Package 'ergm.count'

September 11, 2025

Version 4.1.3

Date 2025-09-10

Title Fit, Simulate and Diagnose Exponential-Family Models for Networks with Count Edges

Depends ergm (>= 4.10.1), network (>= 1.19.0)

Imports statnet.common (>= 4.12.0), Rdpack (>= 2.6.4)

LinkingTo ergm

Suggests covr, knitr, rmarkdown, testthat

RdMacros Rdpack

Description A set of extensions for the 'ergm' package to fit weighted networks whose edge weights are counts. See Krivitsky (2012) <doi:10.1214/12-EJS696> and Krivitsky, Hunter, Morris, and Klumb (2023) <doi:10.18637/jss.v105.i06>.

License GPL-3 + file LICENSE

URL https://statnet.org

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

VignetteBuilder rmarkdown, knitr

RoxygenNote 7.3.2.9000

Encoding UTF-8

Config/testthat/parallel true

Config/testthat/edition 3

NeedsCompilation yes

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

Maintainer Pavel N. Krivitsky <pavel@statnet.org>

Repository CRAN

Date/Publication 2025-09-11 08:10:08 UTC

Contents

`ergm.count-package` *Fit, Simulate and Diagnose Exponential-Family Models for Networks with Count Edges*

Description

ergm.count is a set of extensions to package **ergm** to fit and simulate from exponential-family random graph models for networks whose edge weights are counts (Krivitsky 2012).

Details

Mainly, it implements Poisson, binomial, geometric, and discrete uniform dyadwise reference measures for valued ERGMs (documented here in `ergmReference`), and provides some count-specific change statistics (documented in `ergmTerm`) (Krivitsky 2012; Krivitsky et al. 2023), including `CMP` for the Conway–Maxwell–Poisson Distribution (Shmueli et al. 2005). When publishing results obtained using this package, please cite the original authors as described in `citation(package="ergm.count")`.

All programs derived from this package must cite it.

This package contains functions specific to using `ergm()` to model networks whose dyad values are counts. Examples include counts of conversations, messages, and other interactions.

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.

Known issues

Parameter space constraints:

Poisson- and geometric-reference ERGMs have an unbouded sample space. This means that the parameter space may be constrained in complex ways that depend on the terms used in the model. At this time `ergm` has no way to detect when a parameter configuration had strayed outside of the parameter space, but it may be noticeable on a runtime trace plot (activated via `MCMC.runtime.traceplot` control parameter), when the simulated values keep climbing upwards. (See Krivitsky (2012) for a further discussion.)

A possible remedy if this appears to occur is to try lowering the `control.ergm()` parameter `MCMLE.steplength`.

Author(s)

Maintainer: Pavel N. Krivitsky <pavel@statnet.org> (ORCID)

Other contributors:

- Mark S. Handcock <handcock@stat.ucla.edu> [contributor]
- David R. Hunter <dhunter@stat.psu.edu> [contributor]
- Joyce Cheng <joyce.cheng@student.unsw.edu.au> [contributor]

References

Krivitsky PN (2012). "Exponential-family Random Graph Models for Valued Networks." *Electronic Journal of Statistics*, **6**, 1100–1128. doi:10.1214/12EJS696.

Krivitsky PN, Hunter DR, Morris M, Klumb C (2023). "ergm 4: New Features for Analyzing Exponential-Family Random Graph Models." *Journal of Statistical Software*, **105**(6), 1–44. doi:10.18637/jss.v105.i06.

Shmueli G, Minka TP, Kadane JB, Borle S, Boatwright P (2005). "A Useful Distribution for Fitting Discrete Data: Revival of the Conway–Maxwell–Poisson Distribution." *Journal of the Royal Statistical Society: Series C*, **54**(1), 127–142. ISSN 1467-9876. doi:10.1111/j.14679876.2005.00474.x.

See Also

ergmTerm, ergmReference

`Binomial-ergmReference`

Binomial-reference ERGM

Description

Specifies each dyad's baseline distribution to be binomial with `trials` trials and success probability of 0.5 : $h(y) = \prod_{i,j} \binom{\text{trials}}{y_{i,j}}$. Using valued ERGM terms that are "generalized" from their binary counterparts, with form `"sum"` (see previous link for the list) produces logistic regression.

Usage

```
# Binomial(trials)
```

Arguments

`trails`	model parameter

See Also

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

Keywords: bipartite, directed, discrete, finite, nonnegative, undirected, valued

CMB-ergmTerm *Conway-Maxwell-Binomial Distribution*

Description

If `couple==TRUE` , this term adds one statistic to the model, of the form $\sum_{i,j} \log(y_{i,j}!) + \log(t - y_{i,j}!)$. This turns a Binomial- or a discrete-uniform-reference ERGM into a Conway-Maxwell-Binomial-reference ERGM, allowing it to represent a broad range of disperson values. In particular, combined with a Binomial-reference ERGM, a negative coefficient on this term induces underdispersion and a positive coefficient induces overdispersion.

If `coupled==FALSE` the two summands above are added as their own statistic (each with its own free parameter).

Usage

```
# valued: CMB(trials, coupled = TRUE)
```

Arguments

`trails`	model parameter
`coupled`	logical

See Also

`ergmTerm` for index of model terms currently visible to the package.

Keywords: directed, nonnegative, undirected, valued

CMP-ergmTerm *Conway-Maxwell-Poisson Distribution*

Description

This term adds one statistic to the model, of the form $\sum_{i,j} \log(y_{i,j}!)$. This turns a Poisson- or a geometric-reference ERGM into a Conway-Maxwell-Poisson-reference ERGM, allowing it to represent a broad range of disperson values. In particular, combined with a Poisson-reference ERGM, a negative coefficient on this term induces underdispersion and a positive coefficient induces overdispersion. (This behavior is different from 3.1.1, when the negation of this value was used.)

Usage

```
# valued: CMP
```

Details

Note that its current implementation may not perform well if the data are overdispersed relative to geometric.

See Also

ergmTerm for index of model terms currently visible to the package.

Keywords: directed, nonnegative, undirected, valued

Geometric-ergmReference *Geometric-reference ERGM*

Description

Specifies each dyad's baseline distribution to be uniform on the natural numbers (and 0): $h(y) = 1$. In itself, this "distribution" is improper, but in the presence of sum , a geometric distribution is induced. Using CMP (in addition to sum) induces a Conway-Maxwell-Poisson distribution that is geometric when its coefficient is 0 and Poisson when its coefficient is -1 .

Usage

```
# Geometric
```

See Also

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

Keywords: bipartite, directed, discrete, nonnegative, undirected, valued

---

Poisson-ergmReference *Poisson-reference ERGM*

Description

Specifies each dyad's baseline distribution to be Poisson with mean 1: $h(y) = \prod_{i,j} 1/y_{i,j}!$, with the support of $y_{i,j}$ being natural numbers (and 0). Using valued ERGM terms that are "generalized" from their binary counterparts, with form "sum" (see previous link for the list) produces Poisson regression. Using CMP induces a Conway-Maxwell-Poisson distribution that is Poisson when its coefficient is 0 and geometric when its coefficient is 1 .

@details Three proposal functions are currently implemented, two of them designed to improve mixing for sparse networks. They can can be selected via the MCMC.prop.weights= control parameter. The sparse proposals work by proposing a jump to 0. Both of them take an optional proposal argument p0 (i.e., MCMC.prop.args=list(p0=...)) specifying the probability of such a jump. However, the way in which they implement it are different:

- `"random"`: Select a dyad (i,j) at random, and draw the proposal $y^{\star}_{i,j} \sim \text{Poisson}_{\neq y_{i,j}}(y_{i,j}+0.5)$ (a Poisson distribution with mean slightly higher than the current value and conditional on not proposing the current value).
- `"0inflated"`: As `"random"` but, with probability `p0` , propose a jump to 0 instead of a Poisson jump (if not already at 0). If `p0` is not given, defaults to the "surplus" of 0s in the observed network, relative to Poisson.
- `"TNT"`: (the default) As `"0inflated"` but instead of selecting a dyad at random, select a tie with probability `p0` , and a random dyad otherwise, as with the binary TNT. Currently, `p0` defaults to 0.2.

Usage

```
# Poisson
```

See Also

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

Keywords: bipartite, directed, discrete, nonnegative, undirected, valued

---

zach *Karate club social network of Zachary (1977)*

Description

Zachary (1977) reported observations of social relations in a university karate club, with membership that varied between 50 and 100, of whom 34 individuals: 32 ordinary club members and officers, the club president ("John A."), and the part-time instructor ("Mr. Hi"); consistently interacted outside of the club. Over the course of the study, the club divided into two factions, and, ultimately, split into two clubs, one led by Hi and the other by John and the original club's officers. The split was driven by a disagreement over whether Hi could unilaterally change the level of compensation for his services.

Format

The data are represented as a `network` object, with an edge attribute `contexts`, giving the number of contexts of interaction for that pair of actors. In addition, the following vertex attributes are provided:

`club`**:** the club in which the actor ended up;

`faction`**:** faction alignment of the actor as recorded by Zachary;

`faction.id`**:** faction alignment coded numerically, as -2 (strongly Mr. Hi's), -1 (weakly Mr. Hi's), 0 (neutral), $+1$ (weakly John's), and $+2$ (strongly John's);

`role`**:** role of the actor in the network (Instructor, Member, or President)

Details

Zachary identifies the faction with which each of the 34 actors was aligned and how strongly and reports, for each pair of actors, the count of social contexts in which they interacted. The 8 contexts recorded were

- academic classes at the university;
- Hi's private karate studio in his night classes;
- Hi's private karate studio where he taught on weekends;
- student-teaching at Hi's studio;
- the university rathskeller (bar) located near the karate club;
- a bar located near the university campus;
- open karate tournaments in the area; and
- intercollegiate karate tournaments.

The highest number of contexts of interaction for a pair of individuals that was observed was 7.

Source

Zachary, WW (1977). An Information Flow Model for Conflict and Fission in Small Groups. *Journal of Anthropological Research*, 33(4), 452-473.

Sociomatrix in machine-readable format can be retrieved from https://github.com/bavla/Nets/blob/master/data/Pajek/ucinet/README.md.

References

Zachary, WW (1977). An Information Flow Model for Conflict and Fission in Small Groups. *Journal of Anthropological Research*, 33(4), 452-473.

Examples

```
data(zach)

oldpal <- palette()
palette(gray((1:8) / 8))
plot(zach, vertex.col = "role", displaylabels = TRUE, edge.col = "contexts")
palette(oldpal)


# Fit a binomial-reference ERGM.

zach.fit1 <- ergm(zach ~ nonzero + sum + nodefactor("role", levels = -2) +
                    absdiffcat("faction.id"),
                  response = "contexts", reference = ~Binomial(8))

mcmc.diagnostics(zach.fit1)

summary(zach.fit1)
```

```
## Not run:
# This is much slower.
zach.fit2 <- ergm(zach ~ nonzero + sum + nodefactor("role", levels = -2) +
                    transitiveties,
                  response = "contexts", reference = ~Binomial(8),
                  eval.loglik = FALSE)

mcmc.diagnostics(zach.fit2)

summary(zach.fit2)

## End(Not run)
```

Index