = NULL,
ini.stat = NULL, R = NULL, tabu = 10, score = "bic-g", blacklist = NULL,
whitelist = NULL)
```

Arguments

`x`	A numerical matrix with the variables. If you have a data.frame (i.e. categorical data) turn them into a matrix using `data.frame.to_matrix`. Note, that for the categorical case data, the numbers must start from 0. No missing data are allowed.
`method`	If you have continuous data, you can choose either "pearson" or "spearman". If you have categorical data though, this must be "cat". In this case, make sure the minimum value of each variable is zero. The `g2test` and the relevant functions work that way.
`alpha`	The significance level for assessing the p-values.
`robust`	Do you want outliers to be removed prior to applying the PCHC algorithm? If yes, set this to TRUE to utilise the MCD.
`skel`	If you have the output of the skeleton phase, the output from the function `pchc.skel` plug it here. This can save time.
`ini.stat`	If the initial test statistics (univariate associations) are available, pass them through this parameter.
`R`	If the correlation matrix is available, pass it here.
`restart`	An integer, the number of random restarts.
`tabu`	An integer, the length of the tabu list used in the tabu function.
`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 both discrete and continuous data sets. The available score for continuous variables are: "bic-g" (default), "loglik-g", "aic-g", "bic-g" or "bge". The available score categorical variables are: "bde", "loglik" or "bic".

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 PC algorithm as proposed by Spirtes et al. (2001) is first implemented followed by a scoring phase, such as hill climbing or tabu search. The PCHC was proposed by Tsagris (2021), while the PCTABU algorithm is the same but instead of the hill climbing scoring phase, the tabu search is employed.

Value

A list including:

ini	A list including the output of the pchc.skel 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. (2021). A new scalable Bayesian network learning algorithm with applications to economics. Computational Economics, 57(1):341-367.

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

Tsamardinos I. and Borboudakis G. (2010) Permutation Testing Improves Bayesian Network Learning. In Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2010, 322-337.

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

fedhc, mmhc, pchc.skel, pchc.boot

Examples

```
# simulate a dataset with continuous data
x <- matrix( rnorm(300 * 30, 1, 10), nrow = 300 )
a <- pchc(x)
```

`Topological sort of a Bayesian network`
Topological sort of a Bayesian network

Description

Topological sort of a Bayesian network.

Usage

```
topological_sort(dag)
```

Arguments

`dag`	A square matrix representing a directed graph which contains 0s and 1s. If G[i, j] = 1 it means there is an arrow from node i to node j. When there is no edge between nodes i and j if G[i, j] = 0.

Details

The function is an R translation from an old matlab code.

Value

A vector with numbers indicating the sorting. If the matrix does not correspond to a Bayesian network (or a DAG), NA will be returned.

Author(s)

Michail Tsagris.

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

References

Chickering, D.M. (1995). A transformational characterization of equivalent Bayesian network structures. Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence, Montreal, Canada, 87-98.

See Also

`bnplot`, `pchc`, `pchc.skel`

Examples

```
y <- pchc::rbn3(100, 20, 0.2)
Gtrue <- y$G
a <- pchc::pchc(y$x)
G <- bnmat(a$dag)
topological_sort(G)
```

Utilities for the skeleton of a (Bayesian) network

Utilities for the skeleton of a (Bayesian) Network

Description

Utilities for the skeleton of a (Bayesian) Network

Usage

```
bn.skel.utils(bnskel.obj, G = NULL, roc = TRUE, alpha = 0.05)
bn.skel.utils2(bnskel.obj, G = NULL, roc = TRUE, alpha = 0.05)
```

Arguments

bnskel.obj	An object as retured by pc.skel, glmm.pc.skel or mmhc.skel.
G	The true adjacency matrix with 1 indicating an edge and zero its absence. Symmetric or not is not important. If this is not available, leave it NULL.
roc	Do you want a graph with the ROC curve be returned? Default value is TRUE.
alpha	The significance level (suitable values in (0, 1)) for assessing the p-values. Default value is 0.01.

Details

Given the true adjaceny matrix one can evaluate the estimated adjacency matrix, skeleton, of the PC or the MMHC algorithm.

The bn.skels.utils give you the area under the curve, false discovery rate and sorting of the edges based on their p-values.

The bn.skel.utils2 estimates the confidence of each edge. The estimated proportion of null p-values is estimated the algorithm by Storey and Tibshirani (2003).

Value

For the "bn.skel.utils" a list including:

fdr	The false discovery rate as estimated using the Benjamini-Hochberg correction.
area	This is a list with the elements of the auc function. The area under the curve, the sensitivy and specificity for a range of values, the Youden index, etc.
sig.pvalues	A matrix with the row and column of each significant p-value sorted in asending order. As we move down the matrix, the p-values increase and hence the strength of the associations decreases.

For the "bn.skel.utils2" a list including:

area	This is a list with the elements of the auc function. The area under the curve, the sensitivy and specificity for a range of values, the Youden index, etc.

pxy	A matrix with the row and column of the confidence of each p-value sorted in asending order. As we move down the matrix, the confidences decrease.
lower	The lower confidcence limit of an edge as estimated by conf.edge.lower.

Author(s)

Michail Tsagris.

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

References

Tsamardinos I. and Brown L.E. Bounding the False Discovery Rate in Local Bayesian Network Learning. AAAI, 2008.

Triantafillou S., Tsamardinos I. and Roumpelaki A. (2014). Learning neighborhoods of high confidence in constraint-based causal discovery. In European Workshop on Probabilistic Graphical Models, pp. 487-502.

Storey J.D. and Tibshirani R. (2003). Statistical significance for genome-wide experiments. Proceedings of the National Academy of Sciences, 100: 9440-9445.

Benjamini Y. and Hochberg Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. Journal of the Royal Statistical Society Series B, 57(1), 289-300.

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

See Also

conf.edge.lower, pchc.skel, rbn2

Examples

```
y <- pchc::rbn2(200, p = 25, nei = 3)
x <- y$x
G <- y$G
mod <- pchc::pchc.skel(x, method = "pearson", alpha = 0.05)
G <- G + t(G)
bn.skel.utils(mod, G, roc = FALSE, alpha = 0.05)
bn.skel.utils2(mod, G, roc = FALSE, alpha = 0.05)
```

`Variable selection for continuous data using the FBED algorithm`

Variable selection for continuous data using the FBED algorithm

---

Description

Variable selection for continuous data using the FBED algorithm.

Usage

```
cor.fbed(y, x, ystand = TRUE, xstand = TRUE, alpha = 0.05, K = 0)
```

Arguments

y	The response variable, a numeric vector.
x	A matrix with the data, where the rows denote the samples and the columns are the variables.
ystand	If this is TRUE the response variable is centered. The mean is subtracted from every value.
xstand	If this is TRUE the independent variables are standardised.
alpha	The significance level, set to 0.05 by default.
K	The number of times to repeat the process. The default value is 0.

Details

FBED stands for Forward Backward with Earcly Dropping. It is a variation of the classical forward selection, where at each step, only the statistically significant variables carry on. The rest are dropped. The process stops when no other variables can be selected. If K = 1, the process is repeated testing sequentially again all those that have not been selected. If K > 1, then this is repeated.

In the end, the backward selection is performed to remove any falsely included variables. This backward phase has not been implemented yet.

Value

A list including:

runtime	The duration of the process.
res	A matrix with the index of the selected variable, their test statistic value and the associated p-value.
info	A matrix with two columns. The cumulative number of variables selected and the number of tests for each value of K.

Author(s)

Michail Tsagris.

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

References

Borboudakis G. and Tsamardinos I. (2019). Forward-backward selection with early dropping. Journal of Machine Learning Research, 20(8): 1-39.

See Also

pc.sel, mmpc, cortest, correls

Examples

```
x <- matrix( rnorm(50 * 50), ncol = 50 )
y <- rnorm(50)
a <- pchc::cor.fbed(y, x)
a
```

`Variable selection for continuous data using the MMPC algorithm`
Variable selection for continuous data using the MMPC algorithm

---

Description

Variable selection for continuous data using the MMPC algorithm.

Usage

```
mmpc(y, x, max_k = 3, alpha = 0.05, method = "pearson",
ini = NULL, hash = FALSE, hashobject = NULL, backward = FALSE)
```

Arguments

`y`	The class variable. Provide a numeric vector.
`x`	The main dataset. Provide a numeric matrix.
`max_k`	The maximum conditioning set to use in the conditional independence test. Provide an integer. The default value set is 3.
`alpha`	Threshold for assessing p-values' significance. Provide a double value, between 0.0 and 1.0. The default value set is 0.05.
`method`	Currently only "pearson" is supported.
`ini`	This argument is used for the avoidance of the univariate associations re-calculations, in the case of them being present. Provide it in the form of a list.
`hash`	Boolean value for the activation of the statistics storage in a hash type object. The default value is false.
`hashobject`	This argument is used for the avoidance of the hash re-calculation, in the case of them being present, similarly to ini argument. Provide it in the form of a hash. Please note that the generated hash object should be used only when the same dataset is re-analyzed, possibly with different values of max_k and alpha.
`backward`	Boolean value for the activation of the backward/symmetry correction phase. This option removes and falsely included variables in the parents and children set of the target variable. The backward option seems dubious. Please do not use at the moment.

Details

The MMPC function implements the MMPC algorithm as presented in "Tsamardinos, Brown and Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm" http://www.dsl-lab.org/supplements/mmhc_paper/paper_online.pdf

Value

The output of the algorithm is an list including:

`selected`	The order of the selected variables according to the increasing pvalues.
`hashobject`	The hash object containing the statistics calculated in the current run.
`pvalues`	For each feature included in the dataset, this vector reports the strength of its association with the target in the context of all other variables. Particularly, this vector reports the max p-values found when the association of each variable with the target is tested against different conditional sets. Lower values indicate higher association.
`stats`	The statistics corresponding to the aforementioned pvalues (higher values indicate higher association).
`univ`	This is a list with the univariate associations; the test statistics and their corresponding logged p-values.
`max_k`	The max_k value used in the current execution.
`alpha`	The alpha value used in the current execution.
`n.tests`	If hash = TRUE, the number of tests performed will be returned. If hash != TRUE, the number of univariate associations will be returned.
`runtime`	The time (in seconds) that was needed for the execution of algorithm.

Author(s)

Michail Tsagris.

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

References

Tsagris M. and Tsamardinos I. (2019). Feature selection with the R package MXM. F1000Research 7: 1505

Feature Selection with the R Package MXM: Discovering Statistically Equivalent Feature Subsets, Lagani V. and Athineou G. and Farcomeni A. and Tsagris M. and Tsamardinos I. (2017). Journal of Statistical Software, 80(7).

Tsamardinos, I., Aliferis, C. F. and Statnikov, A. (2003). Time and sample efficient discovery of Markov blankets and direct causal relations. In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 673-678). ACM.

Brown L. E., Tsamardinos, I. and Aliferis C. F. (2004). A novel algorithm for scalable and accurate Bayesian network learning. Medinfo, 711-715.

Tsamardinos, Brown and Aliferis (2006). The max-min hill-climbing Bayesian network structure learning algorithm. Machine learning, 65(1), 31-78.

See Also

cor.fbed, pc.sel

Examples

```
x <- matrix( rnorm(50 * 50), ncol = 50 )
y <- rnorm(50)
a <- pchc::mmpc(y, x)
```

`Variable selection for continuous data using the PC-simple algorithm`

Variable selection for continuous data using the PC-simple algorithm

---

Description

Variable selection for continuous data using the PC-simple algorithm.

Usage

```
pc.sel(y, x, ystand = TRUE, xstand = TRUE, alpha = 0.05)
```

Arguments

y	A numerical vector with continuous data.
x	A matrix with numerical data; the independent variables, of which some will probably be selected.
ystand	If this is TRUE the response variable is centered. The mean is subtracted from every value.
xstand	If this is TRUE the independent variables are standardised.
alpha	The significance level.

Details

Variable selection for continuous data only is performed using the PC-simple algorithm (Buhlmann, Kalisch and Maathuis, 2010). The PC algorithm used to infer the skeleton of a Bayesian Network has been adopted in the context of variable selection. In other words, the PC algorithm is used for a single node.

Value

A list including:

vars	A vector with the selected variables.
n.tests	The number of tests performed.
runtime	The runtime of the algorithm.

Author(s)

Michail Tsagris.

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

References

Buhlmann P., Kalisch M. and Maathuis M. H. (2010). Variable selection in high-dimensional linear models: partially faithful distributions and the PC-simple algorithm. Biometrika, 97(2): 261-278. https://arxiv.org/pdf/0906.3204.pdf

See Also

mmpc, cor.fbed

Examples

```
x <- matrix( rnorm(50 * 50), ncol = 50 )
y <- rnorm(50)
a <- pchc::pc.sel(y, x)
```

Index