

Package: NetworkComparisonTest (via r-universe)

August 27, 2024

Type Package

Title Statistical Comparison of Two Networks Based on Several Invariance Measures

Version 2.2.2

Maintainer Claudia van Borkulo <cvborkulo@gmail.com>

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

Suggests networktools

Description This permutation based hypothesis test, suited for several types of data supported by the estimateNetwork function of the bootnet package (Epskamp & Fried, 2018), assesses the difference between two networks based on several invariance measures (network structure invariance, global strength invariance, edge invariance, several centrality measures, etc.). Network structures are estimated with l1-regularization. The Network Comparison Test is suited for comparison of independent (e.g., two different groups) and dependent samples (e.g., one group that is measured twice). See van Borkulo et al. (2021, in press; the final article will be available, upon publication, via its DOI: 10.1037/met0000476).

License GPL-2

RoxygenNote 6.1.1

Repository <https://vandenman.r-universe.dev>

RemoteUrl <https://github.com/vandenman/networkcomparisonstest>

RemoteRef HEAD

RemoteSha fc2dd7e7c4636f7c382cca4a703148d1dc1b7f56

Contents

NetworkComparisonTest-package	2
NCT-methods	3
NetworkComparisonTest	3

NetworkComparisonTest-package
Statistical Comparison of Two Networks Based on Several Invariance Measures

Description

This permutation based hypothesis test, suited for several types of data supported by the estimateNetwork function of the bootnet package (Epskamp & Fried, 2018), assesses the difference between two networks based on several invariance measures (network structure invariance, global strength invariance, edge invariance, several centrality measures, etc.). Network structures are estimated with l1-regularization. The Network Comparison Test is suited for comparison of independent (e.g., two different groups) and dependent samples (e.g., one group that is measured twice).

Details

Package: NetworkComparisonTest
Type: Package
Version: 2.2.2
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

- Epskamp, S., & Fried, E. I. (2018). A tutorial on regularized partial correlation networks. *Psychological Methods*, 23(4), 617-634.
- 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., van Bork, R., Boschloo, Kossakowski, J., Tio, P., L., Schoevers, R.A., Borsboom, D., & Waldorp, L. J. (2021). Comparing network structures on three aspects: A permutation test. *Psychological Methods*, in press.

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 Several Invariance Measures*

Description

This permutation based hypothesis test, suited for several types of data supported by the estimateNetwork function of the bootnet package (Epskamp & Fried, 2018), assesses the difference between two networks based on several invariance measures (network structure invariance, global strength invariance, edge invariance, several centrality measures, etc.). Network structures are estimated with l1-regularization. The Network Comparison Test is suited for comparison of independent (e.g., two different groups) and dependent samples (e.g., one group that 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.centraliity=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. This argument is ignored when using estimateNetwork() output as input for NCT.
paired	Logical. Can be TRUE or 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 or FALSE to indicate whether the networks to be compared should be weighted or not. If not, the estimated networks are dichotomized. Defaults to TRUE.
AND	Logical. Can be TRUE or 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 or 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 <code>make.positive.definite = TRUE</code> , 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.centralities	Logical. Should centrality metrics be compared across networks?
centrality	Type of centrality metrics to test. Can be any of <code>c("all", "closeness", "betweenness", "strength", "expectedInfluence", "bridgeStrength", "bridgeCloseness", "bridgeBetweenness", "bridgeExpectedInfluence")</code>
nodes	Specific nodes for centrality tests. Can be character names or index numbers. Only used if <code>test.centralities=TRUE</code>
communities	Passed to <code>bridge()</code> if computing bridge centrality
useCommunities	Passed to <code>bridge()</code> 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 <code>binary.data</code> , <code>AND</code> , <code>gamma</code> and <code>make.positive.definite</code> .
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:

<code>glstrinv.real</code>	The difference in global strength between the networks of the observed data sets.
<code>glstrinv.perm</code>	The difference in global strength between the networks of the permuted data sets.
<code>glstrinv.sep</code>	The global strength values of the individual networks
<code>glstrinv.pval</code>	The p value resulting from the permutation test concerning difference in global strength.
<code>nwinv.real</code>	The value of the maximum difference in edge weights of the observed networks
<code>nwinv.perm</code>	The values of the maximum difference in edge weights of the permuted networks
<code>nwinv.pval</code>	The p value resulting from the permutation test concerning the maximum difference in edge weights.

<code>einv.pvals</code>	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 <code>test.edges = TRUE</code> .
<code>einv.real</code>	The value of the difference in edge weight of the observed networks (multiple values if more edges are called to test). Only if <code>test.edges = TRUE</code> .
<code>einv.perm</code>	The values of the difference in edge weight of the permuted networks. Only if <code>test.edges = TRUE</code> .
<code>diffcen.real</code>	The values of the difference in centralities of the observed networks. Only if <code>test.centrality = TRUE</code> .
<code>diffcen.perm</code>	The values of the difference in centralities of the permuted networks. Only if <code>test.centrality = TRUE</code> .
<code>diffcen.pval</code>	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 <code>test.centrality = TRUE</code> .

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., van Bork, R., Boschloo, Kossakowski, J., Tio, P., L., Schoevers, R.A., Borsboom, D., & , Waldorp, L. J. (2021). Comparing network structures on three aspects: A permutation test. DOI: 10.1037/met0000476

Examples

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

### Simulate binary datasets under null hypothesis:
### underlying network structures are similar
# Input:
N <- 6 # Number of nodes
```

```

nSample <- 500 # Number of samples

# Ising parameters:
set.seed(123)
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))
Graph[4,1] <- Graph[4,1]*-1
Graph[1,4] <- Graph[1,4]*-1
Graph[5,1] <- Graph[5,1]*-1
Graph[1,5] <- Graph[1,5]*-1
Graph[6,1] <- Graph[6,1]*-1
Graph[1,6] <- Graph[1,6]*-1
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 ###
## Networks can be compared by either (1) feeding the data directly into NCT (whereby
## you need to specify arguments such as "gamma" and "binary.data") or (2) by using
## estimateNetwork() (bootnet package) and feeding that output into NCT. For the latter
## option, we refer to the help file of estimateNetwork() for its usage. Below, both
## options are illustrated. We recommend using estimateNetwork(), since this function
## has implemented many network estimation methods.

## gamma = 0 (in estimateNetwork this hyperparameter is called "tuning"; to illustrate
## how to specify a different value than the default)
## iterations (it) set to 10 to save time
## Note: Low number of iterations can give unreliable results; should be 1000 at least

## Testing whether there are differences in 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)

## (1) Feeding data directly into NCT
set.seed(123)
NCT_a <- NCT(data1, data2, gamma=0, it=10, binary.data = TRUE,
             test.edges=TRUE, edges=list(c(1,2),c(3,6)))
summary(NCT_a)
## Plot results of global strength invariance test (not reliable with only 10
## permutations!)
plot(NCT_a, what="strength")

## (2) Feeding the estimateNetwork() output into NCT
est_1 <- estimateNetwork(data1, default = "IsingFit", tuning = 0)
est_2 <- estimateNetwork(data2, default = "IsingFit", tuning = 0)
## When using estimateNetwork() output, there is no need to specify gamma and binary.data
## This yields similar output as NCT_a
set.seed(123)

```

```
NCT_b <- NCT(est_1, est_2, it=10, test.edges=TRUE,
edges=list(c(1,2),c(3,6)))
summary(NCT_b)

## Next, an example of testing whether there are differences in node strength
# when data is paired (e.g., a group which is measured pre- and post-treatment).
# Also, here you can see how to specify that you want to take the sign of node strength
# into account (by default, the absolute value is taken and, therefore, the sign is
# ignored).

## abs = FALSE
set.seed(123)
NCT_c = NCT(est_1, est_2, paired = TRUE, abs = FALSE, test.edges = TRUE,
edges = list(c(1,2),c(3,6)), test.centralities = TRUE,
centralities = c("strength"), nodes = "all", it=10)
summary(NCT_c)

## Finally, an example how to test for differences in centrality (e.g., expectedInfluence)

set.seed(123)
NCT_d = NCT(est_1, est_2, paired = TRUE, abs = FALSE, test.edges = TRUE,
edges = list(c(1,2),c(3,6)), test.centralities = TRUE,
centralities = c("expectedInfluence"), nodes = "all", it=10)
summary(NCT_d)
```


Index

NCT, [3](#)
NCT (NetworkComparisonTest), [3](#)
NCT-methods, [3](#)
NetworkComparisonTest, [3](#)
NetworkComparisonTest-package, [2](#)

plot.NCT (NCT-methods), [3](#)
print.NCT (NCT-methods), [3](#)

summary.NCT (NCT-methods), [3](#)