Package ‘wfg’

October 12, 2022

Type Package
Title Weighted Fast Greedy Algorithm
Version 0.1
Date 2016-02-21
Author Han Yu [aut, cre], Rachael Hageman Blair [aut]
Maintainer Han Yu <hyu9@buffalo.edu>
Depends R (>= 3.1.0), igraph
Description Implementation of Weighted Fast Greedy algorithm for community detection in networks with mixed types of attributes.
License GPL (>= 2)
NeedsCompilation no
Repository CRAN
Date/Publication 2016-02-25 00:23:11

R topics documented:

network.simu ................................................. 1
wfg ......................................................... 3

Index

network.simu Simulation of Networks with Community Structures

Description

Simulation of networks under the framework by Girvan and Newman. The vertices are connected with each other randomly and independently with probability p.in (within same community) and p.out (between communities).
Usage

network.simu(nv = c(32, 32, 32, 32),
  p.in = c(0.323, 0.323, 0.323, 0.323),
  p.out = 0.0625, p.del = 0)

Arguments

nv       a vector of community sizes. The number of communities equals the number of elements in this vector.
p.in     a vector of probability of a node to be randomly linked to other nodes in the same community.
p.out    the probability of a node to be randomly linked to nodes in other communities.
p.del    the proportion of links that are randomly deleted.

Value

net      The simulated network.
group    The membership of vertices.

Author(s)

Han Yu & Rachael Hageman Blair

References


Examples

```r
## simulation of a network with four communities, each with size 32
library(wfg)
nv = c(32, 32, 32, 32)
p.in = c(0.452, 0.452, 0.452, 0.452)
p.out = 0.021
p.del = 0
net.simu <- network.simu(nv=nv, p.in=p.in, p.out=p.out, p.del=p.del)
net <- net.simu$net
group <- net.simu$group

## plot simulated network with vertices colored by membership
V(net)$size <- 7
V(net)$color <- group
plot(net, vertex.label='')
```
**Weighted Fast Greedy Algorithm**

**Description**

Implementation of weighted fast greedy algorithm for community detection in networks with mixed types of attributes.

**Usage**

```r
wfg(net, attr=NULL, under.sample=FALSE, prioritize=FALSE)
```

**Arguments**

- `net`: network for community detection
- `attr`: data frame of attribute information. The default value is NULL, when no attribute information will be used. Under default this method is identical to fast greedy community detection algorithm.
- `under.sample`: a boolean parameter. When it is TRUE, the vertex pairs without links will be under-sampled to have the same number as that of the linked pairs of vertices.
- `prioritize`: a boolean parameter. When it is TRUE, a matrix of community-specific coefficients will be returned, by which the communities can be prioritized.

**Details**

Each column of `attr` data frame can be a vector with type of either numeric (continuous) or factor (categorical). The matrix of community-specific coefficients gives estimates as to the relative homogeneity of each attribute within each community. Specifically, a negative beta with large absolute value indicates corresponding attribute is homogeneous.

**Value**

- `beta`: Estimates of coefficients from logistic regression.
- `memb`: Community membership of vertices.

**Author(s)**

Han Yu & Rachael Hageman Blair

**References**

Examples

```
##### implementation of wfg on a computer generated network with
##### structurally relevant continuous attribute and irrelevant categorical attribute
set.seed(7)
##### set network properties
## four groups, each with 32 vertices
nv <- c(32,32,32,32)
l <- length(nv)
## obtain p.in and p.out from z.out
z.out <- 6
z.in < 16-z.out
p.out <- z.out/96
p.in <- rep(z.in/31, l)

##### simulate network
net.simu <- network.simu(nv=nv, p.in=p.in, p.out=p.out, p.del=0)
net <- net.simu$net
group <- net.simu$group

##### simulate attributes
## separation of continuous attribute
delta <- 1
## p's for the multinomial distribution of categorical attributes
p1 <- 0.25
p2 <- (1-p1)/3
attr1 <- c(rnorm(nv[1], 0), rnorm(nv[2], 1*delta), rnorm(nv[3], 2*delta), rnorm(nv[4], 3*delta))
attr2 <- c(sample(c(1,2,3,4), size=nv[1], prob=c(p1, p2, p2, p2), replace=TRUE),
           sample(c(1,2,3,4), size=nv[2], prob=c(p2, p1, p2, p2), replace=TRUE),
           sample(c(1,2,3,4), size=nv[3], prob=c(p2, p2, p1, p2), replace=TRUE),
           sample(c(1,2,3,4), size=nv[4], prob=c(p2, p2, p2, p1), replace=TRUE))
attributes <- data.frame(attr1, attr2)

##### implementation of wfg
wfg.result <- wfg(net=net, attr=attributes, under.sample = FALSE, prioritize = TRUE)

##### plot network colored by wfg result
V(net)$size <- 7
V(net)$color <- wfg.result$memb
plot(net, vertex.label='')
```
Index

network.simu, 1

wfg, 3