Package ‘signnet’

April 28, 2021

**Title**  Methods to Analyse Signed Networks

**Version**  0.7.1


**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**  7.1.1

**Suggests**  testthat (>= 2.1.0), covr, ggplot2, ggraph, knitr, rmarkdown

**LinkingTo**  Rcpp, RcppArmadillo

**VignetteBuilder**  knitr

**NeedsCompilation**  yes

**Author**  David Schoch [aut, cre]

**Maintainer**  David Schoch <david.schoch@manchester.ac.uk>

**Repository**  CRAN

**Date/Publication**  2021-04-28 13:40:02 UTC

**R topics documented:**

<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>as_adj_complex</td>
<td>2</td>
</tr>
<tr>
<td>as_adj_signed</td>
<td>3</td>
</tr>
<tr>
<td>as_complex_edges</td>
<td>3</td>
</tr>
</tbody>
</table>
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.

Value

complex adjacency matrix
as_adj_signed

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.
**as_incidence_complex**

**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

**Value**
a complex matrix

**Author(s)**
David Schoch
as_incidence_signed

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**

```r
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**

```r
as_signed_proj(g)
```

**Arguments**

- `g`  
  igraph object

**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)

Description

convert signed two-mode network to unsigned

Usage

as_unsigned_2mode(g, primary = TRUE)

Arguments

* g    igraph object. Two-mode network. must have a "sign" edge attribute.

* primary    logical. Which mode to transform

Value

igraph object

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)$sign <- c(1,1,-1,1,-1)
V(g)$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

scrapped 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")
balance_score

Arguments

<table>
<thead>
<tr>
<th>g</th>
<th>signed network.</th>
</tr>
</thead>
<tbody>
<tr>
<td>method</td>
<td>string indicating the method to be used. See details for options</td>
</tr>
</tbody>
</table>

Details

The method parameter can be one of

- **triangles** Fraction of balanced triangles. Maximal (=1) if all triangles are balanced.
- **walk** \( \sum \exp(\lambda_i) / \sum \exp(\mu_i) \) 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**  
`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**  
*count complex triangles*

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

**Usage**
`count_complex_triangles(g, attr)`
count_signed_triangles

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

  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

Description

  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

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

doList

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

Doreian, P. and Mrvar, A. (2015). "Structural Balance and Signed International Relations". *Journal of Social Structure*, 16(2)

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

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
ggblock

References


Examples

library(igraph)
data("tribes")
eigen_centrality_signed(tribes)

---

ggblock  

Plot Blockmodel matrix

Description

Plot Blockmodel matrix

Usage

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

Arguments

g
blocks
cols
show_blocks
show_labels

Value

ggplot2 object

Author(s)

David Schoch
Examples

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

---

**ggsigned**

*Plot a signed or complex network*

**Description**

Plot a signed or complex network

**Usage**

```r
ggsigned(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**

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

**Arguments**

- `n` number of nodes
- `r` radius
- `pos` distance fraction between positive edges
- `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**

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

- **g**: input graph. Must have a sign edge attribute
- **type**: string, either "sign" for signed Laplacian or "complex" for complex Laplacian. Defaults to "sign"
- **...**: 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
library(igraph)
g <- sample_islands_signed(3, 10, 5/10, 1)
laplacian_angle(g)
```

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
   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
   library(igraph)
   g <- sample_islands_signed(3, 10, 5/10, 1)
   laplacian_matrix_signed(g)
   laplacian_matrix_signed(g, norm = TRUE)
pn_index | PN Centrality Index
---|---

**Description**

centrality index for signed networks by Everett and Borgatti

**Usage**

```r
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**


**Examples**

```r
library(igraph)
A <- matrix(c(0, 1, 0, 1, 0, 0, 0, -1, -1, 0, 1, 0, 1, -1, 0, 0, 0, 0, 1, 0, 1, -1, -1, 0, 0, 0, 0, 1, -1, 1, -1, 0, 0, 0, 0, 1, -1, 1, 0, 1, 0, -1, 0, -1, 0, 0, -1, 1, 0, 1, 0, 1, -1, 0, -1, 0, -1, 1, 0, 1, 0, 1, -1, 0, -1, 1, 0, 1, 0, 1, 0, 1, -1, 1, -1, 0, 0, -1, 0, 0, 0, 0, 0, 1, 0, -1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, -1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 10, 10), 10, 10)
g <- igraph::graph_from_adjacency_matrix(A,"undirected",weighted = "sign")
pn_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

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

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

signed_blockmodel(g, k, alpha = 0.5, annealing = FALSE)
Argument

- **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)

# sample a signed graph
set.seed(1)  # for reproducibility
nodes <- list(id = c("A", "B", "C", "D", "E")[rep(1:5, each = 10)], value = c(0, 1, 0, 0, 0)[rep(1:5, each = 10)])
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

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

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)$name <- as.character(11:15)
g <- Reduce("%u%",list(g1,g2,g3))
E(g)$sign <- 1
E(g)$sign[1:10] <- -1
g <- add.edges(g,c(rbind(1:5,6:10)),attr = list(sign=-1))
g <- add.edges(g,c(rbind(1:5,11:15)),attr = list(sign=-1))
g <- add.edges(g,c(rbind(11:15,6:10)),attr = list(sign=1))

# specify the link patterns between groups
blockmat <- matrix(c(1,-1,-1,-1,1,1,-1,1,-1),3,3,byrow = TRUE)
res <- signed_blockmodel_general(g,blockmat,0.5)
res$membership
res$criterion

signed_triangles list 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)
triad_census_signed  

Description
triad census for signed graphs

Usage
triad_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)
triad_census_signed(g)

tribes

Description
Signed network of New Guinean highland tribes

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, 23
  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, 6
  as_unsigned_2mode, 6, 6
  avatar, 7
  balance_score, 7
  complex_walks, 9
  count_complex_triangles, 9
  count_signed_triangles, 10, 22
  cowList, 11
  degree_signed, 11
  eigen_centrality_signed, 12
  ggblock, 13
  ggsigned, 14
  graph_circular_signed, 15
  laplacian_angle, 15
  laplacian_matrix, 16, 17
  laplacian_matrix_complex, 5, 16
  laplacian_matrix_signed, 17, 17
  pn_index, 18
  sample_islands_signed, 19
  signed_blockmodel, 13, 19, 21
  signed_blockmodel_general, 20, 21
  signed_triangles, 10, 11, 22
  triad_census_signed, 23
  tribes, 23