Package ‘SemNeT’

March 22, 2020

Title  Methods and Measures for Semantic Network Analysis
Version  1.2.0
Date  2020-03-21
Maintainer  Alexander P. Christensen <alexpaulchristensen@gmail.com>
Description  Implements several functions for the analysis of semantic networks including partial node bootstrapping (Kenett, Anaki, & Faust, 2014 <doi:10.3389/fnhum.2014.00407>), random walk simulation (Kenett & Austerweil, 2016 <http://alab.psych.wisc.edu/papers/files/Kenett16CreativityRW.pdf>), and a function to compute global network measures. Significance tests and plotting features are also implemented.
Depends  R (>= 3.5.0)
License  GPL (>= 3.0)
Encoding  UTF-8
LazyData  true
Imports  lsa, foreach, parallel, doParallel, pbapply, NetworkToolbox, SemNetCleaner, dplyr, plyr, RColorBrewer, purrr, magrittr, ggplot2, grid, igraph, qgraph, networktools
URL  https://github.com/AlexChristensen/SemNeT
BugReports  https://github.com/AlexChristensen/SemNeT/issues
NeedsCompilation  no
Suggests  knitr, markdown, htmlTable, gridExtra, shiny, shinyalert
VignetteBuilder  knitr
RoxygenNote  7.1.0
Author  Alexander P. Christensen [aut, cre]
         (<https://orcid.org/0000-0002-9798-7037>),
         Yoed N. Kenett [aut, ctb] (<https://orcid.org/0000-0003-3872-7689>)
Repository  CRAN
Date/Publication  2020-03-22 02:40:02 UTC
SemNeT-package

R topics documented:

SemNeT-package .................................................. 2
animals.freq ......................................................... 3
compare.nets ......................................................... 3
convert2cytoscape .................................................. 5
net.high .............................................................. 6
net.low .............................................................. 7
one.result ........................................................... 7
open.binary .......................................................... 8
partboot ............................................................... 8
partboot.test .......................................................... 10
plot.partboot .......................................................... 12
randnet.test ........................................................... 13
randwalk ............................................................. 14
semnetmeas ............................................................ 15
SemNeTShiny ........................................................... 16
sim.fluency ............................................................. 16
similarity ............................................................... 17
two.result .............................................................. 18
vignette.plots .......................................................... 19

Index 20

SemNeT-package

Description

Implements several functions for the analysis of semantic networks including partial node boot-straping (Kenett, Anaki, & Faust, 2014), random walk simulation (Kenett & Austerweil, 2016), and a function to compute global network measures. Significance tests and plotting features are also implemented.

Author(s)

Alexander P. Christensen <alexpaulchristensen@gmail.com> & Yoed N. Kenett <yoedkenett@gmail.com>

References

animals.freq

See Also

Useful links:

- https://github.com/AlexChristensen/SemNeT
- Report bugs at https://github.com/AlexChristensen/SemNeT/issues

---

animals.freq  Frequency of Animal Responses

Description

Frequency of animal responses from Christensen & Kenett (2019). These frequencies are used to generate data in the sim.fluency function.

Usage

data(animals.freq)

Format

animals.freq (vector, length = 367)

Examples

data("animals.freq")

---

compare.nets  Plots Networks for Comparison

Description

Uses qgraph and MDSnet to plot networks. Accepts any number of networks and will organize the plots in the number of side-by-side columns using the heuristic of taking the square root of the number of input and rounding down to the nearest integer (i.e., floor(sqrt(length(input)))).

Examples

- 3 networks: 1 x 3
- 6 networks: 2 x 3
- 9 networks: 3 x 3
Usage

```r
compare.nets(
  ..., 
  title, 
  config, 
  placement = c("match", "default"), 
  weighted = FALSE, 
  qgraph.args = list()
)
```

Arguments

- `...`: Matrices or data frames of network adjacency matrices
- `title`: List. Characters denoting titles of plots
- `config`: Character. Defaults to `MDSnet`. See `qgraph` for more options
- `placement`: Character. How should nodes be placed when comparing groups? Defaults to "default"
  - "match" places nodes in the same position for all networks
  - "default" places nodes in the default `config` positions
- `weighted`: Boolean. Should networks be plotted with weights? Defaults to FALSE. Set to TRUE to plot networks with weights corresponding to association strength. Often, unweighted networks are more aesthetically representational of the networks
- `qgraph.args`: List. An argument list to be passed onto `qgraph`. See `qgraph` for possible arguments

Value

Plots networks using `qgraph` or `MDSnet`

Author(s)

Alexander Christensen <alexpaulchristensen@gmail.com>

References


Examples

# Simulate Datasets
one <- sim.fluency(10)
two <- sim.fluency(10)

# Compute similarity matrix
cos1 <- similarity(one, method = "cosine")
cos2 <- similarity(two, method = "cosine")

# Compute networks using NetworkToolbox
net1 <- NetworkToolbox::TMFG(cos1)$A
net2 <- NetworkToolbox::TMFG(cos2)$A

# Compare networks
compare.nets(net1, net2, title = list("One", "Two"), config = "spring")

# Change edge colors
compare.nets(net1, net2, title = list("One", "Two"),
config = "spring", qgraph.args = list(edge.color = "blue"))

---

**convert2cytoscape**  
*Convert Adjacency Matrix to Cytoscape Format*

**Description**

Converts an adjacency matrix to Cytoscape's sparse matrix format

**Usage**

`convert2cytoscape(A)`

**Arguments**

- **A**  
  Matrix or data frame. A cleaned, finalized response matrix ready to be visualized

**Value**

A sparse matrix formatted for Cytoscape

**Author(s)**

Alexander Christensen <alexpaulchristensen@gmail.com>

**References**

Examples

# Simulate Datasets
one <- sim.fluency(10)
two <- sim.fluency(10)

# Compute similarity matrix
cos1 <- similarity(one, method = "cosine")
cos2 <- similarity(two, method = "cosine")

# Compute networks using NetworkToolbox
net1 <- NetworkToolbox::TMFG(cos1)$A
net2 <- NetworkToolbox::TMFG(cos2)$A

# Convert to Cytoscape format
cyto1 <- convert2cytoscape(net1)
cyto2 <- convert2cytoscape(net2)

# Write to .csv
write.csv(cyto1, file.path(tempdir(), "cyto1.csv"), row.names = FALSE)
write.csv(cyto2, file.path(tempdir(), "cyto2.csv"), row.names = FALSE)

net.high

High Openness to Experience Network

Description

High openness to experience network from Christensen & Kenett (2019)

Usage

data(net.high)

Format

net.high (matrix, 160 x 160)

References


Examples

data("net.high")
net.low

---

**Description**

Low openness to experience network from Christensen & Kenett (2019)

**Usage**

data(net.low)

**Format**

net.low (matrix, 160 x 160)

**References**


**Examples**

data("net.low")

---

one.result

---

**Description**

A result of partboot from a simulated dataset

**Usage**

data(one.result)

**Format**

one.result (list, length = 4)

**Examples**

data("one.result")
open.binary  Binary Response Matrices (Openness and Verbal Fluency)

Description

Binary response matrices for the Animals verbal fluency data \((n = 516)\) from Christensen et al. (2018).

Usage

data(open.binary)

Format

open.binary (matrix, 516 x 367)

References


Examples

data("open.binary")

data("open.binary")

partboot  Partial Bootstrapped Semantic Network Analysis

Description

Bootstraps (without replacement) the nodes in the network and computes global network characteristics

Usage

partboot(..., percent = 0.5, sim, weighted = FALSE, iter = 1000, cores)

Arguments

...  Matrices or data frames. Binary response matrices (e.g., binary output from textcleaner)

percent  Numeric. Percent of nodes to remain in the network. Defaults to .50

sim  Character. Similarity measure to use. Defaults to "cosine". See similarity for other options
partboot

weighted  Boolean. Should weighted ASPL and CC be used? Defaults to FALSE. Set to TRUE for weighted ASPL and CC

iter  Numeric. Number of iterations in bootstrap. Defaults to 1000

cores  Numeric. Number of computer processing cores to use for bootstrapping samples. Defaults to \( n / 2 \) total number of cores. Set to any number between 1 and maximum amount of cores on your computer (see parallel::detectCores())

Value

Returns a list containing:

dataMeas  A matrix for the network input in the data argument, where columns are the semantic network measures from semnetmeas and rows are their values from each bootstrapped sample (results in a matrix with the dimensions iter by 3)

dataSumm  Summary statistics across the bootstrapped samples for the network input in the data argument

percent  Outputs the percent used from the percent argument

iter  Outputs the number of bootstrapped samples used from the iter argument

If a paired network is input, then also returns:

pairedMeas  A matrix for the network input in the paired argument, where columns are the semantic network measures from semnetmeas and rows are their values from each bootstrapped sample (results in a matrix with the dimensions iter by 3)

pairedSumm  Summary statistics across the bootstrapped samples for the network input in the paired argument

Author(s)

Alexander Christensen <alexpaulchristensen@gmail.com>

Examples

# Simulate Dataset
one <- sim.fluency(20)

# Run partial bootstrap networks
one.result <- partboot(data = one, percent = .50, iter = 1000, sim = "cosine", cores = 2)
partboot.test  Statistical tests for partboot

Description
Computes statistical tests for partial bootstrapped networks from partboot. Automatically computes t-tests (t.test) or ANOVA (aov) including Tukey’s HSD for pairwise comparisons (TukeyHSD).

Usage
partboot.test(..., formula = NULL, groups = NULL)

Arguments
... Object(s) from partboot
formula Character. A formula for specifying an ANOVA structure. The formula should have the predictor variable as "y" and include the names the variables are grouped by (e.g., formula = "y ~ group_var1 * group_var2"). See Two-way ANOVA example in examples
groups Data frame. A data frame specifying the groups to be input into the formula. The column names should be the variable names of interest. The groups should be in the same order as the groups input into partboot.

Value
Returns a list containing the objects:

- ASPL Test statistics for each percentage of nodes remaining for ASPL
- CC Test statistics for each percentage of nodes remaining for CC
- Q Test statistics for each percentage of nodes remaining for Q

If two groups:
A matrix in each object has the following columns:

- t-statistic Statistic from the t.test
- df Degrees of freedom
- p-value p-value with values equal to 0 being p < .001
- d Cohen’s d
- CI95.lower Lower bound of the 95 percent confidence interval
- CI95.upper Upper bound of the 95 percent confidence interval
- Direction Direction of the effect. The argument groups will specify specifically which group is higher or lower on the measure. If no groups are input, then "d" and "p" are used to represent data and paired samples from partboot, respectively
Row names refer to the percentage of nodes remaining in bootstrapped networks
If three or more groups:
A list containing two objects:

**ANOVA**
A matrix containing the $F$-statistic, group degrees of freedom, residual degrees of freedom, $p$-value, and partial eta squared $p.\ eta.\ sq$

**HSD**
A matrix containing the differences between each group (diff), lower (lwr) and upper (upr) bounds of the 95% confidence interval, and the adjusted $p$-value (p.adj)

Author(s)
Alexander Christensen <alexpaulchristensen@gmail.com>

Examples

```r
# Simulate Dataset
one <- sim.fluency(20)
two <- sim.fluency(20)

# Run partial bootstrap networks
two.result <- partboot(one, two, percent = .50, iter = 1000, sim = "cosine", cores = 2)

# Compute tests
partboot.test(two.result)
```

```r
# Two-way ANOVA example
## Simulated data
hihi <- sim.fluency(50, 500)
hilo <- sim.fluency(50, 500)
lohi <- sim.fluency(50, 500)
lolo <- sim.fluency(50, 500)

## Create groups
hihi.group <- cbind(rep("high",nrow(hihi)),rep("high",nrow(hihi)))
hilo.group <- cbind(rep("high",nrow(hiolo)),rep("low",nrow(hiolo)))
lohi.group <- cbind(rep("low",nrow(lohi)),rep("high",nrow(lohi)))
lolo.group <- cbind(rep("low",nrow(lolo)),rep("low",nrow(lolo)))

## Bind groups into single data frame
groups <- rbind(hihi.group,
                hilo.group,
                lohi.group,
                lolo.group)

## Change column names (variable names)
colnames(groups) <- c("gf","caq")
```

```r
# Change groups into data frame
```
groups <- as.data.frame(groups)

## Run partial bootstrap networks
boot.fifty <- partboot(hihi, hilo, lohi, lolo, percent = .50)
boot.sixty <- partboot(hihi, hilo, lohi, lolo, percent = .60)

## Compute tests
partboot.test(boot.fifty, boot.sixty, formula = "y ~ gf*caq", groups = groups)

---

**plot.partboot**  
*Plot for partboot*

**Description**

Bootstraps (without replacement) the nodes in the network and computes global network characteristics.

**Usage**

```r
## S3 method for class 'partboot'
plot(..., groups = NULL, measures = c("ASPL", "CC", "Q"))
```

**Arguments**

- `...`: Object(s) from `partboot`
- `groups`: Character. Labels for groups in the order they were entered in `partboot`
- `measures`: Character. Measures to be plotted

**Value**

Returns plots for the specified measures

**Author(s)**

Alexander Christensen <alexpaulchristensen@gmail.com>

**Examples**

```r
# Simulate Dataset
one <- sim.fluency(20)

# Run partial bootstrap networks
one.result <- partboot(one, percent = .50, iter = 1000,
                       sim = "cosine", cores = 2)

# Plot
plot(one.result, groups = c("One"))
```
Description

Performs significance tests for global measures of semantic networks against the global measures of equivalent size (and density) random networks.

Usage

```
randnet.test(..., iter, cores)
```

Arguments

- `...`: Matrices or data frames. Semantic networks to be compared against random networks.
- `iter`: Numeric. Number of iterations in bootstrap. Defaults to 1000.
- `cores`: Number of computer processing cores to use for bootstrapping samples. Defaults to `n-1` total number of cores. Set to any number between 1 and maximum amount of cores on your computer.

Value

Returns a matrix containing p-values for the network measures of the input networks against the distribution of equivalent random networks. The last two columns contain the mean ("M.rand") and standard deviation ("SD.rand") of the network measures for the random network distribution.

Author(s)

Alexander Christensen <alexpaulchristensen@gmail.com>

Examples

```
# Simulate Datasets
one <- sim.fluency(10)

# Compute similarity matrix
cos <- similarity(one, method = "cosine")

# Compute networks using NetworkToolbox
net <- NetworkToolbox::TMFG(cos)$A

# Perform random networks test
randnet.test(net, iter = 1000, cores = 2)
```
Description

Simulates random walks over two networks to examine the characteristics of spontaneous spreading activation (see Kenett & Austerweil, 2016)

Usage

randwalk(A, B, reps = 20, steps = 10, iter = 10000, short.res = TRUE, cores)

Arguments

A  Matrix or data frame. Adjacency matrix of a semantic network
B  Matrix or data frame. A comparison adjacency matrix of a semantic network
reps Numeric. Number of repetitions of increments in 10 steps. Defaults to 20
steps Numeric. Number of random steps to begin with. Defaults to 10
iter Numeric. Number of iterations for each random walk
short.res Boolean. Should shorten results (p-values only) be produced? Defaults to TRUE
cores Numeric. Number of computer processing cores to use for bootstrapping samples. Defaults to $n - 1$ total number of cores. Set to any number between 1 and maximum amount of cores on your computer

Value

A result matrix containing the means and standard deviations for several measures as well as $p$-values for a Mann-Whitney U test

Author(s)

Alexander Christensen <alexpaulchristensen@gmail.com> and Yoed Kenett <yoedkenett@gmail.com>

References


Examples

# Simulate Datasets
one <- sim.fluency(10)
two <- sim.fluency(10)

# Compute similarity matrix
cos1 <- similarity(one, method = "cosine")
cos2 <- similarity(two, method = "cosine")

# Compute networks using NetworkToolbox
net1 <- NetworkToolbox::TMFG(cos1)$A
net2 <- NetworkToolbox::TMFG(cos2)$A

# Run random walk analysis
rw.results <- randwalk(net1, net2, iter = 1000, cores = 2)

---

semnetmeas | Semantic Network Measures
---|---

### Description
Computes the average shortest path length (ASPL), clustering coefficient (CC), and modularity (Q) of the network

### Usage
semnetmeas(A, meas = c("ASPL", "CC", "Q"), weighted = FALSE)

### Arguments
- **A**: Matrix or data frame. An adjacency matrix of a network
- **meas**: Character. Global network measures to compute. By default, computes ASPL, CC, and Q. Individual measures can be selected
- **weighted**: Boolean. Should weighted measures be computed? Defaults to FALSE. Set to TRUE for weighted measures

### Value
Returns a values for ASPL, CC, and Q

### Author(s)
Alexander Christensen <alexpaulchristensen@gmail.com>

### Examples
# Simulate Datasets
one <- sim.fluency(10)

# Compute similarity matrix
cos <- similarity(one, method = "cosine")

# Compute networks using NetworkToolbox
net <- NetworkToolbox::TMFG(cos)$A

# Compute global network measures
globmeas <- semnetmeas(net)

---

**SemNeTShiny**

**SemNeTShiny**

**Description**

An interactive Shiny application for running SemNeT analysis.

**Usage**

```r
SemNeTShiny()
```

**Details**

Start SemNeTShiny

This starts the Shiny application for SemNeT

**Value**

Nothing

**Examples**

```r
## Not run:
SemNeTShiny()
## End(Not run)
```

---

**sim.fluency**

**Simulates a verbal fluency binary response matrix**

**Description**

Simulates verbal fluency data based on the number of nodes in the desired network. The summed total of each response is simulated from a poisson distribution (see `rpois`), using frequencies from the `animals.freq` data. Using these sums, participants responses are simulated with a probability of giving a response as the total of the summed response over the number of participants.

**Usage**

```r
sim.fluency(nodes, cases, random = FALSE)
```
**Arguments**

- **nodes**: Numeric. Number of nodes to simulate in data. Defaults to 100
- **cases**: Numeric. Number of participants to simulate in data. Defaults to 500
- **random**: Boolean. Should the frequencies be randomly sampled from? Defaults to FALSE.

**Value**

A binary matrix with p (participants) by n (nodes)

**Author(s)**

Alexander Christensen <alexpaulchristensen@gmail.com>

**Examples**

```r
# Simulate data for 50 nodes and 200 participants
sim.fluency(nodes = 50, cases = 200)
```

---

**similarity**

*Measures of Similarity*

**Description**

Computes several measures of similarity (see Choi, Cha, & Tappert, 2010 for additional measures)

**Usage**

```r
similarity(
  data, 
  method = c("angular", "cor", "cosine", "euclid", "faith", "jaccard", "phi", "rr")
)
```

**Arguments**

- **data**: Matrix or data frame. A binarized dataset of verbal fluency or linguistic data
- **method**: Character. Type of similarity measure to compute.

Below are the definitions for each bin:

```
1  0
1 a b a+b (R1)
0 c d c+d (R2)
   a+c b+d a+b+c+d (N)
(C1) (C2) (N)
```
Options include:

- "angular" = 1 − (2 * \( \text{acos(cosinesimilarity)} \)/π)
- "cosine" = a/√((a+b)(a+c))
- "faith" = a + 0.5d/a + b + c + d
- "jaccard" = a/a + b + c
- "phi" and "cor" = \( \frac{ad - bc}{\sqrt{(R1xR2xC1xC2)}} \)
- "rr" = a/a + b + c + d

Value

A symmetric similarity matrix

Author(s)

Alexander Christensen <alexpaulchristensen@gmail.com>

References


Examples

```r
# Simulate Datasets
one <- sim.fluency(10)

# Compute similarity matrix
cos <- similarity(one, method = "cosine")
```

---

two.result

**Simulated Result for Dataset One and Two**

Description

A result of `partboot` from two simulated datasets

Usage

data(two.result)

Format

two.result (list, length = 6)

Examples

data("two.result")
vignette.plots

Plots for Vignette

Description

Plots for vignette taken from Christensen & Kenett (2019)

Usage

data(vignette.plots)

Format

vignette.plots (list, length = 3)

References


Examples

data("vignette.plots")
Index

*Topic CTT
SemNeTShiny, 16

*Topic datasets
animals.freq, 3
net.high, 6
net.low, 7
one.result, 7
open.binary, 8
two.result, 18
vignette.plots, 19

animals.freq, 3, 16
aov, 10

compare.nets, 3
convert2cytoscape, 5

MDSnet, 3, 4
net.high, 6
net.low, 7
one.result, 7
open.binary, 8

partboot, 7, 8, 10, 12, 18
partboot.test, 10
plot.partboot, 12

qgraph, 3, 4

randnet.test, 13
randwalk, 14
rpois, 16

SemNeT, 16
SemNeT (SemNeT-package), 2
SemNeT-package, 2
semnetmeas, 9, 15
SemNeTShiny, 16
sim.fluency, 3, 16

similarity, 8, 17
t.test, 10
textcleaner, 8
TukeyHSD, 10
two.result, 18

vignette.plots, 19