Package ‘cssTools’

June 15, 2016

Type Package
Title Cognitive Social Structure Tools
Version 1.0
Date 2016-06-04
Author Deniz Yenigun, Gunes Ertan, Michael Siciliano
Maintainer Deniz Yenigun <deniz.yenigun@bilgi.edu.tr>
Description A collection of tools for estimating a network from a random sample of cognitive social structure (CSS) slices. Also contains functions for evaluating a CSS in terms of various error types observed in each slice.
License GPL (>= 2)
Depends sna
Imports graphics
NeedsCompilation no
Repository CRAN
Date/Publication 2016-06-15 11:42:58

R topics documented:

  cssTools-package ......................................................... 2
  atm ................................................................. 3
  cssTools2sna .......................................................... 4
  ftm ................................................................. 5
  highTechManagers ....................................................... 7
  rtm ................................................................. 8
  rtmPlot ............................................................. 9
  s14 ................................................................. 10
  sliceQuality .......................................................... 12
  sna2cssTools .......................................................... 13

Index 15
**cssTools-package  Cognitive Social Structure Tools**

### Description

Formalized by Krackhardt (1987), Cognitive Social Structure (CSS) network studies collect relational data on respondents direct ties and their cognition of ties among all other individuals in the network. This package provides a collection of tools for estimating a network from a random sample of CSS slices. The package also contains functions for evaluating a CSS in terms of various error types observed in each slice.

### Details

The DESCRIPTION file:

- **Package:** cssTools
- **Type:** Package
- **Title:** Cognitive Social Structure Tools
- **Version:** 1.0
- **Date:** 2016-06-04
- **Author:** Deniz Yenigun, Gunes Ertan, Michael Siciliano
- **Maintainer:** Deniz Yenigun <deniz.yenigun@bilgi.edu.tr>
- **Description:** A collection of tools for estimating a network from a random sample of cognitive social structure (CSS) slices.
- **License:** GPL (>= 2)
- **Depends:** sna
- **Imports:** graphics

### Index of help topics:

- `atm` Estimate a Network Using the Adaptive Threshold Method
- `cssTools-package` Cognitive Social Structure Tools
- `cssTools2sna` Convert a CSS in 'cssTools' Format to a CSS in 'sna' Format
- `ftm` Aggregate CSS Slices for a Fixed Threshold
- `highTechManagers` High Tech Managers Data Set
- `rtm` Estimate a Network Using the ROC Based Threshold Method
- `rtmPlot` Plots for the ROC Based Threshold Method for Estimating Networks
- `s14` Calculate s14 Similarity Index
- `sliceQuality` Evaluate Several Characteristics of Slices from a CSS
- `sna2cssTools` Convert a CSS in 'sna' Format to a CSS in 'cssTools' Format
Author(s)

Deniz Yenigun, Gunes Ertan, Michael Siciliano
Maintainer: Deniz Yenigun <deniz.yenigun@bilgi.edu.tr>

References


See Also

atm, cssTools2sna, ftm, highTechManagers, rtm, rtmPlot, s14, sliceQuality, sna2cssTools

atm Estimate a Network Using the Adaptive Threshold Method

Description

Estimate a network of interest by aggregating the sampled CSS slices using the adaptive threshold method. This requires setting a tolerable level of type 1 error.

Usage

atm(d, sampled, alpha)

Arguments

d Sampled CSS slices in cssTools package format.
sampled A vector indicating which network individuals are sampled.
alpha Tolerable type 1 error.

Details

Given a random sample of observed CSS slices and a tolerable type 1 error, the atm function uses the adaptive threshold method (ATM) of Siciliano et. al. (2012) to aggregate the observed slices and provides an estimate for the network of interest.

Value

estimatedNetwork An estimate of the network of interest.
threshold The threshold value required to reach the given type 1 error rate.
Author(s)
Deniz Yenigun, Gunes Ertan, Michael Siciliano

References

See Also
ftm, rtm

Examples
# Consider the example in Siciliano et al. (2012), # a network with five actors A, B, C, D, E
sA=matrix(c(0,0,1,0,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), nrow=5, ncol=5, byrow=TRUE)
sB=matrix(c(0,1,0,0,0,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), nrow=5, ncol=5, byrow=TRUE)
sC=matrix(c(0,1,1,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), nrow=5, ncol=5, byrow=TRUE)
sD=matrix(c(0,0,1,0,1,0,0,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), nrow=5, ncol=5, byrow=TRUE)
sE=matrix(c(0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0), nrow=5, ncol=5, byrow=TRUE)
d=array(dim=c(5,5,5))
d[,1]=sA
d[,2]=sB
d[,3]=sC
d[,4]=sD
d[,5]=sE

# Suppose you randomly sampled A, D, and E
sampled=c(1,4,5)

dSampled=d[,sampled]

# For a given alpha value, say 0.2, we can combine these slices as follows,
# which gives an estimate of the complete network
atm(dSampled,sampled,0.2)
Arguments

d A CSS in cssTools package format.

Details

In cssTools package, a CSS d is coded in a three dimensional array such that $d[i, j]$ is the $i$-th slice. In sna package, the same object is coded in a three dimensional array such that $d[i, j]$ is the $i$-th slice. The cssTools2sna function transforms cssTools format to sna format.

Value

The same CSS coded in sna format.

Author(s)

Deniz Yenigun, Gunes Ertan, Michael Siciliano

See Also

sna2cssTools

Examples

# Consider the example in Siciliano et. al. (2012),
# a network with five actors A, B, C, D, E
sA=matrix(c(0,0,1,0,1,0,1,0,1,0,1,0,1,0,0,0,1,0,0,0),5,5)
sB=matrix(c(0,1,0,0,0,0,1,0,0,0,1,0,0,0,1,0,0,0),5,5)
sC=matrix(c(1,1,0,0,1,0,0,0,1,0,0,0,0,1,0,0,0,0),5,5)
sD=matrix(c(0,0,1,0,1,0,1,1,0,1,0,0,0,0,1,0,0,1),5,5)
sE=matrix(c(0,0,0,0,1,0,0,0,1,0,0,0,0,0,1,0,0,1),5,5)
d=array(dim=c(5,5))
d[,1]=sA
d[,2]=sB
d[,3]=sC
d[,4]=sD
d[,5]=sE

# Here d is coded in cssTools package format
# Switching between sna and cssTools formats
e=cssTools2sna(d)
f=sna2cssTools(e)

Description

Estimate a network of interest by aggregating the sampled CSS slices for a fixed threshold.
Usage

`ftm(d, sampled, k)`

Arguments

d	Sampled CSS slices in `cssTools` package format.
sampled	A vector indicating which network individuals are sampled.
k	A threshold for aggregating the CSS slices.

Details

Given a random sample of observed CSS slices and a fixed threshold value \( k \) for aggregation, the `ftm` function aggregates the observed slices and provides an estimate for the network of interest by using the fixed threshold method (FTM) given in Yenigun et. al. (2016). The function also returns the estimated type 1 and type 2 errors.

Value

- **estimatedNetwork**: An estimate of the network of interest.
- **type1Error**: Estimated type 1 error rate.
- **type2Error**: Estimated type 2 error rate.
- **type1Count**: Total number of type 1 errors committed.
- **type1Instances**: Number of instances for a potential type 1 error. In other words, number of zeros in the knowledge region of the true network. Here by knowledge region we mean the ties in the network such that both actors are sampled, and the tie is estimated by the intersection of the self reports from both actors. Note that `type1Error` equals `type1Count` divided by `type1Instances`.
- **type2Count**: Total number of type 2 errors committed.
- **type2Instances**: Number of instances for a potential type 2 error. In other words, number of ones in the knowledge region of the true network. Note that `type2Error` equals `type2Count` divided by `type2Instances`.

Author(s)

Deniz Yenigun, Gunes Ertan, Michael Siciliano

References


See Also

- `atm`, `rtm`
Examples

# Consider the example in Siciliano et. al. (2012),
# a network with five actors A, B, C, D, E
sA=matrix(c(0,0,1,0,1,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0),5,5)
sB=matrix(c(0,1,0,0,0,1,0,1,0,0,0,0,0,0,0,0,1,1,0,0,0),5,5)
sC=matrix(c(0,1,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0),5,5)
sD=matrix(c(0,0,1,0,1,0,0,1,1,0,1,0,0,0,0,1,1,0,0,0,0),5,5)
sE=matrix(c(0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,1,1,0,1,0,0),5,5)
d=array(dim=c(5,5,5))
d[,1]=sA
d[,2]=sB
d[,3]=sC
d[,4]=sD
d[,5]=sE

# Suppose you randomly sampled A, D, and E
# sampled=c(1,4,5)

# Then all you have is the following three sampled slices of A, D and E
dSample=d[, sampled]

# For a given threshold, say 2, we can combine these slices as follows,
# which gives an estimate of the complete network
ftm(dSample, sampled, 2)

Description

Krackhardt (1987) reports the CSS data collected from 21 managers in a high tech machinery firm. Perceptions of all individuals on the whole network is provided.

Usage

data(highTechManagers)

Format

A 21 by 21 by 21 array of zeroes (nonexistence of tie) and ones (existence of tie), where the perception slice of the i-th individual corresponds to highTechManagers[, , i].

Details

In a CSS data set, each actor not only reports his or her self-ties, but also answers questions on all possible ties in the network. Then a CSS for a network involving N individuals may be represented by a three dimensional array $R_{i,j,m}$ ($i,j,m = 1,...,N$), where i is the sender, j is the receiver, and m is the perceiver of the relationship. This data set contains the CSS given in Krackhardt (1987), which reports the perceptions of all individuals in a network of 21 managers in a high tech
machinery firm. In the original data 17th slice is problematic since row 17 in this slice consists of ones only. To overcome this, we replaced row 17 with column 17.

References


Examples

data(highTechManagers)
sliceQuality(highTechManagers)

---

rtm

Estimate a Network Using the ROC Based Threshold Method

Description

Estimate a network of interest by aggregating the sampled CSS slices using the ROC based threshold method.

Usage

```r
rtm(d, sampled)
```

Arguments

- `d` Sampled CSS slices in cssTools package format.
- `sampled` A vector indicating which network individuals are sampled.

Details

Given a random sample of observed CSS slices, the `rtm` function uses the density weighted ROC based threshold method (RTM) of Yenigun et. al. (2016) to aggregate the observed slices, and provides an estimate for the network of interest. Slice densities are computed by the `gden` function in the sna package.

Value

- `estimatedNetwork` An estimate of the network of interest.
- `type1Error` Estimated type 1 error rate at the optimum threshold returned by the density weighted ROC method.
- `type2Error` Estimated type 2 error rate at the optimum threshold returned by the density weighted ROC method.
- `threshold` The optimum threshold value.
- `details` A table giving the details of the density weighted ROC method. Columns indicate the threshold, type 1 error (false positive rate), type 2 error, true positive rate (1 - type 2 error), type 1 error count, type 2 error count, and distance.
Author(s)

Deniz Yenigun, Gunes Ertan, Michael Siciliano

References


See Also

atm, ftm

Examples

# Consider the example in Siciliano et al. (2012),
# a network with five actors A, B, C, D, E
sA=matrix(c(0,0,1,0,1,0,0,1,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0),5,5)
sB=matrix(c(0,1,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0),5,5)
sC=matrix(c(0,1,1,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0),5,5)
sD=matrix(c(0,0,1,0,1,0,0,1,1,0,1,0,0,0,0,0,0,0,0,1,0,0,0),5,5)
sE=matrix(c(0,0,1,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1,0,1,0,1,0,1,0),5,5)
d=array(dim=c(5,5,5))
d[,1]=sA
d[,2]=sB
d[,3]=sC
d[,4]=sD
d[,5]=sE

# Suppose you randomly sampled A, D, and E
sampled=c(1,4,5)

dSampled=d[,sampled]

# Then all you have is the following three sampled slices of A, D and E
dSampled=d[,sampled]

# We can combine these slices as follows,
# which gives an estimate of the complete network
rtm(dSampled,sampled)
Arguments

rtmOutput: Output from the function rtm.

Details

The function rtm uses the density weighted ROC based threshold method (RTM) of Yenigun et al. (2016) for estimating networks from a random sample of CSS slices. The output from rtm is visualized by the function rtmPlot, which displays the ROC curve, as well as the type 1 and type 2 error counts for each threshold value.

Author(s)

Deniz Yenigun, Gunes Ertan, Michael Siciliano

References


See Also

rtm

Examples

# Load the highTechManagers data given in cssTools package
data(highTechManagers)

# There are 21 CSS slices in the complete data
# Suppose we only observed the 10 slices with the following indexes
sampled=c(2,4,5,8,9,10,11,14,18,19)

# Then the observed data is the following
dSampled=highTechManagers[, , sampled]

# Apply the ROC based threshold method to estimate the network
y=rtm(dSampled, sampled)

# Now plot the ROC curve and the error types for various threshold values
rtmPlot(y)

s14

Calculate s14 Similarity Index

Description

Computes the $S_{14}$ similarity index between two network matrices.
Usage

`s14(d1, d2)`

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>d1</td>
<td>An $n$ by $n$ matrix representing a network.</td>
</tr>
<tr>
<td>d2</td>
<td>An $n$ by $n$ matrix representing a network.</td>
</tr>
</tbody>
</table>

Details

Given two networks of interest, a common measure of similarity is the $S_{14}$ index introduced by Gower and Legendre (1986). The function `s14` computes this similarity measure for two networks having the same dimensions.

Value

The $S_{14}$ similarity index.

Author(s)

Deniz Yenigun, Gunes Ertan, Michael Siciliano

References


See Also

`sliceQuality`

Examples

```r
# Consider two matrices representing networks, d1 and d2
d1 = matrix(c(0,0,1,0,1,0,0,1,0,0,1,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0),5,5)
d2 = matrix(c(0,1,0,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0),5,5)

# The similarity index between d1 and d2
s14(d1,d2)
```
sliceQuality  

*Evaluate Several Characteristics of Slices from a CSS*

**Description**

Given a fully observed CSS, this function evaluates the quality of each slice by comparing them with the true network obtained by LAS intersection.

**Usage**

sliceQuality(d)

**Arguments**

- **d**  
  A CSS in cssTools package format.

**Details**

A common way of defining a true network for a given CSS is the LAS intersection (see, for example, Siciliano et. al. 2012, or Krackhardt, 1987). For a given CSS, the function sliceQuality first computes the true network by LAS intersection, and then compares each slice with the true network. The considered quantities are matching zeros, matching ones, type 1 errors, type 2 errors, $S_{14}$ similarity index, error proportion and correlation.

**Value**

- **trueNetwork**  
  The true network obtained by LAS intersection method.
- **sliceQuality**  
  A table summarizing the quality of each CSS slice in rows. Columns indicate A (matching zeros), B (0 in CSS slice, 1 in true matrix, i.e., type 2 error), C (1 in CSS slice, 0 in true network, i.e., type 1 error) D (matching ones), s14 ($S_{14}$ similarity index between the CSS slice and the true network), errorProp (proportion of unmatching cells), and correlation (correlation between the CSS slice and the true network computed by the gcor function in the sna package).

**Author(s)**

Deniz Yenigun, Gunes Ertan, Michael Siciliano

**References**


sna2cssTools

See Also

s14

Examples

# Consider the example in Siciliano et al. (2012),
# a network with five actors A, B, C, D, E
sA=matrix(c(0,0,1,0,0,1,0,1,0,1,0,1,0,0,0,0,0,0,1,0,0,0),5,5)
sB=matrix(c(0,1,0,0,1,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0),5,5)
sC=matrix(c(0,1,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0),5,5)
sD=matrix(c(0,0,1,0,1,0,1,0,0,0,0,1,0,0,0,1,0,0,0,0,0),5,5)
sE=matrix(c(0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,0,0),5,5)
d=array(dim=c(5,5,5))
d[,1]=sA
d[,2]=sB
d[,3]=sC
d[,4]=sD
d[,5]=sE

# Compute the quality of CSS slices
sliceQuality(d)

sna2cssTools Convert a CSS in sna Format to a CSS in cssTools Format

Description

Converts a CSS in sna package format to a CSS in cssTools package format.

Usage

sna2cssTools(d)

Arguments

d A CSS in sna package format.

Details

In sna package, a CSS d is coded in a three dimensional array such that d[,i,] is the i-th slice. In cssTools package, the same object is coded in a three dimensional array such that d[,i] is the i-th slice. The sna2cssTools function transforms sna format to cssTools format.

Value

The same CSS coded in cssTools format.
Author(s)
Deniz Yenigun, Gunes Ertan, Michael Siciliano

See Also
cssTools2sna

tool2cssTools

Examples
# Consider the example in Siciliano et. al. (2012),
# a network with five actors A, B, C, D, E
sA=matrix(c(0,0,1,0,0,1,0,1,1,0,0,0,0,0,1,0,0,0,1,0,0,0,5,5)
sB=matrix(c(0,1,0,0,0,1,0,1,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,5,5)
sC=matrix(c(0,1,0,0,0,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,5,5)
sD=matrix(c(0,1,0,0,1,0,1,1,0,1,0,0,1,0,0,0,1,0,1,0,1,0,1,5,5)
sE=matrix(c(0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,5,5)
d=array(dim=c(5,5,5))
d[,1]=sA
d[,2]=sB
d[,3]=sC
d[,4]=sD
d[,5]=sE

# Here d is coded in cssTools package format
# Switching between sna and cssTools formats
e=cssTools2sna(d)
f=sna2cssTools(e)
Index

*Topic datasets
  highTechManagers, 7
*Topic package
  cssTools-package, 2

atm, 3, 3, 6, 9

cssTools (cssTools-package), 2
cssTools-package, 2
cssTools2sna, 3, 4, 14

ftm, 3, 4, 5, 9

highTechManagers, 3, 7

rtm, 3, 4, 6, 8, 10
rtmPlot, 3, 9

s14, 3, 10, 13
sliceQuality, 3, 11, 12
sna2cssTools, 3, 5, 13