Package ‘NetworkComparisonTest’

October 12, 2022

Type Package

Title Statistical Comparison of Two Networks Based on Three Invariance Measures

Version 2.2.1

Maintainer Claudia van Borkulo <`cvborkulo@gmail.com`>

Imports qgraph, IsingFit, IsingSampler, reshape2, stats, graphics, utils, Matrix, methods

Suggests networktools

Description This permutation based hypothesis test, suited for Gaussian and binary data, assesses the difference between two networks based on several invariance measures (e.g., network structure invariance, global strength invariance, edge invariance). Network structures are estimated with l1-regularized partial correlations (Gaussian data) or with l1-regularized logistic regression (eLasso, binary data). Suited for comparison of independent and dependent samples. For dependent samples, only supported for data of one group which is measured twice. See van Borkulo et al. (2017) <doi:10.13140/RG.2.2.29455.38569>.

License GPL-2

RoxygenNote 6.1.1

NeedsCompilation no

Author Claudia van Borkulo [aut, cre],
Sacha Epskamp [aut],
Payton Jones [aut],
Jonas Haslbeck [ctb],
Alex Millner [ctb]

Repository CRAN

Date/Publication 2019-06-28 08:10:02 UTC

R topics documented:

`NetworkComparisonTest-package`

Statistical Comparison of Two Networks Based on Three Invariance Measures

---

Description

This permutation based hypothesis test, suited for gaussian and binary data, assesses the difference between two networks based on several invariance measures (network structure invariance, global strength invariance, edge invariance). Network structures are estimated with l1-regularized partial correlations (gaussian data) or with l1-regularized logistic regression (eLasso, binary data). Suited for comparison of independent and dependent samples. For dependent samples, only supported for data of one group which is measured twice.

Details

Package:	NetworkComparisonTest
Type:	Package
Version:	2.2.1
License:	GPL-2

Author(s)

Claudia D. van Borkulo, with contributions from Jonas Haslbeck, Sacha Epskamp, Payton Jones and Alex Millner

Maintainer: Claudia D. van Borkulo <cvborkulo@gmail.com>

References

Ernst, M.D. Permutation methods: A basis for exact inference. Stat Sci. 2004;19(4):676-685.

Good, P.I. Permutation, parametric and bootstrap tests of hypotheses. Vol. 3. New York:: Springer, 2005.

van Borkulo, C. D., Boschloo, L., Borsboom, D., Penninx, B. W. J. H., Waldorp, L. J., & Schoevers, R.A. (2015). Association of symptom network structure with the course of depression. JAMA Psychiatry. 2015;72(12). doi:10.1001/jamapsychiatry.2015.2079

van Borkulo, C. D., Boschloo, Kossakowski, J., Tio, P., L., Schoevers, R.A., Borsboom, D., & , Waldorp, L. J. (2016). Comparing network structures on three aspects: A permutation test. doi:10.13140/RG.2.2.29455.38569

`NCT-methods` *Methods for NCT objects*

Description

Print method, prints the NCT output, plot method plots the output, summary method returns a summary of the output.

Usage

```
## S3 method for class 'NCT'
print(x,...)
## S3 method for class 'NCT'
summary(object,...)
## S3 method for class 'NCT'
plot(x, what = c("strength","network","edge","centrality"),...)
```

Arguments

`x`	output of `NCT`
`object`	output of `NCT`
`what`	defines what has to be plotted: results pertaining to test on invariance of global strength ("strength"), network structure ("network"), edge strength ("edge"), or specific centrality measure ("centrality")
`...`	Arguments only used in plot method

Author(s)

Claudia D. van Borkulo, with contributions from Jonas Haslbeck, Sacha Epskamp, Payton Jones and Alex Millner

Maintainer: Claudia D. van Borkulo <cvborkulo@gmail.com>

`NetworkComparisonTest` *Statistical Comparison of Two Networks Based on Three Invariance Measures*

Description

This permutation based hypothesis test, suited for gaussian and binary data, assesses the difference between two networks based on several invariance measures (network structure invariance, global strength invariance, edge invariance). Network structures are estimated with l1-regularized partial correlations (gaussian data) or with l1-regularized logistic regression (eLasso, binary data). Suited for comparison of independent and dependent samples. For dependent samples, only supported for data of one group which is measured twice.

Usage

```
NCT(data1, data2,
      gamma, it = 100, binary.data=FALSE,
      paired=FALSE, weighted=TRUE, AND=TRUE, abs=TRUE,
      test.edges=FALSE, edges="all",
      progressbar=TRUE, make.positive.definite=TRUE,
      p.adjust.methods= c("none","holm","hochberg","hommel",
                        "bonferroni","BH","BY","fdr"),
      test.centrality=FALSE,
      centrality=c("strength","expectedInfluence"),nodes="all",
      communities=NULL,useCommunities="all",
      estimator, estimatorArgs = list(),
      verbose = TRUE)
```

Arguments

`data1`	One of two datasets. The dimension of the matrix is nobs x nvars; each row is a vector of observations of the variables. Must be cross-sectional data. Can also be the result of `estimateNetwork` from the bootnet package.
`data2`	The other of two datasets. The dimension of the matrix is nobs x nvars; each row is a vector of observations of the variables. Must be cross-sectional data. Can also be the result of `estimateNetwork` from the bootnet package.
`gamma`	A single value between 0 and 1. When not entered, gamma is set to 0.25 for binary data and 0.50 for gaussian data. Networks are estimated with this value for hyperparameter gamma in the extended BIC.
`it`	The number of iterations (permutations).
`binary.data`	Logical. Can be TRUE or FALSE to indicate whether the data is binary or not. If binary.data is FALSE, the data is regarded gaussian.
`paired`	Logical. Can be TRUE of FALSE to indicate whether the samples are dependent or not. If paired is TRUE, relabeling is performed within each pair of observations. If paired is FALSE, relabeling is not restricted to pairs of observations. Note that, currently, dependent data is assumed to entail one group measured twice.
`weighted`	Logical. Can be TRUE of FALSE to indicate whether the networks to be compared should be weighted of not. If not, the estimated networks are dichotomized. Defaults to TRUE.
`AND`	Logical. Can be TRUE of FALSE to indicate whether the AND-rule or the OR-rule should be used to define the edges in the network. Defaults to TRUE. Only necessary for binary data.
`abs`	Logical. Should global strength consider the absolute value of edge weights, or the raw value (i.e., global expected influence)?
`test.edges`	Logical. Can be TRUE of FALSE to indicate whether or not differences in individual edges should be tested.
`edges`	Character or list. When 'all', differences between all individual edges are tested. When provided a list with one or more pairs of indices referring to variables, the provided edges are tested.

`progressbar`	Logical. Should the pbar be plotted in order to see the progress of the estimation procedure? Defaults to TRUE.
`make.positive.definite`	If `make.positive.definite` = `TRUE`, the covariance matrices used for the glasso are projected to the nearest positive definite matrices, if they are not yet positive definite. This is useful for small n, for which it is very likely that at least one of the bootstrap comparisons involves a covariance matrix that is not positive definite.
`p.adjust.methods`	Character. Can be one of "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", or "none". To control (or not) for testing of multiple edges. Defaults to "none".
`test.centrality`	Logical. Should centrality metrics be compared across networks?
`centrality`	Type of centrality metrics to test. Can be any of c("all", "closeness", "betweenness", "strength", "expectedInfluence", "bridgeStrength", "bridgeCloseness", "bridgeBetweenness", "bridgeExpectedInfluence")
`nodes`	Specific nodes for centrality tests. Can be character names or index numbers. Only used if test.centrality=TRUE
`communities`	Passed to bridge() if computing bridge centrality
`useCommunities`	Passed to bridge() if computing bridge centrality
`estimator`	A function that takes data as input and returns a network structure. This can be used for custom estimation algorithms. Note, supplying this function will overwrite the arguments `binary.data`, `AND`, `gamma` and `make.positive.definite`.
`estimatorArgs`	Arguments to the `estimator` function
`verbose`	Logical: Should some warnings and notes be printed?

Value

NCT returns a 'NCT' object that contains the following items:

`glstrinv.real`	The difference in global strength between the networks of the observed data sets.
`glstrinv.perm`	The difference in global strength between the networks of the permutated data sets.
`glstrinv.sep`	The global strength values of the individual networks
`glstrinv.pval`	The p value resulting from the permutation test concerning difference in global strength.
`nwinv.real`	The value of the maximum difference in edge weights of the observed networks
`nwinv.perm`	The values of the maximum difference in edge weights of the permuted networks
`nwinv.pval`	The p value resulting from the permutation test concerning the maximum difference in edge weights.
`einv.pvals`	p-values (corrected for multiple testing or not according to 'p.adjust.methods') per edge from the permutation test concerning differences in edges weights. Only returned if test.edges = TRUE.

`edges.tested`	The pairs of variables between which the edges are called to be tested. Only if test.edges = TRUE.
`einv.real`	The value of the difference in edge weight of the observed networks (multiple values if more edges are called to test). Only if test.edges = TRUE.
`einv.perm`	The values of the difference in edge weight of the permuted networks. Only if test.edges = TRUE.
`diffcen.real`	The values of the difference in centralities of the observed networks. Only if test.centrality = TRUE.
`diffcen.perm`	The values of the difference in centralities of the permuted networks. Only if test.centrality = TRUE.
`diffcen.pval`	p-values(corrected for multiple testing or not according to 'p.adjust.methods') per node from the permutation test concerning differences in centralities. Only if test.centrality = TRUE.

Note

See also my website: http://cvborkulo.com

Author(s)

Claudia D. van Borkulo, with contributions from Jonas Haslbeck, Sacha Epskamp, Payton Jones and Alex Millner

Maintainer: Claudia D. van Borkulo <cvborkulo@gmail.com>

References

Ernst, M.D. Permutation methods: A basis for exact inference. Stat Sci. 2004;19(4):676-685.

Good, P.I. Permutation, parametric and bootstrap tests of hypotheses. Vol. 3. New York:: Springer, 2005.

van Borkulo, C. D., Boschloo, L., Borsboom, D., Penninx, B. W. J. H., Waldorp, L. J., & Schoevers, R.A. (2015). Association of symptom network structure with the course of depression. JAMA Psychiatry. 2015;72(12). doi:10.1001/jamapsychiatry.2015.2079

van Borkulo, C. D., Boschloo, Kossakowski, J., Tio, P., L., Schoevers, R.A., Borsboom, D., & , Waldorp, L. J. (2016). Comparing network structures on three aspects: A permutation test. doi:10.13140/RG.2.2.29455.38569

Examples

```
library("IsingSampler")
library("IsingFit")

### Simulate binary datasets under null hypothesis:
### underlying network structures have the same strength
# Input:
N <- 6 # Number of nodes
nSample <- 500 # Number of samples
```

```
# Ising parameters:
Graph <- matrix(sample(0:1,N^2,TRUE,prob = c(0.8, 0.2)),N,N) * runif(N^2,0.5,2)
Graph <- pmax(Graph,t(Graph))
diag(Graph) <- 0
Thresh <- -rowSums(Graph) / 2

# Simulate:
data1 <- IsingSampler(nSample, Graph, Thresh)
data2 <- IsingSampler(nSample, Graph, Thresh)
colnames(data1) <- colnames(data2) <- c('V1', 'V2', 'V3', 'V4', 'V5', 'V6')

### Compare networks of data sets using NCT ###
# with gamma = 0.
# Iterations (it) set to 10 to save time.
# Low number of iterations can give unreliable results. Should be 1000 at least.

# Testing the three aspects that are validated (network invariance, global strength, edge weight)
# 2 edges are tested here: between variable 1 and 2,
# and between 3 and 6 (can be list(c(2,1),c(6,3)) as well)
Res_1 <- NCT(data1, data2, gamma=0, it=10, binary.data = TRUE,
test.edges=TRUE, edges=list(c(1,2),c(3,6)))

## Plotting of NCT results
## See the help file of plot.NCT for more information about the plotting function and its arguments

# Plot results of the network structure invariance test (not reliable with only 10 permutations!):
plot(Res_1, what="network")

# Plot results of global strength invariance test (not reliable with only 10 permutations!):
plot(Res_1, what="strength")

# Plot results of the edge invariance test (not reliable with only 10 permutations!):
# Note that two distributions are plotted
plot(Res_1, what="edge")

# Without testing for (an) individual edge(s)
# The arguments 'test.edges' and 'edges' don't need to be specified
# Not run
# Res_0 <- NCT(data1, data2, gamma=0, it=10, binary.data = TRUE)
```

Index