Package ‘social’

July 18, 2017

Type Package
Title Social Autocorrelation
Version 1.0
Date 2017-07-16
Author Tom Pike
Maintainer Tom Pike <tpike@lincoln.ac.uk>
Description A set of functions to quantify and visualise social autocorrelation.
License GPL (>= 2)
Imports Rcpp (>= 0.12.9)
LinkingTo Rcpp
Depends stats, graphics
RoxygenNote 6.0.1
LazyData true
NeedsCompilation yes
Repository CRAN
Date/Publication 2017-07-18 21:43:17 UTC

R topics documented:

social.all.paths .................................................. 2
social.cor.matrix ................................................. 3
social.example1 .................................................. 3
social.example2 .................................................. 4
social.plot ....................................................... 4
social.signal ..................................................... 5

Index 7
social.all.paths  All paths between two nodes

Description
Estimate all the possible paths between two nodes in a simple graph using the stochastic method described by Roberts & Kroese (2007).

Usage
social.all.paths(A, start.node, end.node, max.depth = nrow(A),
       n.pilot = 5000, n.estimate = 10000)

Arguments
A     a (possibly weighted) adjacency matrix.
start.node the index of the vertex from which the paths will be calculated.
end.node the index of the vertex to which the paths will be calculated.
max.depth the maximum length of the paths to the returned.
n.pilot the number of naive paths to generate (see Roberts & Kroese, 2007).
n.estimate the number of paths to generate (see Roberts & Kroese, 2007).

Value
An estimate of all the unique paths between start.node and end.node as an nrow(A)xN matrix, padded with zeros.

References

Examples
# Using the data from Figure 1 in Roberts & Kroese (2007)
A = matrix(c(0,1,0,1,0,
             1,0,0,1,1,
             0,0,0,1,1,
             1,1,1,0,0,
             0,1,1,0,0), nrow=5)
paths = social.all.paths(A, 1, 5)
social.cor.matrix

Social correlation matrix

Description
Calculates the social correlation matrix for a given network.

Usage
social.cor.matrix(A, max.depth = nrow(A), n.pilot = 5000, n.estimate = 10000)

Arguments
- **A**: a (possibly weighted) adjacency matrix.
- **max.depth**: the maximum length of the paths to use.
- **n.pilot**: parameter to be passed to `social.all.paths`.
- **n.estimate**: parameter to be passed to `social.all.paths`.

Value
The calculated social correlation matrix.

Examples
A = matrix(c(0,1,0,1,0, 1,0,0,1,1, 0,0,0,1,1, 1,1,0,0, 0,1,1,0,0), nrow=5)
S = social.cor.matrix(A)

data(social.example1)

social.example1

Example dataset 1

Description
An example dataset for demonstrating the functions available in the social package.

Usage
data(social.example1)
Format

The dataset consists of a list with 3 items: A, a 30x30 adjacency matrix; S, a 30x30 social correlation matrix derived from A using $S = \text{social.cor.matrix}(A, \text{max.depth}=5)$; and social.data, a 30-row data frame containing two columns of numeric data, x and y, and a column of node IDs (node.id, corresponding to the row and column names of A and S).

Examples

```
data(social.example1)
```

---

**social.example2**

**Example dataset 2**

Description

An example dataset for demonstrating the functions available in the social package.

Usage

```
data(social.example2)
```

Format

The dataset consists of a list with 3 items: A, a 30x30 adjacency matrix; S, a 30x30 social correlation matrix derived from A using $S = \text{social.cor.matrix}(A, \text{max.depth}=5)$; and social.data, a 30-row data frame containing two columns of numeric data, x and y, and a column of node IDs (node.id, corresponding to the row and column names of A and S).

Examples

```
data(social.example2)
```

---

**social.plot**

**Social scatterplot**

Description

A plot of social data against its socially lagged values

Usage

```
social.plot(x, S, ...)
```
**social.signal**

**Arguments**
- **x**  
  a numeric vector of social data.
- **S**  
  a social correlation matrix.
- ...  
  further arguments to be passed to `plot`.

**Value**

None

**Examples**

```r
A = matrix(c(0,1,0,1,0, 
             1,0,0,1,1, 
             0,0,0,1,1, 
             1,1,1,0,0, 
             0,1,1,0,0), nrow=5)
S = social.cor.matrix(A)
x = rnorm(nrow(A))
social.plot(x, S, ylim=c(min(x),max(x)), xlab="x", ylab="Socially lagged x")
abline(0, 1, lty=2)
```

---

**social.signal**  
**Social signal**

**Description**

Calculates the social signal for a given variable (essentially just Moran's I, but using the social correlation matrix as the weights)

**Usage**

```r
social.signal(x, S)
```

**Arguments**
- **x**  
  a numeric vector of social data.
- **S**  
  a social correlation matrix.

**Value**

A list containing the computed global social signal (`I.s`), the p-value of a test of the null hypothesis that there is no social autocorrelation under the assumption of normality (`p.value`), and the local social signal for each node (`I.local`).
Examples

A = matrix(c(0,1,0,1,0,
             1,0,1,1,1,
             0,0,0,1,1,
             1,1,1,0,0,
             0,1,1,0,0), nrow=5)
S = social.cor.matrix(A)
x = rnorm(nrow(A))
s = social.signal(x, S)
Index

*Topic **datasets**
  social.example1, 3
  social.example2, 4

plot, 5

social.all.paths, 2, 3
social.cor.matrix, 3
social.example1, 3
social.example2, 4
social.plot, 4
social.signal, 5