

Package ‘ergMargins’

September 13, 2025

Type Package

Title Process Analysis for Exponential Random Graph Models

Version 1.6.1

Maintainer Scott Duxbury <duxbury@email.unc.edu>

Description Calculates marginal effects and conducts process analysis in exponential family random graph models (ERGM).

Includes functions to conduct mediation and moderation analyses and to diagnose multicollinearity.

URL: <<https://github.com/sduxbury/ergMargins>>.

BugReports: <<https://github.com/sduxbury/ergMargins/issues>>.

Duxbury, Scott W (2021) <[doi:10.1177/0049124120986178](https://doi.org/10.1177/0049124120986178)>.

Long, J. Scott, and Sarah Mustillo (2018) <[doi:10.1177/0049124118799374](https://doi.org/10.1177/0049124118799374)>.

Mize, Trenton D. (2019) <[doi:10.15195/v6.a4](https://doi.org/10.15195/v6.a4)>.

Karlson, Kristian Bernt, Anders

Holm, and Richard Breen (2012) <[doi:10.1177/0081175012444861](https://doi.org/10.1177/0081175012444861)>.

Duxbury, Scott W (2018) <[doi:10.1177/0049124118782543](https://doi.org/10.1177/0049124118782543)>.

Duxbury, Scott W, Jenna Wertsching (2023) <[doi:10.1016/j.socnet.2023.02.003](https://doi.org/10.1016/j.socnet.2023.02.003)>.

Huang, Peng, Carter Butts (2023) <[doi:10.1016/j.socnet.2023.07.001](https://doi.org/10.1016/j.socnet.2023.07.001)>.

License GPL (>= 2)

Encoding UTF-8

Imports numDeriv, stats, ergm, Matrix, btergm, parallel, dplyr, sna, network, sampling

Suggests MASS, knitr, rmarkdown, methods, statnet.common, Rcpp, ergm.count

NeedsCompilation no

Author Scott Duxbury [aut, cre, cph]

Repository CRAN

Date/Publication 2025-09-13 05:10:48 UTC

Contents

edge.prob2 2

ergm.AME	3
ergm.MEM	7
ergm.mma	10
ergm.mod.mma	13
ergm.MSE	16
ergm.msma	19
GW_helper	22
vif.ergm	24
Index	27

edge . prob2	<i>Function to compute tie probabilities for each dyad.</i>
--------------	---

Description

Computes tie probabilities for each dyad. Largely a wrapper for edgeprob() in btergm.

Usage

```
edge . prob2(model , verbose=FALSE)
```

Arguments

model	model is the ergm object.
verbose	Boolean operator. If set to T, updates user on progress of edge.prob2. May be useful for large networks.

Details

Computes tie probabilities for each dyad in the ERGM sample space.This function is largely a wrapper for the edgeprob function in btergm, but can be applied to curved ERGMs.

Value

Returns a dataframe where weach row is a dyad and each column is a term in the model, an identifi-
cation variable for the vertices/dyads, or the estimated probability of observing a tie. See edgeprob()
in btergm for more details.

Author(s)

Scott Duxbury, Associate Professor, University of North Carolina–Chapel Hill, Department of So-
ciology. original author is Philip Leifeld at University of Essex.

See Also

[edgeprob](#)

Examples

```
library(ergm)

data("faux.dixon.high")

my.ergm<-ergm(faux.dixon.high~edges+
              nodeifactor("sex")+
              nodeofactor("sex")+
              nodematch("sex")+
              mutual)
edge.prob2(my.ergm)
```

ergm.AME

Function to compute average marginal effects in ERGM.

Description

ERGM function to compute average marginal effects for main effects or at various levels of a moderator. Also provides tests of significance for second differences for interaction terms. Currently accepts `ergm`, `btergm`, `mtergm`, `ergm.count`, `mlergm`, and `ergm.multi` objects. See `ergm.MEM` to compute marginal effects at means.

Usage

```
ergm.AME(model,
          var1,
          var2=NULL,
          inter=NULL,
          at.2=NULL,
          at.controls=NULL,
          control_vals=NULL,
          return.dydx=FALSE,
          return.at.2=FALSE)
```

Arguments

<code>model</code>	the <code>ergm</code> object to be analyzed.
<code>var1</code>	the character string containing the name of the focal variable on which to compute the average marginal effect. A vector of character strings can be provided matching model output to compute the marginal effect of a linear combination of covariates.
<code>var2</code>	if examining moderation, the character string containing the name of the moderator variable. Default value is <code>NULL</code> .

<code>inter</code>	if examining moderation, the character string containing the name of the interaction between <code>var1</code> and <code>var2</code> . Default value is <code>NULL</code> .
<code>at.2</code>	the levels of <code>var2</code> at which to compute the average marginal effects. If <code>var2=NULL</code> , this parameter is ignored. If <code>var2!=NULL</code> and <code>at.2=NULL</code> , the function computes the AME at each unique value of <code>var2</code> . <code>at.2</code> can also be set to meaningful values, such as the mean of <code>var2</code> or other discrete values of interest. If <code>var2</code> is discrete, leaving <code>at.2</code> <code>NULL</code> will compute the AME for each discrete category of the moderator variable. It is usually safe to leave <code>at.2=NULL</code> unless examining a large network where <code>var2</code> has a large number of unique values, in which case computation times can be restrictive.
<code>at.controls</code>	A character string containing the names of all control variables to be specified at fixed levels when calculating the marginal effect. Should match ERGM output exactly.
<code>control_vals</code>	A vector specifying the values to hold <code>at.controls</code> variables. Ignored if <code>at.controls = NULL</code>
<code>return.dydx</code>	logical parameter indicating whether to return the marginal effects for each dyad. Defaults to <code>FALSE</code> .
<code>return.at.2</code>	logical parameter indicating whether to return the levels of <code>at.2</code> as part of the output list object. Default value is <code>F</code> . Intended primarily for internal use.

Details

Computes the AME for variables specified. If `var2` and `inter` are left at `NULL`, this is the simple AME. All AMEs are computed as tie probabilities (not on the linear scale). Standard errors are estimated using the Delta method. If a moderator is specified using `var2` and `inter`, a table of AMEs is returned at each level of `at.2`. If `at.2=NULL`, this is computed for every unique level of `var2`. So, if `var2` has many levels and the model is complex, run times may be restrictive when `at.2` is left `NULL`.

When `at.2` and `var2` are specified, a table of second differences is also returned. The table of second differences assesses whether the difference in marginal effects is significant between levels of a moderator (values of `at.2`). If the moderator is binary, the second difference tests the overall significance of an interaction. The tests statistic is a Wald test.

If more than 2 values of `at.2` are specified, the function also computes the average absolute Wald test statistic and average second difference. The average second difference measures the average change in the AME of `var1` between levels of a moderator. The average absolute Wald test is the mean absolute value of the Wald tests for all second differences. It assesses whether, on average, there is a significant difference between the second differences and zero. A significant result means the interaction is significant.

Note that when an interaction term is an absolute difference between nodal covariates, `ergm.AME` holds the value of `var1` at its mean. This is because the partial derivative for the absolute difference between covariates treats `var1` to vary, which tells us little about tendencies towards tie variables when two nodal covariates are at similar values. Thus, the mean is used as a representative value at which to compute the AME for absolute differences.

If `at.controls` and `control_vals` are provided, the AME is calculated while holding specified control variables at fixed values provided in `control_vals`, rather than their observed values.

Value

If var2 and inter=NULL, then a table is returned with the AME for var1, its standard error, and test statistic. Note again that all AMEs are in the metric of tie probabilities, not log-odds.

If a moderator is specified, a list is returned with the following elements:

‘Average Marginal Effects‘

is the table of AMEs at each level of at.2. If at.2=NULL, the table contains unique entries for every unique value of at.2.

‘Second differences‘

is a table containing point estimates, standard errors, and Wald tests for the difference in AMEs between each level of at.2. It has a number of rows equal to length(at.2)-1

‘Aggregate output‘

is a table of summary statistics when at.2 has length > 2. Reports the mean second difference and mean absolute Wald statistic for the second differences. If at.2 has length < 3, it is not returned

‘Marginal effects‘

if return.dydx=TRUE, returns the marginal effects (partial derivatives with respect to var1) for each dyad in the ERGM sample space.

Warning

Note that AMEs can take a long time to estimate in large networks or high dimensional models. If AME computation time proves to be restrictive, see the ergm.MEM function, which computes marginal effects at means.

Note

Syntax is loosely based on the margins package by Thomas Leeper and Stata’s margins commands. Currently only available for two-way interactions.

Author(s)

Scott Duxbury, Associate Professor, University of North Carolina–Chapel Hill, Department of Sociology.

References

- Duxbury, Scott W. 2023. "The Problem of Scaling in Exponential Random Graph Models." *Sociological Methods & Research*.
- Duxbury, Scott W., and Jenna Wertsching. 2023. "Scaling Bias in Pooled Exponential Random Graph Models." *Social Networks*.
- Long, J. Scott, and Sarah Mustillo. 2018. "Using Predictions and Marginal Effects to Compare Groups in Regression Models for Binary Outcomes." *Sociological Methods and Research*
- Mize, Trenton D. 2019. "Best Practices for Estimating, Interpreting, and Presenting Nonlinear Interaction Effects." *Sociological Science*

ergm.MEM ergm.mma margins

```
library(ergm)

data("faux.dixon.high")

set.seed(21093)

my.ergm<-ergm(faux.dixon.high~edges+
  nodeicov("grade")+
  nodeocov("grade")+
  nodeifactor("sex")+
  nodeofactor("sex")+
  absdiff("grade")+
  nodematch("sex")+
  mutual)

#main effect
main.eff<-ergm.AME(my.ergm,"mutual")

#main effect of "mutual" when sender and receiver effects for sex == 1
ergm.AME(my.ergm,"mutual",
  at.controls=c("nodeifactor.sex.2","nodeofactor.sex.2"),
  control_vals=c(1,1))

##with moderator
sex.inter<-ergm.AME(my.ergm,"nodeifactor.sex.2","nodeofactor.sex.2","nodematch.sex")

##with moderator at specific values
grade.inter<-ergm.AME(my.ergm,"nodeicov.grade","nodeocov.grade","absdiff.grade",
  at.2=c(9,12))

##when moderator changes from mean to mean +sd
tie.probs<-edge.prob2(my.ergm)
m<-mean(tie.probs$nodeocov.grade)
m.sd<-m+sd(tie.probs$nodeocov.grade)

grade.inter.at.means<-ergm.AME(my.ergm,"nodeicov.grade","nodeocov.grade",
  "absdiff.grade",at.2=c(m,m.sd))
```

ergm.MEM

*Function to compute marginal effects at means in ERGM.***Description**

ERGM function to compute marginal effects at means for main effects or at various levels of a moderator. Also provides tests of significance for second differences for interaction terms. Currently accepts `ergm`, `btergm`, `mtergm`, `ergm.count`, `mlergm`, and `ergm.multi` objects. See `ergm.AME` to compute average marginal effect.

Usage

```
ergm.MEM(model,
          var1,
          var2=NULL,
          inter=NULL,
          at.2=NULL,
          at.controls=NULL,
          control_vals=NULL,
          return.dydx=FALSE)
```

Arguments

<code>model</code>	the <code>ergm</code> object to be analyzed.
<code>var1</code>	the character string containing the name of the focal variable on which to compute the MEM. A vector of character strings can be provided matching model output to compute the marginal effect of a linear combination of covariates.
<code>var2</code>	the character string containing the name of the moderator variable. Default value is <code>NULL</code> .
<code>inter</code>	the character string containing the name of the interaction between <code>var1</code> and <code>var2</code> . Default value is <code>NULL</code> .
<code>at.2</code>	the levels of <code>var2</code> at which to compute the MEMs. If <code>var2=NULL</code> , this parameter is ignored. If <code>var2!=NULL</code> and <code>at.2=NULL</code> , the function computes the AME at each unique value of <code>var2</code> . <code>at.2</code> can also be set to meaningful values, such as the mean of <code>var2</code> or other discrete values of interest. If <code>var2</code> is discrete, leaving <code>at.2 NULL</code> will compute the AME for each discrete category of the moderator variable. It is usually safe to leave <code>at.2=NULL</code> unless examining a large network where <code>var2</code> has a large number of unique values, in which case computation times can be restrictive.
<code>at.controls</code>	A character string containing the names of all control variables to be specified at fixed levels when calculating the marginal effect. Should match ERGM output exactly.

<code>control_vals</code>	A vector specifying the values to hold at.controls variables. Ignored if at.controls = NULL
<code>return.dydx</code>	logical parameter indicating whether to return the marginal effects for each dyad observation. Defaults to FALSE.

Details

Largely a wrapper for `ergm.AME`. The only difference is that marginal effects are computed at variable means.

Computes the MEM for variables specified. If `var2` and `inter` are left at NULL, this is the simple MEM. All MEMs are computed as tie probabilities (not on the linear scale). Standard errors are computed using the Delta method. If a moderator is specified using `var2` and `inter`, a table of MEMs is returned at each level of `at.2`. If `at.2=NULL`, this is computed for every unique level of `var2`.

When `at.2` and `var2` are specified, a table of second differences is also returned. The table of second differences assesses whether the difference in marginal effects is significant between levels of a moderator (values of `at.2`). If the moderator is binary, the second difference tests the overall significance of an interaction. The tests statistic is a Wald test.

If more than 2 values of `at.2` are specified, the function also computes the average absolute Wald test statistic and average second difference. The average second difference measures the average change in the MEM of `var1` between levels of a moderator. The average absolute Wald test is the mean absolute value of the Wald tests for all second differences. It assesses whether, on average, there is a significant difference between the second differences and zero. A significant result means the interaction is significant.

If `at.controls` and `control_vals` are provided, the MEM is calculated while holding specified control variables at fixed values provided in `control_vals`, rather than at their means

Value

If `var2` and `inter=NULL`, then a table is returned with the MEM for `var1`, its standard error, and test statistic. Note again that all MEMs are in the metric of tie probabilities, not log-odds. If a moderator is specified, a list is returned with the following elements:

‘Marginal effects at means‘

is the table of MEMs at each level of `at.2`. If `at.2=NULL`, the table contains unique entries for every unique value of `at.2`.

‘Second differences‘

is a table containing point estimates, standard errors, and Wald tests for the difference in MEMs between each level of `at.2`. It has a number of rows equal to `length(at.2)-1`

‘Aggregate output‘

is a table of summary statistics when `at.2` has `length > 2`. Reports the mean second difference and mean absolute Wald statistic for the second differences. If `at.2` has `length<3`, it is not returned

‘Marginal effects‘

if `return.dydx=TRUE`, returns the marginal effects (partial derivatives with respect to `var1`) for each dyad in the ERGM sample space.

Note

Syntax is loosely based on the margins package by Thomas Leeper and Stata's margins commands. See ergm.AME to compute average marginal effects. Currently only considers two-ways interactions.

Author(s)

Scott Duxbury, Associate Professor, University of North Carolina–Chapel Hill, Department of Sociology.

References

- Duxbury, Scott W. 2023. "The Problem of Scaling in Exponential Random Graph Models." *Sociological Methods & Research*.
- Duxbury, Scott W., and Jenna Wertsching. 2023. "Scaling Bias in Pooled Exponential Random Graph Models." *Social Networks*.
- Duxbury, Scott W. 2025. "Marginal Substructural Effects for Exponential Random Graph Models." Working Paper.
- Long, J. Scott, and Sarah Mustillo. 2018. "Using Predictions and Marginal Effects to Compare Groups in Regression Models for Binary Outcomes." *Sociological Methods and Research*
- Mize, Trenton D. 2019. "Best Practices for Estimating, Interpreting, and Presenting Nonlinear Interaction Effects." *Sociological Science*

See Also

[ergm.AME](#) [ergm.mma](#) [margins](#)

Examples

```
library(ergm)

data("faux.dixon.high")

set.seed(21093)

my.ergm<-ergm(faux.dixon.high~edges+
              nodeicov("grade")+
              nodeocov("grade")+
              nodeifactor("sex")+
              nodeofactor("sex")+
              absdiff("grade")+
              nodematch("sex")+
              mutual)

#main effect
```

```

main.eff<-ergm.MEM(my.ergm,"mutual")

#main effect of "mutual" when sender and receiver effects for sex == 1
ergm.MEM(my.ergm,"mutual",
          at.controls=c("nodeifactor.sex.2","nodeofactor.sex.2"),
          control_vals=c(1,1))

##with moderator
sex.inter<-ergm.MEM(my.ergm,"nodeifactor.sex.2","nodeofactor.sex.2","nodematch.sex")

##with moderator at specific levels
grade.inter<-ergm.MEM(my.ergm,"nodeicov.grade","nodeocov.grade","absdiff.grade",
                      at.2=c(9,12))

##when moderator changes from mean to mean +sd
tie.probs<-edge.prob2(my.ergm)
m<-mean(tie.probs$nodeocov.grade)
m.sd<-m+sd(tie.probs$nodeocov.grade)

grade.inter.at.means<-ergm.MEM(my.ergm,"nodeicov.grade","nodeocov.grade",
                               "absdiff.grade",at.2=c(m,m.sd))

```

ergm.mma

Function to conduct marginal effects mediation analysis in ERGM.

Description

ergm.mma is a function to conduct mediation analysis in ERGM using marginal effects. It computes the total, direct, and indirect average marginal effects for two nested ERGMs fit to the same data. Can be used with discrete or continuous mediators and for multiple mediator analyses. Currently accepts ergm, btergm, mtergm, ergm.count, mlergm, and ergm.multi objects.

Usage

```

ergm.mma(restricted.model,
          full.model,
          direct.effect,
          mediator,
          at.controls=NULL,
          control_vals=NULL,
          ME="AME")

```

Arguments

restricted.model
the ergm without the mediator variable.

<code>full.model</code>	the ergm including the mediator variable.
<code>mediator</code>	a character string containing the name of the mediator variable. When examining joint mediation with multiple mediators, this is a vector with each entry containing the character strings for each mediator variable.
<code>direct.effect</code>	a character string containing the name of the direct effect.
<code>at.controls</code>	a vector of character strings containing the names of control variables to hold at prespecified values.
<code>control_vals</code>	a vector of numeric values to hold <code>at.controls</code> at during estimation.
<code>ME</code>	the type of marginal effect to calculate. Currently accepts either "AME" or "MEM". Default it "AME"

Details

ergm.mma conducts marginal effects mediation analysis for nested ERGMs fit to the same network by computing the change in average marginal effects between models. The AMEs are estimated on the scale of tie probabilities, not the linear component (log-odds). Standard errors are computed using the Delta method.

A few notes on using ergm.mma:

- ergm.mma default specification uses the AME, which can be computationally intensive in large networks or high dimensional models. If run times are an issue, try specifying `ME="MEM"`.
- ergm.mma can be applied to discrete mediators or multiple mediator analysis.

Value

Returns a table including the point estimates and standard errors for total, direct, and indirect effects. A single indirect effect is reported for joint mediation analyses. All marginal effects are computed on the scale of tie probabilities, not the linear component (log-odds).

Note

Note that ergm.mma may have restrictive run times in large networks or high dimensional models.

Author(s)

Scott Duxbury, Associate Professor, University of North Carolina–Chapel Hill, Department of Sociology.

References

Duxbury, Scott W. 2023. "The Problem of Scaling in Exponential Random Graph Models." *Sociological Methods & Research*.

See Also

[ergm.AME](#) [ergm.mod.mma](#)

Examples

```

library(ergm)

data("faux.mesa.high")

faux.mesa.high

set.seed(21093)

my.ergm1<-ergm(faux.mesa.high~edges+
  nodecov("Grade")+
  nodefactor("Race")+
  nodefactor("Sex")+
  nodematch("Grade")+
  nodematch("Sex")+
  nodematch("Race"),
  estimate = "MLE")

set.seed(21093)

my.ergm2<-ergm(faux.mesa.high~edges+
  nodecov("Grade")+
  nodefactor("Race")+
  nodefactor("Sex")+
  nodematch("Grade")+
  nodematch("Sex")+
  nodematch("Race")+
  gwesp(.5,fixed=TRUE))

###gwesp as single mediator
ergm.mma(my.ergm1,my.ergm2,mediator="gwesp.fixed.0.5",
  direct.effect="nodecov.Grade")

###multiple mediators

set.seed(21093)

my.ergm3<-ergm(faux.mesa.high~edges+
  nodecov("Grade")+
  nodefactor("Race")+
  nodefactor("Sex")+
  nodematch("Grade")+
  nodematch("Sex")+
  nodematch("Race")+
  gwesp(1,fixed=TRUE)+
  gwesp(.5,fixed=TRUE),

```

```

        #increase MC sample to facilitate convergence
        control = control.ergm(MCMC.samplesize=5000)
    )

    ##joint mediation for gwdsp and gwesp
    ergm.mma(my.ergm2,my.ergm3,mediator=c("gwdsp.fixed.1","gwesp.fixed.0.5"),
            direct.effect=names(my.ergm3$coef[8]))

```

ergm.mod.mma	<i>Function to assess mediated moderation using marginal effects in ERGM.</i>
--------------	---

Description

ergm.mod.mma is a function to assess mediated moderation in ERGM using average marginal effects (AME), e.g. when the main effect is a part of an interaction. If joint=TRUE, the function computes the total, partial, and indirect effects for all parts of the interaction. If int.eff=TRUE, the function computes the same values, but only for the moderating effect (e.g., only the moderated portion). Both joint=TRUE and int.eff=TRUE can be set simultaneously. Currently accepts ergm, btergm, mtergm, ergm.count, mlergm, and ergm.multi objects.

Usage

```

ergm.mod.mma(restricted.model,
             full.model,
             var1,
             var2,
             inter,
             mediator,
             at.2=NULL,
             joint=FALSE,
             int.eff=FALSE,
             at.controls=NULL,
             control_vals=NULL)

```

Arguments

restricted.model	the ergm without the mediator variable.
full.model	the ergm including the mediator variable.
var1	the character string containing the name of the main effect of the direct effect.
var2	the character string containing the name of the moderator variable for the direct effect.
inter	the character string containing the name of the interaction for the direct effect.

<code>at.2</code>	the levels of <code>var2</code> at which to compute the average marginal effects. If <code>at.2=NULL</code> , the function computes the AME at each unique value of <code>var2</code> . <code>at.2</code> can also be set to meaningful values, such as the mean of <code>var2</code> or other discrete values of interest. If <code>var2</code> is discrete, leaving <code>at.2</code> <code>NULL</code> will compute the AME for each discrete category of the moderator variable. It is usually safe to leave <code>at.2=NULL</code> unless examining a large network where <code>var2</code> has a large number of unique values, in which case computation times can be restrictive.
<code>mediator</code>	a character string containing the name of the mediator variable. When examining joint mediation with multiple mediators, this is a vector with each entry containing the character strings for each mediator variable.
<code>joint</code>	logical parameter indicates whether to compute the mediated composite effect.
<code>int.eff</code>	logical parameter indicating whether to compute the mediated effect <i>only</i> for the interaction, net of the main effect. Either <code>int.eff</code> or <code>joint</code> must be set to <code>TRUE</code> for function to work.
<code>at.controls</code>	a vector of character strings containing the names of control variables to hold at prespecified values.
<code>control_vals</code>	a vector of numeric values to hold <code>at.controls</code> at during estimation.

Details

`ergm.mod.mma` conducts mediation analyses using marginal effects when the main effect is part of an interaction. There are three ways to construe the mediating pathways. First, a researcher may be interested in whether a mediator variable mediates the main effect in an interaction, net of the moderator. To assess this, use `ergm.mma` for `var1`. This will compute the AME when the effect of the moderator is zero. Second, a researcher may be interested in the total or composite effect of an interaction, including the moderator and main effect. To assess this, set `joint=TRUE`. This computes the indirect effect for the AME at every level of `at.2`. The third scenario is when a researcher is interested in only the interaction effect. The interaction effect is defined here as the difference in AMEs between levels of a moderator (e.g., levels of `at.2`). Thus, the indirect effect is the third difference, or the difference in second differences between the restricted and full models. An example is whether triad closure mediates the effect of matched nodal characteristics, net of the main effect for nodal attributes (e.g., net of `nodefactor/nodecov`). To assess this, set `int.eff=TRUE`. Both `joint=TRUE` and `int.eff=TRUE` can be specified at the same time, but at least one must be specified for the function to work.

Value

Returns a list object containing the total, partial, and indirect AMEs at each level of `at.2`. If `joint=TRUE` is specified, returns:

<code>marginal.effects</code>	a list containing the indirect, total, and partial effects for each level of <code>at.2</code>
<code>summary.stats</code>	a table reporting the mean second difference and mean absolute Wald statistic for the indirect AMEs.

If `int.eff=TRUE` is specified, returns a list with elements:

<code>partial.marginal.effects</code>	the marginal effects for each level of <code>at.2</code> in the full model.
---------------------------------------	---

total.marginal.effects the marginal effects for each level of at.2 in the restricted model.

partial.second.diffs the second differences between levels of at.2 in the full model.

total.second.diffs the second differences between levels of at.2 in the restricted model.

third.diffs the difference in second differences between the partial and full models.

summary.output the mean third difference and mean absolute Wald statistic for the third differences.

If both int.eff=TRUE and joint=TRUE, returns a list with elements:

joint.effect a list containing all elements when joint=TRUE

moderator.effect a list containing all elements when int.eff=TRUE.

Author(s)

Scott Duxbury, Associate Professor, University of North Carolina–Chapel Hill, Department of Sociology.

References

Duxbury, Scott W. 2023. "The Problem of Scaling in Exponential Random Graph Models." *Sociological Methods & Research*.

See Also

[ergm.AME](#) [ergm.MEM](#) [ergm.mma](#)

Examples

```
library(ergm)

data("faux.dixon.high")

faux.dixon.high

my.ergm<-ergm(faux.dixon.high~edges+
  nodeifactor("sex")+
  nodeofactor("sex")+
  nodematch("sex"))

my.ergmb<-ergm(faux.dixon.high~edges+
  nodeifactor("sex")+
  nodeofactor("sex")+
  nodematch("sex")+
```

```

mutual)

summary(my.ergmb)

sex.mod.med<-ergm.mod.mma(my.ergm,my.ergmb,var1="nodeifactor.sex.2",var2="nodeofactor.sex.2",
inter="nodematch.sex",mediator="mutual",int.eff = TRUE, joint=TRUE)

```

ergm.MSE

*Function to compute marginal substructural effects in ERGM.***Description**

ERGM function to compute marginal substructural effects in ERGM, as described by Duxbury (2025). Intended for use with endogenous substructural terms in ERGMs. Also provides utilities to test interactions between substructural terms and exogenous covariates. See [ergm.AME](#) for average marginal effects and [ergm.MEM](#) for marginal effects at means.

Usage

```

ergm.MSE(model,
  substructural_effect,
  higher_order_term=NULL,
  lower_order_term=NULL,
  at.lower_order_term=NULL,
  estimate="aMSE",
  var2=NULL,
  inter=NULL,
  at.2=NULL,
  at.controls=NULL,
  control_vals=NULL,
  return_Jac=FALSE)

```

Arguments

<code>model</code>	the model to obtain MSE estimates. Currently accepts <code>ergm</code> , <code>btergm</code> , <code>mtergm</code> , <code>ergm.count</code> , <code>mlergm</code> , and <code>ergm.multi</code> objects.
<code>substructural_effect</code>	the character string containing the name of the endogenous substructural term to use to estimate the MSE.
<code>higher_order_term</code>	character string or vector of character strings containing the name(s) of all higher-order endogenous substructures. Optional value. Default value is <code>NULL</code> .
<code>lower_order_term</code>	character string or vector of character strings containing the name(s) of all lower-order endogenous substructures. Terms provided will be held at a fixed value as provided with the <code>at.lower_order_term</code> argument.

at.lower_order_term	a vector of numeric values to assign to variables provided in lower_order_term argument. lower_order_term will be held at those values when calculating MSE. Should be a 1:1 correspondence between the terms in at.lower_order_term and lower_order_term. Defaults to 1 if left NULL
estimate	the type of MSE summary statistic to compute. Should be one of "aMSE", "MSEm", "tMSE", or "tMSEm". See details below for information on each estimand. Default is "aMSE"
var2	the character string containing the name of an exogenous variable that interactions with substructural_term. Used when there the researcher provides an interaction between an endogenous graph statistic and an exogenous attribute. Default value is NULL. Optional; can be ignored if no interactions are specified.
inter	the character string containing the interaction between substructural_term and the exogenous covariate assigned to var2. Default value is NULL. Optional; can be ignored if no interactions are specified.
at.2	the levels of var2 at which to compute the MSE in cases where there is an interaction between substructural_term and var2. Provided as a vector of values. If var2=NULL, this parameter is ignored. If var2!=NULL and at.2=NULL, computed holding var2 at c(0, 1).
at.controls	A character string containing the names of all control variables to be specified at fixed levels when calculating the MSE. Should match ERGM output exactly.
control_vals	A vector specifying the values to hold at.controls variables. Ignored if at.controls = NULL
return_Jac	logical parameter indicating whether to return the Jacobian matrix of partial derivatives used to compute the MSE covariance matrix. Defaults to FALSE.

Details

Computes the marginal substructural effect for a specified substructural term. Formally, the MSE for a unit is given by

$$MSE_{ij} = Pr(Y_{ij} = 1 | -y_{ij}, \delta_{ij}^+(y) = 1, x_{ij}) - Pr(Y_{ij} = 0 | -y_{ij}, \delta_{ij}^+(y) = 0, x_{ij})$$

where $\delta_{ij}^+(y)$ is the substructural effect of interest. The estimate argument determines whether the MSE should be taken holding controls at observed values (when set to "aMSE"), holding controls at means ("MSEm"), or by combining the estimates from multiple substructures (when set to "aTSE").

The terms provided in the lower_order_terms argument will be held constant at a value of 1 when estimating the MSE, unless alternative values are provided with the at.lower_order_term argument. If the higher_order_term argument is provided, the specified term will be held at a value of 0 when estimating the marginal substructural effect

When an endogenous network term is specified as an interaction with an exogenous term, the MSE for each level of the interaction can be obtained by providing the var2 and inter arguments. This returns the MSE at different levels of var2 along with a test of the differences between the MSEs. If at.controls and control_vals are provided, the MEM is calculated while holding specified control variables at fixed values provided in control_vals, rather than at their means

Value

A table containing the MSE estimate and variance estimates. If `inter` and `var2` are provided, a list is returned containing:

<code>change_in_MSE</code>	is the change in MSE when the value of <code>var2</code> changes in value, usually from 0 to 1 unless specified with the <code>at.2</code> argument
<code>MSE</code>	is a table containing MSE estimates, standard errors, and Wald tests for the MSE at each level of <code>var2</code>

Note

Syntax is loosely based on the `margins` package by Thomas Leeper and Stata's `margins` commands. See `ergm.AME` to compute average marginal effects. Currently only considers two-ways interactions.

Author(s)

Scott Duxbury, Associate Professor, University of North Carolina–Chapel Hill, Department of Sociology.

References

- Duxbury, Scott W. 2023. "The Problem of Scaling in Exponential Random Graph Models." *Sociological Methods & Research*.
- Duxbury, Scott W., and Jenna Wertsching. 2023. "Scaling Bias in Pooled Exponential Random Graph Models." *Social Networks*.
- Duxbury, Scott W. 2025. "Marginal Substructural Effects for Exponential Random Graph Models." Working Paper.
- Long, J. Scott, and Sarah Mustillo. 2018. "Using Predictions and Marginal Effects to Compare Groups in Regression Models for Binary Outcomes." *Sociological Methods and Research*
- Mize, Trenton D. 2019. "Best Practices for Estimating, Interpreting, and Presenting Nonlinear Interaction Effects." *Sociological Science*

See Also

[ergm.AME](#) [ergm.mma](#) [ergm.msma](#) [ergm.MEM](#) [margins](#)

Examples

```
library(ergm)

data("faux.dixon.high")

set.seed(21093)
```

```

my.ergm<-ergm(faux.dixon.high~edges+
  nodeicov("grade")+
  nodeocov("grade")+
  nodeifactor("sex")+
  nodeofactor("sex")+
  absdiff("grade")+
  nodematch("sex")+
  mutual+
  gwidegree(.5,fixed=TRUE))

#MSE at means
ergm.MSE(my.ergm,
  substructural_effect="mutual",
  lower_order_term="gwideg.fixed.0.5",
  estimate="MSEm")

#total effect of both endogenous terms
ergm.MSE(my.ergm,
  substructural_effect="mutual",
  lower_order_term="gwideg.fixed.0.5",
  estimate="tMSEm")

```

ergm.msma	<i>Function to compare marginal substructural effects between nested ERGMs.</i>
-----------	---

Description

ergm.msma tests the difference in a marginal substructural effect between two ERGMs fit to the same network that differ only in their specification. It computes the total, direct, and indirect marginal substructural effects. Can be used with discrete or continuous mediators and for multiple mediator analyses. Currently accepts ergm, btergm, mtergm, ergm.count, mlergm, and ergm.multi objects.

Usage

```

ergm.msma(restricted.model,
  full.model,
  direct_substructural_effect,
  higher_order_term=NULL,
  lower_order_term=NULL,
  at.lower_order_term=NULL,
  mediator,
  at.controls=NULL,

```

```
control_vals=NULL,
estimate="aMSE")
```

Arguments

<code>restricted.model</code>	the ergm without the mediator variable.
<code>full.model</code>	the ergm including the mediator variable.
<code>direct_substructural_effect</code>	a character string identifying the marginal substructural effect to be compared between models
<code>higher_order_term</code>	an optional parameter identifying any higher-order substructures to be held at 0 when calculating marginal substructural effects. <code>higher_order_term</code> must appear in both models when specified
<code>lower_order_term</code>	an optional parameter identifying lower-order substructures nested in <code>direct_substructural_effect</code> . When provided, the lower-order terms will be held at fixed values calculating direct and indirect marginal substructural effects
<code>at.lower_order_term</code>	optional parameter identifying the values to assign to <code>lower_order_term</code> . Should be provided as a vector of numeric values with each entry corresponding to a single entry in <code>lower_order_term</code> . Default is 1
<code>mediator</code>	a character string containing the name of the mediator variable. Can be specified as a vector with each entry containing the character strings for several mediator variables when there are multiple mediators.
<code>at.controls</code>	a vector of character strings containing the names of control variables to hold at prespecified values.
<code>control_vals</code>	a vector of numeric values to hold <code>at.controls</code> at during estimation.
<code>estimate</code>	the type of marginal substructural effect to calculate. Must be one of "aMSE", "MSEm", "tMSE", or "tMSEm"

Details

`ergm.msm` calculates the change in a marginal substructural effect between two models fit to the same data. Formally, define two models Mod^1 and Mod^2 that differ by the inclusion of one or more mediating variables. For a substructural network term of interest given by the change statistic $\delta_{ij}^+(y)$, the `ergm.msm` provides the following quantities:

$$\begin{aligned}
 &MSE^{Mod^1}(\delta_{ij}^+(y)) \\
 &MSE^{Mod^2}(\delta_{ij}^+(y)) \\
 &MSE^{Mod^1}(\delta_{ij}^+(y)) - MSE^{Mod^2}(\delta_{ij}^+(y))
 \end{aligned}$$

Where $MSE^{Mod^1}(\delta_{ij}^+(y))$ is the total effect corresponding to the marginal substructural effect of $\delta_{ij}^+(y)$ operating through all indirect pathways, $MSE^{Mod^2}(\delta_{ij}^+(y))$ is the direct effect of $\delta_{ij}^+(y)$ on tie formation, and $MSE^{Mod^1}(\delta_{ij}^+(y)) - MSE^{Mod^2}(\delta_{ij}^+(y))$ is the portion of the total effect explained by controlling for a confounding variable.

Value

Returns a table including the point estimates and standard errors for total, direct, and indirect effects. A single indirect effect is reported for joint mediation analyses. All marginal substructural effects are computed on the scale of tie probabilities, not the linear component (log-odds).

Note

Note that `ergm.msma` may have restrictive run times in large networks or high dimensional models.

Author(s)

Scott Duxbury, Associate Professor, University of North Carolina–Chapel Hill, Department of Sociology.

References

- Duxbury, Scott W. 2023. "The Problem of Scaling in Exponential Random Graph Models." *Sociological Methods & Research*.
- Duxbury, Scott W., and Jenna Wertsching. 2023. "Scaling Bias in Pooled Exponential Random Graph Models." *Social Networks*.
- Duxbury, Scott W. 2025. "Marginal Substructural Effects for Exponential Random Graph Models." Working Paper.

See Also

[ergm.AME](#) [ergm.mod.mma](#) [ergm.mma](#) [ergm.MSE](#) [ergm.MEM](#)

Examples

```
library(ergm)

data("faux.mesa.high")

faux.mesa.high

#does sociality explain triadic effects?

#model without sociality effects
my.ergm1<-ergm(faux.mesa.high~edges+
  nodematch("Grade")+
  nodematch("Sex")+
  nodematch("Race")+
  gwesp(.5,fixed=TRUE),
  control = control.ergm(MCMLE.termination="Hummel",
    MCMLE.effectiveSize = NULL,
    MCMC.samplesize = 5000,
    MCMLE.maxit=120))
```

```

#model with sociality effects
my.ergm2<-ergm(faux.mesa.high~edges+
  nodecov("Grade")+
  nodefactor("Race")+
  nodefactor("Sex")+
  nodematch("Grade")+
  nodematch("Sex")+
  nodematch("Race")+
  gwesp(.5,fixed=TRUE),
  control = control.ergm(MCMLE.termination="Hummel",
                          MCMLE.effectiveSize = NULL,
                          MCMC.samplesize = 5000,
                          MCMLE.maxit=120))

###sociality effects as mediators
ergm.msm(ergm1,
  my.ergm2,
  direct_substructural_effect="gwesp.fixed.0.5",
  mediator=c("nodecov.Grade","nodefactor.Sex.M","nodefactor.Race.White",
    "nodefactor.Race.Other","nodefactor.Race.Hisp","nodefactor.Race.NatAm"),
  estimate="aMSE")

```

GW_helper

Function to compute weighted change statistics for geometrically weighted terms.

Description

Computes geometric weights for a pre-specified subgraph count given a known decay parameter. Intended primarily for use with `ergm.MSE` to simplify specifications for values of lower-order terms.

Usage

```

GW_helper(term_count,
          decay,
          lower_bound=0)

```

Arguments

<code>term_count</code>	is the count of subgraphs to which a researcher wants to assign geometric weights. Should be provided as a scalar (single numeric value).
<code>decay</code>	is the decay parameter used to assign geometric weights provided by the researcher.

`lower_bound` is the baseline value for the subgraph count when calculating the geometric weights. Default value is 0.

Details

Computes the weighted value to assign to change statistics for a desired subgraph count. `term_count` corresponds to the subgraph count to assign geometric weights. For example, consider an ERGM specification where triadic effects are specified with a `gwap` statistic. `GW_helper` would be used to determine the change statistic for a given triad count. If the triad count is 2, this would be specified by setting the `term_count` argument to equal 2 and providing the decay parameter used to estimate the `gwap` term in an `ergm` call. `GW_helper` would return the change statistic for the `gwap` term when a potential tie increases the triad count to a value of 2, weighted by decay.

The `lower_bound` argument assigns the baseline value for the subgraph count in question. By default, `lower_bound` is set to 0. This means that, in the above example, if `term_count` is set to 2, the default behavior of `GW_helper` is to calculate the weighted change statistic for a `gwap` term that increases the number of triangles from 0 to 2. If `lower_bound` is set to 1 and `term_count` is set to 3, then `GW_helper` will calculate the change statistic when increasing the number of triangles from 1 to 3. Since geometrically weighted terms assign diminishing weights for each statistic added, selection of `lower_bound` and `term_count` will affect the value of the resulting change statistic in meaningful ways.

If the geometrically weighted term is estimated with fixed decay, `decay` is the value provided by the researcher within the `ergm` formula. If the term is estimated without fixed decay, `decay` is the decay parameter estimate from the data corresponding to a single entry in the `ergm` summary table parameter vector.

Value

Returns a scalar value reflecting the change statistic for a potential tie that increases the subgraph count captured by a geometrically weighted term by the values assigned to `lower_bound` and `term_count`. Intended primarily to facilitate use of `ergm.MSE` and `ergm.msma`.

Author(s)

Scott Duxbury, Associate Professor, University of North Carolina–Chapel Hill, Department of Sociology.

References

- Duxbury, Scott W. 2025. "Marginal Structural Effects for ERGM" *Working paper*.
 Hunter, David. 2007. "Curved Exponential Family Models for Social Networks" *Social Networks*.

Examples

```
library(ergm)
set.seed(21093)
data("faux.dixon.high")
```

```

# fit a model
my.ergm<-ergm(faux.dixon.high~edges+
  nodeicov("grade")+
  nodeocov("grade")+
  nodeifactor("sex")+
  nodeofactor("sex")+
  absdiff("grade")+
  nodematch("sex")+
  mutual+
  gwidegree(.5,fixed=TRUE))

#for a fixed decay of .5, what is the change statistic for a tie that increases
# degree from 0 to 2?

GW_helper(2,.5)

#for a fixed decay of .5, what is the change statistic for a tie that increases
# degree from 1 to 2?
GW_helper(2,.5,lower_bound=1)

##within a call to ergm.MSE
#creating a single mutual tie requires, at a minimum, increasing indegree
#from 0 to 1 for the recipient node

ergm.MSE(my.ergm,
  substructural_effect="mutual",
  lower_order_term="gwideg.fixed.0.5",
  at.lower_order_term=GW_helper(1,.5),
  estimate="MSEm")

#What happens if the tie moves indegree from 4 to 5 for the recipient node?

ergm.MSE(my.ergm,
  substructural_effect="mutual",
  lower_order_term="gwideg.fixed.0.5",
  at.lower_order_term=GW_helper(5,.5,lower_bound=4),
  estimate="MSEm")

```

vif.ergm

Function to diagnose multicollinearity in ERGM.

Description

Computes the variance inflation factor from the covariance matrix of the ERGM estimator.

Usage

```
vif.ergm(my.ergm)
```

Arguments

`my.ergm` an ergm object.

Details

Computes the variance inflation factor using the covariance matrix of the ERGM estimator. Values above 20 indicate the onset of moderate collinearity problems, values above 100 indicate that strong collinearity likely exists and may be problematizing the convergence of the Markov chain to the stationarity distribution. High VIFs alongside a near-singular Hessian matrix is a clear indicator of multicollinearity. Consider changing model specification in such cases.

Value

Returns a matrix of VIFs for each model parameter. Higher values indicate that the variable in one column of the data matrix is highly correlated with a linear combination of the remaining model covariates. High VIFs can also arise from a large bivariate correlation.

Author(s)

Scott Duxbury, Associate Professor, University of North Carolina–Chapel Hill, Department of Sociology.

References

Duxbury, Scott W. 2021. "Diagnosing Multicollinearity in Exponential Random Graph Models" *Sociological Methods and Research*.

Examples

```
set.seed(21093)

library(ergm)
data("faux.mesa.high")

my.ergm2<-ergm(faux.mesa.high~edges+
  nodecov("Grade")+
  nodefactor("Race")+
  nodefactor("Sex")+
  nodematch("Grade")+
  nodematch("Sex")+
  nodematch("Race"))

vif.ergm(my.ergm2)

#to compute the tolerance instead of the VIF

tol<-1/(vif.ergm(my.ergm2))
```

```
#to compute R squared  
r.sq<-1-tol
```

Index

- * **~ERGM**
 - edge.prob2, [2](#)
 - * **~collinearity**
 - vif.ergm, [24](#)
 - * **~ergm**
 - ergm.AME, [3](#)
 - ergm.MEM, [7](#)
 - ergm.mma, [10](#)
 - ergm.mod.mma, [13](#)
 - ergm.MSE, [16](#)
 - ergm.msma, [19](#)
 - GW_helper, [22](#)
 - vif.ergm, [24](#)
 - * **~marginal effects**
 - ergm.AME, [3](#)
 - ergm.MEM, [7](#)
 - ergm.mma, [10](#)
 - ergm.mod.mma, [13](#)
 - ergm.MSE, [16](#)
 - ergm.msma, [19](#)
 - GW_helper, [22](#)
 - * **~mediated moderation**
 - ergm.mod.mma, [13](#)
 - * **~mediation**
 - ergm.mma, [10](#)
 - ergm.mod.mma, [13](#)
 - ergm.msma, [19](#)
 - * **~moderation**
 - ergm.AME, [3](#)
 - ergm.MEM, [7](#)
 - ergm.mod.mma, [13](#)
 - ergm.MSE, [16](#)
 - * **~networks**
 - ergm.AME, [3](#)
 - ergm.MEM, [7](#)
 - ergm.mma, [10](#)
 - ergm.mod.mma, [13](#)
 - ergm.MSE, [16](#)
 - ergm.msma, [19](#)
 - GW_helper, [22](#)
 - vif.ergm, [24](#)
 - * **~tie probabilities**
 - edge.prob2, [2](#)
- edge.prob2, [2](#)
- edgeprob, [2](#)
- ergm.AME, [3](#), [9](#), [11](#), [15](#), [16](#), [18](#), [21](#)
- ergm.MEM, [6](#), [7](#), [15](#), [16](#), [18](#), [21](#)
- ergm.mma, [6](#), [9](#), [10](#), [15](#), [18](#), [21](#)
- ergm.mod.mma, [11](#), [13](#), [21](#)
- ergm.MSE, [16](#), [21](#)
- ergm.msma, [18](#), [19](#)
- GW_helper, [22](#)
- margins, [6](#), [9](#), [18](#)
- vif.ergm, [24](#)