Package 'ergm.rank'

June 1, 2022

Version 4.1.0

Date 2022-06-01

Title Fit, Simulate and Diagnose Exponential-Family Models for Rank-Order Relational Data

Depends R (>= 4.0),
ergm (>= 4.2.2),
network (>= 1.15)

Imports statnet.common (>= 4.2.0),
utils

LinkingTo ergm

Suggests covr,
knitr,
rmarkdown

Description A set of extensions for the 'ergm' package to fit weighted networks whose edge weights are ranks. See Krivitsky and Butts (2017) <doi:10.1177/0081175017692623> and Krivitsky, Hunter, Morris, and Klumb (2021) <arXiv:2106.04997>.

License GPL-3 + file LICENSE

URL https://statnet.org

BugReports https://github.com/statnet/ergm.rank/issues

VignetteBuilder rmarkdown, knitr

RoxygenNote 7.2.0

Roxygen list(markdown = TRUE)

Encoding UTF-8

R topics documented:

`ergm.rank-package` *Fit, Simulate and Diagnose Exponential-Family Models for Rank-Order Relational Data*

Description

`ergm.rank` is a set of extensions to package `ergm` to fit and simulate from exponential-family random graph models for networks whose edge weights are ranks. For a list of functions type `help(package='ergm')` and `help(package='ergm.rank')`

Details

Mainly, it implements the `CompleteOrder` reference measure for valued ERGMs (documented here), and provides some rank-order change statistics (documented here).

For a complete list of the functions, use `library(help="ergm")` and `library(help="ergm.rank")` or read the rest of the manual.

When publishing results obtained using this package, please cite the original authors as described in `citation(package="ergm.rank")`.

All programs derived from this package must cite it.

This package contains functions specific to using `ergm` to model networks whose dyad values are ranks. Examples include preferences, valued ties reduced to ranks, etc..

These terms have a specialized interpretation, and are therefore generally prefixed by "`rank.`", though they should take any valued data.

For detailed information on how to download and install the software, go to the Statnet project website: `https://statnet.org`. A tutorial, support newsgroup, references and links to further resources are provided there.

Author(s)

Pavel N. Krivitsky <pavel@statnet.org>

References

Krivitsky PN (2012). Exponential-Family Random Graph Models for Valued Networks. *Electronic Journal of Statistics*, 2012, 6, 1100-1128. c("\Sexpr[results=rd,stage=build]tools:::Rd_expr_doi(\"#1\")", "doi:10.1214/12-EJS696")\ifelse{text}{doi:10.1214/12-EJS696 <https://doi.org/10.1214/12-EJS696>}{\ifelse{latex}{\href{
EJS696}{doi:10.1214\out{\slash{}}12\out{\-}EJS696}}{\href{https://doi.org/10.1214/12-EJS696}{doi:10.1214/12-EJS696}}}

Krivitsky PN and Butts CT (2017). Exponential-Family Random Graph Models for Rank-Order Relational Data. *Sociological Methodology*, 2017, 47, 68-112. doi:10.1177/0081175017692623

See Also

ergm-terms, ergm-references

`AlterSwap-ergmProposal`

A proposal that swaps values of two alters incident on an ego

---

Description

This proposal randomly selects two dyads (i, j) and (i, j') with a common sender and proposes to swap their values if distinct.

Details

This proposal is not referenced in the lookup table.

`CompleteOrder-ergmReference`

A uniform distribution over the possible complete orderings of the alters by each ego

---

Description

A uniform distribution over the possible complete orderings of the alters by each ego

Usage

```
# CompleteOrder
```

See Also

ergmReference for index of reference distributions currently visible to the package.

newcomb *Newcomb's Fraternity Networks*

Description

These 14 networks record weekly sociometric preference rankings from 17 men attending the University of Michigan in the fall of 1956; Data were collected longitudinally over 15 weeks, although data from week 9 are missing.

Format

A list of 15 networks.

Details

The men were recruited to live in off-campus (fraternity) housing, rented for them as part of the Michigan Group Study Project supervised by Theodore Newcomb from 1953 to 1956. All were incoming transfer students with no prior acquaintance of one another.

The data set, derived from one in the unreleased `netdata` package, contains a network list `newcomb` with 14 networks. Each network is complete and contains two edge attributes:

list("rank") the preference of the ith man for the jth man from 1 through 16, with 1 being the highest preference.

list("descrank") the same, but 1 indicates lowest preference.

Licenses and Citation

If the source of the data set does not specified otherwise, this data set is protected by the Creative Commons License https://creativecommons.org/licenses/by-nc-nd/2.5/.

When publishing results obtained using this data set the original authors should be cited. In addition this should be cited as:

Vladimir Batagelj and Andrej Mrvar (2006): *Pajek datasets*
http://vlado.fmf.uni-lj.si/pub/networks/data/

Source

http://vlado.fmf.uni-lj.si/pub/networks/data/ucinet/ucidata.htm#newfrat

References

See the link above. Newcomb T. (1961). The acquaintance process. New York: Holt, Reinhard and Winston.

Nordlie P. (1958). A longitudinal study of interpersonal attraction in a natural group setting. Unpublished doctoral dissertation, University of Michigan.

White H., Boorman S. and Breiger R. (1977). Social structure from multiple networks, I. Blockmodels of roles and positions. American Journal of Sociology, 81, 730-780.

Examples

```
# Note: This takes a long time.
data(newcomb)

# Fit a model for the transition between initial (time 0) ranking and
# ranking after one week (time 1). Note that MCMC interval has been
# decreased to save time.
newcomb.1.2.fit <- ergm(newcomb[[2]]~
                        rank.inconsistency(newcomb[[1]],"descrank")+
                        rank.deference+rank.nonconformity("all")+
                        rank.nonconformity("localAND"),
                        response="descrank", reference=~CompleteOrder,
                        control=control.ergm(MCMC.interval=10))
# Check MCMC diagnostics (barely adequate).
mcmc.diagnostics(newcomb.1.2.fit)
summary(newcomb.1.2.fit)
```

---

rank.deference-ergmTerm *Deference (aversion)*

Description

Measures the amount of "deference" in the network: configurations where an ego i ranks an alter j over another alter k, but j, in turn, ranks k over i . A lower-than-chance value of this statistic and/or a negative coefficient implies a form of mutuality in the network.

Usage

```
# valued: rank.deference
```

---

rank.edgecov-ergmTerm *Dyadic covariates*

Description

Models the effect of a dyadic covariate on the propensity of an ego i to rank alter j highly.

Usage

```
# valued: rank.edgecov(x, attrname)
```

Arguments

x, attrname	either a square matrix of covariates, one for each possible edge in the network, the name of a network attribute of covariates, or a network; if the latter, optional argument attrname provides the name of the quantitative edge attribute to use for covariate values (in this case, missing edges in x are assigned a covariate value of zero).

---

rank.inconsistency-ergmTerm

(Weighted) Inconsistency

Description

Measures the amount of disagreement between rankings of the focus network and a fixed covariate network x , by couting the number of pairwise comparisons for which the two networks disagree.

Usage

```
# valued: rank.inconsistency(x, attrname, weights, wtname, wtcenter)
```

Arguments

x, attrname	x can be a network with an edge attribute attrname containing the ranks or a matrix of appropriate dimension containing the ranks. If x is not given, it defaults to the LHS network, and if attrname is not given, it defaults to the response edge attribute.
weights	optional parameter to weigh the counts. Can be either a 3D $n \times n \times n$ -array whose (i, j, k) th element gives the weight for the comparison by i of j and k or a function taking three arguments, i, j, and k, and returning the weight of this comparison.
wtname, wtcenter	If wtcenter=TRUE , the calculated weights will be centered around their mean. wtname can be used to label this term.

---

rank.nodeicov-ergmTerm

Attractiveness/Popularity covariates

Description

Models the effect of one or more nodal covariates on the propensity of an actor to be ranked highly by the others.

Usage

```
# valued: rank.nodeicov(attr)
```

Arguments

attr	quantitative attribute (see Specifying Vertex attributes and Levels (?nodal_attributes) for details.)

---

rank.nonconformity-ergmTerm *Nonconformity*

Description

Measures the amount of "nonconformity" in the network: configurations where an ego i ranks an alter j over another alter k, but ego l ranks k over j .

Usage

```
# valued: rank.nonconformity(to, par)
```

Arguments

to	which controls to whom an ego may conform: • "all" (the default): Nonconformity to all egos is counted. A lower-than-chance value of this statistic and/or a negative coefficient implies a degree of consensus in the network. • "localAND": Nonconformity of i to ego l regarding the relative ranking of j and k is only counted if i ranks l over both j and k . A lower-than-chance value of this statistic and/or a negative coefficient implies a form of hierarchical transitivity in the network. This is the recommended form of local nonconformity (over "local1" and "local2"). • "local1": Nonconformity of i to ego l regarding the relative ranking of j and k is only counted if i ranks l over j . • "local2": Nonconformity of i to ego l regarding the relative ranking of j and k is only counted if i ranks l over k . • "thresholds": Nonconformity of i to ego l regarding the relative ranking of j and k is only counted if i ranks l above par, where par can be a vector with multiple thresholds. • "geometric": Nonconformity of i to ego l regarding the relative ranking of j and k is weighted by par taken to the power of the rank of l by i , where par is a scalar.
par	additional parameters for some types of nonconformity.

Index