Package ‘signnet’

February 13, 2022

Title Methods to Analyse Signed Networks

Version 0.8.0


Depends R (>= 3.2.0)

URL https://github.com/schochastics/signnet

BugReports https://github.com/schochastics/signnet/issues

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports igraph, Rcpp, Matrix

RoxygenNote 7.1.2

Suggests testthat (>= 2.1.0), covr, ggplot2, ggraph, knitr, rmarkdown, ompr, ompr.roi, ROI.plugin.glpk, ROI

LinkingTo Rcpp, RcppArmadillo

VignetteBuilder knitr

NeedsCompilation yes

Author David Schoch [aut, cre] (<https://orcid.org/0000-0003-2952-4812>)

Maintainer David Schoch <david@schochastics.net>

Repository CRAN

Date/Publication 2022-02-13 19:00:02 UTC

R topics documented:

as_adj_complex ................................................................. 2
as_adj_signed ................................................................. 3
as_adj_complex

Convert a signed graph to a complex adjacency matrix

Description

This function returns the adjacency matrix for a signed graph that contains ambivalent ties

Usage

as_adj_complex(g, attr)

Arguments

g    igraph object
attr  edge attribute name that encodes positive ("P"), negative ("N") and ambivalent ("A") ties.
as_adj_signed

Value
complex adjacency matrix

See Also
as_adj_signed

as_adj_signed  Convert a signed graph to a signed adjacency matrix

Description
This function returns the adjacency matrix for a signed graph

Usage
as_adj_signed(g, sparse = FALSE)

Arguments
g  igraph object. Must have a "sign" edge attribute.
sparse  Logical scalar, whether to return the result as a sparse matrix. The Matrix package is required for sparse matrices.

Value
signed adjacency matrix

See Also
as_adj_complex

as_complex_edges  Convert Signed Network to Complex

Description
Convert Signed Network to Complex

Usage
as_complex_edges(g, attr = "type")
Arguments

- **g**: igraph object. Must have a "sign" edge attribute.
- **attr**: new edge attribute name that encodes positive ("P"), negative ("N") and ambivalent ("A") ties.

Value

igraph object

Author(s)

David Schoch

Examples

```r
g <- sample_islands_signed(2,10,1,10)
as_complex_edges(g)
```

---

**as_incidence_complex**  
Complex Incidence Matrix

**Description**

The complex incidence matrix of a signed graph containing ambivalent ties.

**Usage**

```r
as_incidence_complex(g, attr)
```

**Arguments**

- **g**: igraph object.
- **attr**: edge attribute name that encodes positive ("P"), negative ("N") and ambivalent ("A") ties.

**Details**

This function is slightly different than `as_incidence_matrix` since it is defined for bipartite graphs. The incidence matrix here is defined as a \( S \in C^{n,m} \), where \( n \) is the number of vertices and \( m \) the number of edges. Edges \((i,j)\) are oriented such that \( i-j \) and entries are defined as

\[
S_{i(i,j)} = \sqrt{A_{ij}} \\
S_{j(i,j)} = -\sqrt{A_{ji}} if (i,j) is an ambivalent tie \\
S_{j(i,j)} = -A_{ji} \sqrt{A_{ji}} else
\]
as_incidence_signed

Value
a complex matrix

Author(s)
David Schoch

See Also
laplacian_matrix_complex,as_adj_complex

as_incidence_signed  Convert a signed two-mode network to a signed matrix

Description
This function returns the incidence matrix for a signed two-mode network.

Usage
as_incidence_signed(g, sparse = FALSE)

Arguments

- **g**  igraph object (bipartite). Must have a "sign" edge attribute.
- **sparse**  Logical scalar, whether to return the result as a sparse matrix. The Matrix package is required for sparse matrices.

Value
signed incidence matrix

as_signed_proj  convert unsigned projection to signed

Description
convert unsigned projection to signed

Usage
as_signed_proj(g)

Arguments

- **g**  igraph object
as_unsigned_2mode

convert signed two-mode network to unsigned

Value
igraph object

Author(s)
David Schoch

See Also
as_unsigned_2mode

Examples
library(igraph)

# create a simple signed two mode network
el <- matrix(c(1,"a",1,"b",1,"c",2,"a",2,"b"),ncol = 2,byrow = TRUE)
g <- graph_from_edgelist(el,directed = FALSE)
E(g)$sign <- c(1,1,-1,1,-1)
V(g)$type <- c(FALSE,TRUE,TRUE,TRUE,FALSE)

# convert to unsigned two-mode network and project
l <- as_unsigned_2mode(g,primary = TRUE)
p <- bipartite_projection(l,which="true")

# turn the unsigned projection back to a signed network
as_signed_proj(p)
Author(s)
David Schoch

See Also
as_signed_proj

Examples

library(igraph)

# create a simple signed two mode network
el <- matrix(c(1,"a",1,"b",1,"c",2,"a",2,"b"),ncol = 2,byrow = TRUE)
g <- graph_from_edgelist(el,directed = FALSE)
E(g)$sign <- c(1,-1,-1,-1)
V(g)$type <- c(FALSE,TRUE,TRUE,TRUE,FALSE)

# convert to unsigned two-mode network and project
l <- as_unsigned_2mode(g,primary = TRUE)
p <- bipartite_projection(l,which="true")

# turn the unsigned projection back to a signed network
as_signed_proj(p)

avatar

Signed networks from Avatar: The Last Airbender

Description
Allies/Enemy relations from Avatar: The Last Airbender

Usage

avatar

Format

igraph object

Source

scraped from Avatar Wiki (https://avatar.fandom.com/wiki/Category:Characters)
**balance_score**

**balancedness of signed network**

**Description**

Implements several indices to assess the balancedness of a network.

**Usage**

`balance_score(g, method = "triangles")`

**Arguments**

- `g`: signed network.
- `method`: string indicating the method to be used. See details for options.

**Details**

The method parameter can be one of:

- **triangles**: Fraction of balanced triangles. Maximal (=1) if all triangles are balanced.
- **walk**: \[ \frac{\sum \exp(\lambda_i)}{\sum \exp(\mu_i)} \] where \( \lambda_i \) are the eigenvalues of the signed adjacency matrix and \( \mu_i \) of the unsigned adjacency matrix. Maximal (=1) if all walks are balanced.
- **frustration**: The frustration index assumes that the network can be partitioned into two groups, where intra group edges are positive and inter group edges are negative. The index is defined as the sum of intra group negative and inter group positive edges. Note that the problem is NP complete and only an upper bound is returned (based on simulated annealing). Exact methods can be found in the work of Aref. The index is normalized such that it is maximal (=1) if the network is balanced.

**Value**

balancedness score

**Author(s)**

David Schoch

**References**


**Examples**

```r
library(igraph)
g <- graph.full(4)
E(g)$sign <- c(-1,1,1,-1,-1,1)

balance_score(g, method = "triangles")
balance_score(g, method = "walk")
```

---

**complex_walks**  
*Count Walks in complex signed network*

**Description**

Count Walks in complex signed network

**Usage**

```r
complex_walks(g, attr, k)
```

**Arguments**

- **g**: igraph object.
- **attr**: edge attribute that encodes positive ("P"), negative ("N") and ambivalent ("A") ties.
- **k**: integer. length of walks

**Value**

igraph object

**Author(s)**

David Schoch

**Examples**

```r
g <- sample_islands_signed(2,10,1,10)
g <- as_complex_edges(g, attr="type")
complex_walks(g, attr="type", k = 3)
```
count_complex_triangles

**Description**

Counts the number of all possible signed triangles (+++), (++-), (+–) and (—)

**Usage**

```r
count_complex_triangles(g, attr)
```

**Arguments**

- `g` igraph object.
- `attr` edge attribute name that encodes positive ("P"), negative ("N") and ambivalent ("A") ties.

**Value**

counts for all complex triangle types

**Author(s)**

David Schoch

**See Also**

`signed_triangles`

**Examples**

```r
library(igraph)
g <- graph.full(4)
E(g)$type <- c("P","N","A","A","P","N")
count_complex_triangles(g,attr = "type")
```
**count_signed_triangles**

Counts the number of all possible signed triangles (+++), (++-), (+–) and (—)

**Usage**

`count_signed_triangles(g)`

**Arguments**

- `g`  
  igraph object with signed edge attribute

**Value**

counts for all 4 signed triangle types

**Author(s)**

David Schoch

**See Also**

`signed_triangles`

**Examples**

```r
library(igraph)
g <- graph.full(4)
E(g)$sign <- c(-1,1,1,-1,-1,1)
count_signed_triangles(g)
```

---

**cowList**  
Signed networks from Correlates of War

**Description**

51 signed networks of inter state relations

**Usage**

`cowList`
Format

List of igraph objects

Source

http://mrvar.fdv.uni-lj.si/pajek/SVG/CoW/default.htm

References


degree_signed

Signed Degree

Description

several options to calculate the signed degree of vertices

Usage

degree_signed(
  g,
  mode = c("all", "in", "out"),
  type = c("pos", "neg", "ratio", "net")
)

Arguments

g igraph object. Must have a "sign" edge attribute.
mode character string, "out" for out-degree, "in" for in-degree or "all" for undirected networks.
type character string, "pos" or "neg" for counting positive or negative neighbors only, "ratio" for pos/(pos+neg), or "net" for pos-neg.

Value

centrality scores as numeric vector.

Author(s)

David Schoch
**Signed Eigenvector centrality**

**Description**

returns the eigenvector associated with the dominant eigenvalue from the adjacency matrix.

**Usage**

`eigen_centrality_signed(g, scale = TRUE)`

**Arguments**

- `g` igraph object. Must have a "sign" edge attribute.
- `scale` Logical scalar, whether to scale the result to have a maximum score of one. If no scaling is used then the result vector is the same as returned by `eigen()`.

**Details**

Note that, with negative values, the adjacency matrix may not have a dominant eigenvalue. This means it is not clear which eigenvector should be used. In addition it is possible for the adjacency matrix to have repeated eigenvalues and hence multiple linearly independent eigenvectors. In this case certain centralities can be arbitrarily assigned. The function returns an error if this is the case.

**Value**

centrality scores as numeric vector.

**Author(s)**

David Schoch

**References**


**Examples**

```r
library(igraph)
data("tribes")
eigen_centrality_signed(tribes)
```
frustration_exact  

Description

Computes the frustration index of a signed network using linear programming

Usage

frustration_exact(g, ...)

Arguments

g  signed network

...  additional parameters for the ompr solver

Details

The frustration index indicates the minimum number of edges whose removal results in a balance network. The function needs the following packages to be installed: ompr, ompr.roi, ROI, and ROI.plugin.glpk. The function implements the AND model in Aref et al., 2020

Value

list containing the frustration index and the bipartition of nodes

Author(s)

David Schoch

References


ggblock  

Plot Blockmodel matrix

Description

Plot Blockmodel matrix

Usage

ggblock(
  g,
  blocks = NULL,
  cols = NULL,
  show_blocks = FALSE,
  show_labels = FALSE
)

Arguments

g  igraph object. Must have a "sign" edge attribute.
blocks  vector of block membership as obtained, e.g. from signed_blockmodel
cols  colors used for negative and positive ties
show_blocks  logical. Should block borders be displayed? (Default: FALSE)
show_labels  logical. Should node labels be displayed? (Default: FALSE)

Value

ggplot2 object

Author(s)

David Schoch

Examples

## Not run:
library(igraph)
data("tribes")
clu <- signed_blockmodel(tribes,k = 3,alpha=0.5,annealing = TRUE)
ggblock(tribes,clu$membership,show_blocks = TRUE,show_labels = TRUE)

## End(Not run)
**ggs signed**

*Plot a signed or complex network*

---

**Description**

Plot a signed or complex network

**Usage**

```r
ggs signed(g, type = "signed", attr = NULL, edge cols = NULL, weights = FALSE)
```

**Arguments**

- **g**: igraph object. Must have a "sign" edge attribute or an attribute containing "P", "N", "A"
- **type**: character string. either "signed" or "complex"
- **attr**: character string. edge attribute that containing "P", "N", "A" if type="complex"
- **edge cols**: colors used for negative and positive (and ambivalent) ties
- **weights**: logical. If TRUE, weights are computed based on sign. Defaults to FALSE

**Details**

This is a very rudimentary visualization of a signed network. If you are fluent in 'ggraph', you can probably cook up something more sophisticated. The function is thus mostly meant to give a quick overview of the network.

**Value**

ggplot2 object

**Author(s)**

David Schoch

---

**graph_circular_signed**

*circular signed graph*

---

**Description**

circular graph with positive and negative edges.

**Usage**

```r
graph_circular_signed(n, r = 1, pos = 0.1, neg = 0.1)
```
**Arguments**

- \( n \) number of nodes
- \( r \) radius
- \( \text{pos} \) distance fraction between positive edges
- \( \text{neg} \) distance fraction between negative edges

**Value**

igraph graph

**Author(s)**

David Schoch

**Examples**

```r
library(igraph)
graph_circular_signed(n = 50)
```

---

### laplacian_angle

**Angle between Eigenvectors**

**Description**

Computes the angle between eigenvectors of the signed or complex Laplacian.

**Usage**

```r
laplacian_angle(g, type = "sign", ...)
```

**Arguments**

- \( g \) input graph. Must have a sign edge attribute
- \( \text{type} \) string, either "sign" for signed Laplacian or "complex" for complex Laplacian. Defaults to "sign"
- \( \ldots \) additional parameters for Laplacian matrix such as the attribute containing "P", "N", and "A" for the complex Laplacian

**Details**

angle between eigenvectors and zero.

**Value**

a numeric matrix
**Author(s)**

David Schoch

**Examples**

```r
corrected_text <- library(igraph)
g <- sample_islands_signed(3, 10, 5/10, 1)
laplacian_angle(g)
```

---

**laplacian_matrix_complex**

*Complex Graph Laplacian*

---

**Description**

The Laplacian of a signed graph containing ambivalent ties.

**Usage**

```r
laplacian_matrix_complex(g, attr, norm = FALSE)
```

**Arguments**

- `g`: igraph object.
- `attr`: edge attribute name that encodes positive ("P"), negative ("N") and ambivalent ("A") ties.
- `norm`: Whether to calculate the normalized Laplacian. See definitions below.

**Details**

See `laplacian_matrix` of igraph for more details. In the complex case, D is a diagonal matrix containing the absolute values of row sums of the complex adjacency matrix.

**Value**

- a complex matrix

**Author(s)**

David Schoch

**See Also**

- `laplacian_matrix_signed`
Description
The Laplacian of a signed graph.

Usage

```r
laplacian_matrix_signed(g, norm = FALSE, sparse = FALSE)
```

Arguments

- `g` - igraph object. Must have a "sign" edge attribute.
- `norm` - Whether to calculate the normalized Laplacian. See definitions below.
- `sparse` - Logical scalar, whether to return the result as a sparse matrix. The Matrix package is required for sparse matrices.

Details
See `laplacian_matrix` of igraph for more details. In the signed case, D is a diagonal matrix containing the absolute values of row sums of the signed adjacency matrix.

Value

a numeric matrix

Author(s)

David Schoch

Examples

```r
library(igraph)
g <- sample_islands_signed(3, 10, 5/10, 1)
laplacian_matrix_signed(g)
laplacian_matrix_signed(g, norm = TRUE)
```
Description
centrality index for signed networks by Everett and Borgatti

Usage
pn_index(g, mode = c("all", "in", "out"))

Arguments
g  igraph object. Must have a "sign" edge attribute.
mode character string, "out" for out-pn, "in" for in-pn or "all" for undirected networks.

Value
centrality scores as numeric vector.

Author(s)
David Schoch

References
Everett, M. and Borgatti, S. (2014) Networks containing negative ties. Social Networks 38 111-120

Examples
library(igraph)
A <- matrix(c(0, 1, 0, 0, 0, 0, 0, -1, -1, 0, 1, 0, 1, 0, 0, 0, 0, -1, -1, 0, 1, -1, 0, 0, 0, 0, 0, 1, 0, 1, 0, -1, 0, 1, 0, -1, 0, -1, 1, 0, 1, 0, 0, 0, 1, 0, -1, 1, 0, 0, -1, 1, 0, 1, 0, 0, -1, 1, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), 10, 10)
g <- igraph::graph_from_adjacency_matrix(A,"undirected",weighted = "sign")
fn_index(g)
sample_islands_signed

A graph with random subgraphs connected by negative edges

Description
Create a number of Erdos-Renyi random graphs with identical parameters, and connect them with the specified number of negative ties.

Usage

```r
sample_islands_signed(islands.n, islands.size, islands.pin, n.inter)
```

Arguments

- `islands.n`: The number of islands in the graph.
- `islands.size`: The size of the islands in the graph.
- `islands.pin`: The probability of intra-island edges.
- `n.inter`: Number of negative edges between two islands.

Value

A signed igraph graph

Author(s)

David Schoch

Examples

```r
library(igraph)
sample_islands_signed(3, 10, 0.5, 1)
```

signed_blockmodel

Blockmodelling for signed networks

Description

Finds blocks of nodes with intra-positive and inter-negative edges

Usage

```r
signed_blockmodel(g, k, alpha = 0.5, annealing = FALSE)
```
signed_blockmodel

Arguments

- **g**: igraph object. Must have a "sign" edge attribute.
- **k**: number of blocks
- **alpha**: see details
- **annealing**: logical. if TRUE, use simulated annealing (Default: FALSE)

Details

The function minimizes \( P(C) = \alpha N + (1 - \alpha)P \), where \( N \) is the total number of negative ties within plus-sets and \( P \) be the total number of positive ties between plus-sets. This function implements the structural balance model. That is, all diagonal blocks are positive and off-diagonal blocks negative. For the generalized version see signed_blockmodel_general.

Value

numeric vector of block assignments and the associated criterion value

Author(s)

David Schoch

References


Examples

```r
library(igraph)

g <- sample_islands_signed(10,10,1,20)
clu <- signed_blockmodel(g, k = 10, alpha = 0.5)
table(clu$membership)
clu$criterion

# Using simulated annealing (less change of getting trapped in local optima)
data("tribes")
clu <- signed_blockmodel(tribes, k = 3, alpha=0.5, annealing = TRUE)
table(clu$membership)
clu$criterion
```
**signed_blockmodel_general**

*Generalized blockmodelling for signed networks*

**Description**

Finds blocks of nodes with specified inter/intra group ties

**Usage**

```r
signed_blockmodel_general(g, blockmat, alpha = 0.5)
```

**Arguments**

- `g`: igraph object. Must have a "sign" edge attribute.
- `blockmat`: Integer Matrix. Specifies the inter/intra group patterns of ties
- `alpha`: see details

**Details**

The function minimizes $P(C) = \alpha N + (1 - \alpha) P$, where $N$ is the total number of negative ties within plus-sets and $P$ be the total number of positive ties between plus-sets. This function implements the generalized model. For the structural balance version see `signed_blockmodel`.

**Value**

numeric vector of block assignments and the associated criterion value

**Author(s)**

David Schoch

**References**


**Examples**

```r
library(igraph)
# create a signed network with three groups and different inter/intra group ties
g1 <- g2 <- g3 <- graph.full(5)
V(g1)$name <- as.character(1:5)
V(g2)$name <- as.character(6:10)
V(g3)$name <- as.character(11:15)
g <- Reduce("%u%", list(g1, g2, g3))
```
signed_triangles

Description
lists all possible signed triangles

Usage
signed_triangles(g)

Arguments
g  igraph object with signed edge attribute

Value
matrix of vertex ids and the number of positive ties per triangle

Author(s)
David Schoch

See Also
count_signed_triangles

Examples
library(igraph)
g <- graph.full(4)
E(g)$sign <- c(-1,1,1,-1,1)
signed_triangles(g)
trip_census_signed

Description
triad census for signed graphs

Usage
trip_census_signed(g)

Arguments
g

igraph object with signed edge attribute

Value
counts for all 139 signed directed triangle types

Author(s)
David Schoch

Examples
library(igraph)
g <- graph.full(4,directed = TRUE)
E(g)$sign <- c(-1,1,1,-1,-1,1)
trip_census_signed(g)

tribes

Signed network of New Guinean highland tribes

Description
Signed social network of tribes of the Gahuku–Gama alliance structure of the Eastern Central Highlands of New Guinea, from Kenneth Read. The network contains sixteen tribes connected by friendship ("rova") and enmity ("hina").

Usage
tribes

Format
An igraph object
Source

http://vlado.fmf.uni-lj.si/pub/networks/data/ucinet/gama.dat

References

Index

* datasets
    avatar, 7
    cowList, 11
    tribes, 25
    as_adj_complex, 2, 3, 5
    as_adj_signed, 3, 3
    as_complex_edges, 3
    as_incidence_complex, 4
    as_incidence_matrix, 4
    as_incidence_signed, 5
    as_signed_proj, 5, 7
    as_unsigned_2mode, 6, 6
    avatar, 7
    balance_score, 8
    complex_walks, 9
    count_complex_triangles, 10
    count_signed_triangles, 11, 24
    cowList, 11
    degree_signed, 12
    eigen_centrality_signed, 13
    frustration_exact, 14
    ggblock, 15
    ggsigned, 16
    graph_circular_signed, 16
    laplacian_angle, 17
    laplacian_matrix, 18, 19
    laplacian_matrix_complex, 5, 18
    laplacian_matrix_signed, 18, 19
    pn_index, 20
    sample_islands_signed, 21
    signed_blockmodel, 15, 21, 23
    signed_blockmodel_general, 22, 23
    signed_triangles, 10, 11, 24
    triad_census_signed, 25
    tribes, 25