Package: NetSwan (via r-universe)

January 26, 2025

Type Package

Title Network Strengths and Weaknesses Analysis

Version 0.1

Date 2015-10-31

Author Serge Lhomme

Maintainer Serge Lhomme <serge.lhomme@u-pec.fr>

Description A set of functions for studying network robustness,
resilience and vulnerability.

Depends igraph

License GPL (>= 2)

NeedsCompilation no

Date/Publication 2015-11-01 00:38:04

Additional_repositories https://cranhaven.r-universe.dev

Config/pak/sysreqs libglpk-dev libxml2-dev

Repository https://cranhaven.r-universe.dev

RemoteUrl https://github.com/cranhaven/cranhaven.r-universe.dev

RemoteRef package/NetSwan

RemoteSha 278b041a1776194281be0a93027e264cc4ae4aba

RemoteSubdir NetSwan

Contents

NetSwan-package *Network Strengths and Weaknesses Analysis*

Description

A set of functions for studying network robustness, resilience and vulnerability.

Details

The main goal of the 'NetSwan' library is to provide a set of functions to study vulnerability, resilience and robustness of graphs. It depends on 'igraph' package. 'Igraph' graphs have a class used in 'NetSwan'.

Author(s)

Serge Lhomme

Maintainer: Serge Lhomme <serge.lhomme@u-pec.fr>

Examples

```
library(igraph)
library(NetSwan)
elec <- matrix(nc=2, byrow=TRUE, c(11,1, 11,10, 1,2, 2,3, 2,9,
3,4, 3,8, 4,5, 5,6, 5,7, 6,7, 7,8, 8,9, 9,10))
gra<-graph.edgelist(elec, directed=FALSE)

f<-swan_efficiency(gra)
vertex_attr(gra, "efficiency_loss", index = V(gra))<-f
vertex_attr(gra)

f2<-swan_closeness(gra)
bet<-betweenness(gra)
reg<-lm(bet~f2)
summary(reg)

f3<-swan_connectivity(gra)

f4<-swan_combinatory(gra,10)
plot(f4[,1],f4[,5], type='o', col='yellow',xlab="Fraction of nodes removed",
      ylab="Connectivity loss")
lines(f4[,1],f4[,3], type='o', col='red')
lines(f4[,1],f4[,4], type='o', col='orange')
lines(f4[,1],f4[,2], type='o', col='blue')
legend('bottomright',c("Random", "Betweenness", "Degree", "Cascading"),
         lty=c(1,1,1,1), pch=c(1,1,1,1),
         col=c("yellow","blue","red", "orange"))
```

`swan_closeness` *Impact on closeness when a node is removed*

Description

Swan_closeness measures the change in the sum of the inverse of distances between all node pairs when excluding that node.

Usage

```
swan_closeness(g)
```

Arguments

`g`	The graph to analyze. Graph builds with igraph.

Details

Components can be disconnected when a node is removed from a network. Closenness can defined as the inverse of farness, which in turn, is the sum of distances to all other nodes. Nevertheless, as the distance between nodes in disconnected components of a network is infinite, this measure cannot be applied to networks with disconnected components (indeed the sum of distances to all other nodes is infinite). Thus it is possible to define the closeness equation as the sum of inversed distances to all other nodes (instead of the inversed of the sum of distances to all other nodes). So swan_closeness measures the change in the sum of the inverse of distances between all node pairs when excluding that node.

Value

Numeric vector with the swan_closeness values of all the vertices.

Author(s)

Serge Lhomme

References

Lhomme S., 2015, Analyse spatiale de la structure des reseaux techniques dans un contexte de risques, Cybergeo : European Journal of Geography.

Examples

```
library(igraph)
library(NetSwan)
elec <- matrix(nc=2, byrow=TRUE, c(11,1, 11,10, 1,2, 2,3, 2,9,
3,4, 3,8, 4,5, 5,6, 5,7, 6,7, 7,8, 8,9, 9,10))
gra<-graph.edgelist(elec, directed=FALSE)

f2<-swan_closeness(gra)
```

```
bet<-betweenness(gra)
reg<-lm(bet~f2)
summary(reg)
```

swan_combinatory *Error and attack tolerance of complex networks*

Description

Swan_combinatory assesses networks vulnerability, resistance of networks to the removal of nodes, due either to random breakdowns or to intentional attacks.

Usage

```
swan_combinatory(g, k)
```

Arguments

`g`	The graph to analyze. Graph builds with igraph.
`k`	Number of iterations for assessing random error.

Details

Many complex systems display a surprising degree of tolerance against errors. However, error tolerance comes at a high price in that these networks are extremely vulnerable to attacks (that is, to the selection and removal of a few nodes that play a vital role in maintaining the network's connectivity). Swan_combinatory removes nodes one by one, first randomly, then in the decreasing order of their degree or load (betweenness). Finally, swan_combinatory uses a cascading scenario, where the loads (betweenness) are recalculated after each node removed.

Value

Matrices corresponding to the loss of connectivity for different strategies. First column corresponds to the fraction of nodes removed. Second column corresponds to the loss of connectivity induced by betweenness based attack. Third column corresponds to the loss of connectivity induced by degree based attack. Fourth column corresponds to the loss of connectivity induced by cascading attack. Fifth column corresponds to the loss of connectivity induced by random error.

Author(s)

Serge Lhomme

References

Albert R., Jeong H., Barabasi A., 2000, Error and attack tolerance of complex networks, Nature, 406(6794), pp. 378-382.

Examples

```
library(igraph)
library(NetSwan)
elec <- matrix(nc=2, byrow=TRUE, c(11,1, 11,10, 1,2, 2,3, 2,9,
3,4, 3,8, 4,5, 5,6, 5,7, 6,7, 7,8, 8,9, 9,10))
gra<-graph.edgelist(elec, directed=FALSE)

f4<-swan_combinatory(gra,10)
plot(f4[,1],f4[,5], type='o', col='yellow',xlab="Fraction of nodes removed",
       ylab="Connectivity loss")
lines(f4[,1],f4[,3], type='o', col='red')
lines(f4[,1],f4[,4], type='o', col='orange')
lines(f4[,1],f4[,2], type='o', col='blue')
legend('bottomright',c("Random", "Betweenness", "Degree", "Cascading"),
           lty=c(1,1,1,1), pch=c(1,1,1,1),
           col=c("yellow","blue","red", "orange"))
```

`swan_connectivity` *Impact on connectivity when a node is removed*

Description

Swan_connectivity measures the loss of connectivity when excluding a node.

Usage

```
swan_connectivity(g)
```

Arguments

g	The graph to analyze. Graph builds with igraph.

Details

Connectivity loss indices quantify the decrease in the number of relationships between each node when one or several components are removed. Swan_connectivty measures the loss of connectivity when excluding a node.

Value

Numeric vector with the swan_connectivity values of all the vertices.

Author(s)

Serge Lhomme

References

Lhomme S., 2015, Analyse spatiale de la structure des reseaux techniques dans un contexte de risques, Cybergeo : European Journal of Geography

Examples

```
library(igraph)
library(NetSwan)
elec <- matrix(nc=2, byrow=TRUE, c(11,1, 11,10, 1,2, 2,3, 2,9,
3,4, 3,8, 4,5, 5,6, 5,7, 6,7, 7,8, 8,9, 9,10))
gra<-graph.edgelist(elec, directed=FALSE)

f3<-swan_connectivity(gra)
```

`swan_efficiency` *Impact on fareness when a node is removed*

Description

Swan_efficiency measures the change in the sum of distances between all node pairs when excluding that node.

Usage

```
swan_efficiency(g)
```

Arguments

`g`	The graph to analyze. Graph builds with igraph.

Details

The easiest way for studying network vulnerability and identifying most critical components is to assess network disruptions due to a single component breakdown without taking into account cascading failures. This kind of measures can be used to determine the importance of nodes or edges in a graph and may be considered as centrality indices. In this context, one indicator is chosen to assess global network performance. For instance, two common indices are used to assess transportation network performance: closeness accessibility and geographic accessibility. Swan_efficiency is based on geographic accessibility. So, swan_efficiency measures the change in the sum of distances between all node pairs when excluding that node.

Value

Numeric vector with the swan_efficiency values of all the vertices.

Author(s)

Serge Lhomme

References

Lhomme S., 2015, Analyse spatiale de la structure des reseaux techniques dans un contexte de risques, Cybergeo : European Journal of Geography

Examples

```
library(igraph)
library(NetSwan)
elec <- matrix(nc=2, byrow=TRUE, c(11,1, 11,10, 1,2, 2,3, 2,9,
3,4, 3,8, 4,5, 5,6, 5,7, 6,7, 7,8, 8,9, 9,10))
gra<-graph.edgelist(elec, directed=FALSE)

f2<-swan_closeness(gra)
bet<-betweenness(gra)
reg<-lm(bet~f2)
summary(reg)
```

Index