

# Package ‘ergm.rank’

May 15, 2019

**Version** 1.2.0

**Date** 2019-05-15

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

**Depends** ergm ( $\geq 3.10.1$ ), network ( $\geq 1.15$ )

**Imports** statnet.common ( $\geq 4.2.0$ )

**LinkingTo** ergm

## Description

A set of extensions for the 'ergm' package to fit weighted networks whose edge weights are ranks.

**License** GPL-3 + file LICENSE

**URL** <http://statnet.org>

**BugReports** <https://github.com/statnet/ergm.rank/issues>

**NeedsCompilation** yes

**Author** Pavel N. Krivitsky [aut, cre] (<<https://orcid.org/0000-0002-9101-3362>>),  
Carter T. Butts [ctb],  
Mark S. Handcock [ctb],  
David R. Hunter [ctb]

**Maintainer** Pavel N. Krivitsky <[pavel@uow.edu.au](mailto:pavel@uow.edu.au)>

**Repository** CRAN

**Date/Publication** 2019-05-15 07:43:03 UTC

## R topics documented:

ergm.rank-package	2
ergm-references	3
ergm-terms	4
newcomb	5

<b>Index</b>	<b>7</b>
--------------	----------

---

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

---

## Description

`ergm.rank` is a set of extensions to R 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')`

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.

## Details

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..

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

## Author(s)

Pavel N. Krivitsky <[pavel@uow.edu.au](mailto:pavel@uow.edu.au)>

## References

Krivitsky PN (2012). Exponential-Family Random Graph Models for Valued Networks. *Electronic Journal of Statistics*, 2012, 6, 1100-1128. doi: [10.1214/12EJS696](https://doi.org/10.1214/12EJS696)

Krivitsky PN and Butts CT (2012). Exponential-Family Random Graph Models for Rank-Order Relational Data. <http://arxiv.org/abs/1210.0493>

## See Also

[ergm-terms](#), [ergm-references](#)

---

ergm-references	<i>Reference Measures for Exponential-Family Random Graph Models for Rank-Order Relational Data</i>
-----------------	---

---

## Description

This page describes the possible reference measures (baseline distributions) for modeling rank data found in the [ergm.rank](#) package.

Each of these is specified on the RHS of a one-sided formula passed as the reference argument to [ergm](#). See the [ergm](#) documentation for a complete description of how reference measures are specified.

## Known issues

### [MCMLE.trustregion](#):

Because Monte Carlo MLE's approximation to the likelihood becomes less accurate as the estimate moves away from the one used for the sample, [ergm](#) limits how far the optimization can move the estimate for every iteration: the log-likelihood may not change by more than [MCMLE.trustregion](#) control parameter, which defaults to 20. This is an adequate value for binary ERGMs, but because each dyad in a valued ERGM contains more information, this number may be too small, resulting in unnecessarily many iterations needed to find the MLE.

Automatically setting [MCMLE.trustregion](#) is work in progress, but, in the meantime, you may want to set it to a high number (e.g., 1000).

## Possible reference measures to represent baseline distributions

Reference measures currently available are:

[CompleteOrder](#) A uniform distribution over the possible complete orderings of the alters by each ego.

[DiscUnif\(a,b\)](#) A discrete uniform distribution used as a baseline distribution for ranks. See [DiscUnif](#) documentation in the [ergm](#) for arguments. Using `reference=~DiscUnif(1,n-1)` (for network size  $n$ ) can be used to model weak orderings, though this approach is currently untested. Note that it entails a specific assumption about actors' propensity to rank.

## References

Krivitsky PN (2012). Exponential-Family Random Graph Models for Valued Networks. *Electronic Journal of Statistics*, 2012, 6, 1100-1128. doi: [10.1214/12EJS696](https://doi.org/10.1214/12EJS696)

Krivitsky, PN and Butts, CT (2012). Exponential-Family Random Graph Models for Rank-Order Relational Data. <http://arxiv.org/abs/1210.0493>

## See Also

[ergm](#), [network](#), [%v%](#), [%n%](#), [sna](#), [summary.ergm](#), [print.ergm](#)

## Description

This page describes the possible terms (and hence network statistics) included in the [ergm.rank](#) package.

See the [ergm-terms](#) documentation in the `ergm` package for a complete description of what ERGM terms are and how they are used.

## Terms to represent network statistics included in the [ergm.rank](#) package

All terms listed are valued. For their specific forms, see Krivitsky and Butts (2012).

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

`rank.deference` *Deference (aversion)*: 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.

`rank.edgescov(x, attrname)` *Dyadic covariates*: Models the effect of a dyadic covariate on the propensity of an ego  $i$  to rank alter  $j$  highly. See the [edgescov](#) ERGM term documentation for arguments.

`rank.inconsistency(x, attrname, weights, wname, wtcenter)` *(Weighted) Inconsistency*: Measures the amount of disagreement between rankings of the focus network and a fixed covariate network  $x$ , by counting the number of pairwise comparisons for which the two networks disagree.  $x$  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.

Optionally, the count can be weighted by the `weights` argument, which 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. If `wtcenter=TRUE`, the calculated weights will be centered around their mean. `wname` can be used to label this term.

`rank.nodeicov(attrname, transform, transformname)` *Attractiveness/Popularity covariates*: Models the effect of a nodal covariate on the propensity of an actor to be ranked highly by the others. See the [nodeicov](#) ERGM term documentation for arguments.

`rank.nonconformity(to, par)` *Nonconformity*: 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  $j$ .

This statistic has an argument `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  $k$ .

## References

- Handcock M. S., Hunter D. R., Butts C. T., Goodreau S. G., Krivitsky P. N. and Morris M. (2012). `_Fit, Simulate and Diagnose Exponential-Family Models for Networks_`. Version 3.1. Project home page at <URL: <http://www.statnet.org>>, <URL: [CRAN.R-project.org/package=ergm](http://CRAN.R-project.org/package=ergm)>.
- Krivitsky P. N. (2012). Exponential-Family Random Graph Models for Valued Networks. *Electronic Journal of Statistics*, 2012, 6, 1100-1128. doi: [10.1214/12EJS696](https://doi.org/10.1214/12EJS696)
- Krivitsky PN and Butts CT (2012). Exponential-Family Random Graph Models for Rank-Order Relational Data. <http://arxiv.org/abs/1210.0493>

## See Also

[ergm-terms](#) (from the `ergm` package), `ergm`, `network`, `%v%`, `%n%`

---

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.

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:

`rank` the preference of the  $i$ th man for the  $j$ th man from 1 through 16, with 1 being the highest preference.

`descrank` the same, but 1 indicates lowest preference.

## Usage

`data(newcomb)`

## Licenses and Citation

If the source of the data set does not specified otherwise, this data set is protected by the Creative Commons License <http://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:

Mark S. Handcock, David Hunter, Carter T. Butts, Steven M. Goodreau, and Martina Morris. 2003 *statnet: An R package for the Statistical Modeling of Social Networks* <http://www.statnet.org> and the source 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. Block-models 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(MCMLE.trustregion=10000,
    MCMC.interval=10))

# Check MCMC diagnostics (barely adequate).
mcmc.diagnostics(newcomb.1.2.fit)
summary(newcomb.1.2.fit)
```

# Index

## \*Topic **datasets**

newcomb, [5](#)

## \*Topic **models**

ergm-references, [3](#)

ergm-terms, [4](#)

ergm.rank-package, [2](#)

## \*Topic **package**

ergm.rank-package, [2](#)

%n%, [5](#)

%v%, [5](#)

CompleterOrder (ergm-references), [3](#)

DiscUnif, [3](#)

documented here, [2](#)

edgecov, [4](#)

ergm, [2](#), [3](#), [5](#)

ergm-references, [3](#)

ergm-terms, [4](#)

ergm.rank, [2–4](#)

ergm.rank-package, [2](#)

ergm.references (ergm-references), [3](#)

ergm.terms (ergm-terms), [4](#)

InitErgmReference.CompleteOrder  
(ergm-references), [3](#)

InitWtErgmProposal.AlterSwap  
(ergm-references), [3](#)

InitWtErgmTerm.rank.deference  
(ergm-terms), [4](#)

InitWtErgmTerm.rank.edgecov  
(ergm-terms), [4](#)

InitWtErgmTerm.rank.inconsistency  
(ergm-terms), [4](#)

InitWtErgmTerm.rank.nodeicov  
(ergm-terms), [4](#)

InitWtErgmTerm.rank.nonconformity  
(ergm-terms), [4](#)

MCMLE.trustregion, [3](#)

network, [4](#), [5](#)

newcomb, [5](#)

nodeicov, [4](#)

rank.deference (ergm-terms), [4](#)

rank.edgecov (ergm-terms), [4](#)

rank.inconsistency (ergm-terms), [4](#)

rank.nodeicov (ergm-terms), [4](#)

rank.nonconformity (ergm-terms), [4](#)

references-ergm (ergm-references), [3](#)

references.ergm (ergm-references), [3](#)

terms-ergm (ergm-terms), [4](#)

terms.ergm (ergm-terms), [4](#)