Package 'pchc'

August 28, 2020

Type Package

Title Bayesian Network Learning with the PCHC Algorithm

Version 0.2

URL

Date 2020-08-27

Author Michail Tsagris [aut, cre]

Maintainer Michail Tsagris <mtsagris@uoc.gr>

Depends R (>= 3.6.0)

Imports bnlearn, Rfast, stats

Description Bayesian network learning using 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 relevant paper has been submitted and is currently in revision.

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2020-08-28 18:30:03 UTC

R topics documented:

pchc-package *Bayesian Network Learning with the PCHC Algorithm*

Description

Bayesian network learning 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 relevant paper has been submitted and is currently in revision.

Details

Package:	pchc
Type:	Package
Version:	0.2
Date:	2020-08-27
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.

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.

`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 χ^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. (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)
```

`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>.

See Also

pcor

Examples

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

`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` 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 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)

Giorgos Borboudakis. The permutation version used a C++ code by John Burkardt.

R implementation and documentation: Manos Papadakis <papadakm95@gmail.com>.

References

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`,`cortest`

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 )
```

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 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 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

cortest,pc.skel

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)
```

Plot of a Bayesian network

Plot of a Bayesian network

Description

Plot of a Bayesian network.

Usage

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

Arguments

dag	A BN object, an object of class "bn".
shape	A character string defining the shape of the nodes, "circle", "ellipse" or "rectangle".
main	The main title of the graph displayed on the top.
sub	The subtitle of the graph displayed at the bottom.

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

```
## Not run:
# simulate a dataset with continuous data
x <- matrix( rnorm(200 * 10, 1, 10), nrow = 200 )
a <- pchc(x)
bnplot(a$dag)

## End(Not run)
```

`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 thedata 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(x)
sim <- rbn( 100, dagobj = a$dag, x = x )
```

`Skeleton of the PC algorithm`

The skeleton of a Bayesian network produced by the PC algorithm

---

Description

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

Usage

```
pchc.skel(x, method = "pearson", alpha = 0.05, ini.stat = NULL, ini.pvalue = 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.
`ini.stat`	If the initial test statistics (univariate associations) are available, pass them through this parameter.
`ini.pvalue`	if the initial p-values of the univariate associations are available, pass them through this parameter.

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 amount of time it took to run 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.

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

See Also

pchc

Examples

```
# simulate a dataset with continuous data
x <- matrix( rnorm(200 * 50, 1, 100), nrow = 200 )
a <- pchc.skel(x)
```

The PCHC Bayesian network learning algorithm

The PCHC Bayesian network learning algorithm

Description

The PCHC Bayesian network learning algorithm.

Usage

```
pchc(x, method = "pearson", alpha = 0.05, ini.stat = NULL,
ini.pvalue = NULL, restart = 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.
ini.stat	If the initial test statistics (univariate associations) are available, pass them through this parameter.
ini.pvalue	if the initial p-values of the univariate associations are available, pass them through this parameter.
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 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. (2000) is first implemented followed by a scoring phase, such as hill climbing.

Value

A list including:

`a`	A list including the output of the `pchc.skel` funciton.
`dag`	A "bn" class output. A list including the outcome of the Hill-Climbing phase. See the R package "bnlearn" for more details.
`mhvale`	A data frame with two columns (optionally labeled "from" and "to"), containing a set of arcs not to be included in the graph.
`scoring`	The score value.
`runtime`	The amount of time it took to run the algorithm.

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.

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.

See Also

`pchc.skel`

Examples

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

Index