Package 'ergMargins'

May 8, 2024

R topics documented:

---

edge.prob2   *Function to compute tie probabilities for each dyad.*

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 identification variable for the vertices/dyads, or the estimated probability of observing a tie. See edgeprob() in btergm for more details.

Author(s)

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

model	the ergm object to be analyzed.
var1	the character string containing the name of the focal variable on which to compute the average marginal effect. If var2 is NULL, this is the average marginal effect.
var2	if examining moderation, the character string containing the name of the moderator variable. Default value is NULL.
inter	if examining moderation, the character string containing the name of the interaction between var1 and var2. Default value is NULL.
at.2	the levels of var2 at which to compute the average marginal effects. If var2=NULL, this parameter is ignored. If var2!=NULL and at.2=NULL, the function computes the AME at each unique value of var2. at.2 can also be set to meaningful values, such as the mean of var2 or other discrete values of interest. If var2 is

	discrete, leaving at.2 NULL will compute the AME for each discrete category of the moderator variable. It is usually safe to leave at.2=NULL unless examining a large network where var2 has a large number of unique values, in which case computation times can be restrictive.
at.controls	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.
control_vals	A vector specifying the values to hold at.controls variables. Ignored if at.controls = NULL
return.dydx	logical parameter indicating whether to return the marginal effects for each dyad. Defaults to FALSE.
return.at.2	logical parameter indicating whether to return the levels of at.2 as part of the output list object. Default value is F. 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 allows 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 vales.

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, Assistant Professor, University of North Carolina–Chapel Hill, Department of Sociology.

References

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

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.MEM 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.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

`model`	the ergm object to be analyzed.
`var1`	the character string containing the name of the focal variable on which to compute the MEM. If var2 is NULL, this is the simple MEM.
`var2`	the character string containing the name of the moderator variable. Default value is NULL.
`inter`	the character string containing the name of the interaction between var1 and var2. Default value is NULL.
`at.2`	the levels of var2 at which to compute the MEMs. If var2=NULL, this parameter is ignored. If var2!=NULL and at.2=NULL, the function computes the AME at each unique value of var2. at.2 can also be set to meaningful values, such as the mean of var2 or other discrete values of interest. If var2 is discrete, leaving at.2 NULL will compute the AME for each discrete category of the moderator variable. It is usually safe to leave at.2=NULL unless examining a large network where var2 has a large number of unique values, in which case computation times can be restrictive.
`at.controls`	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.
`control_vals`	A vector specifying the values to hold at.controls variables. Ignored if at.controls = NULL
`return.dydx`	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, Assistant Professor, University of North Carolina–Chapel Hill, Department of Sociology.

References

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

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.
full.model	the ergm including the mediator variable.
mediator	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.
direct.effect	a character string containing the name of the direct effect.
at.controls	a vector of character strings containing the names of control variables to hold at prespecified values.
control_vals	a vector of numeric values to hold at.controls at during estimation.
ME	the type of marginal effect to calculate. Currently accepts either "AME" or "MEM". Defaul 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, Assistant Professor, University of North Carolina–Chapel Hill, Department of Sociology.

References

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

Karlson, Kristian Bernt, Holm, Anders, and Richard Breen. 2012. "Comparing Regression Coefficients Between Same-sample Nested Models Using Logit and Probit: A New Method." *Sociological Methodology* 42: 286 - 313.

See Also

[ergm.AME](ergm.mod.mma) 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")+
                gwdsp(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	*Function to assess mediated moderation using marginal effects in ERGM.*

Description

ergm.mod.mma is a function to asssess 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.

at.2   the levels of var2 at which to compute the average marginal effects. If at.2=NULL, the function computes the AME at each unique value of var2. at.2 can also be set to meaningful values, such as the mean of var2 or other discrete values of interest. If var2 is discrete, leaving at.2 NULL will compute the AME for each discrete category of the moderator variable. It is usually safe to leave at.2=NULL unless examining a large network where var2 has a large number of unique values, in which case computation times can be restrictive.

mediator   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.

joint   logical parameter indicates whether to compute the mediated composite effect.

int.eff   logical parameter indicating whether to compute the mediated effect *only* for the interaction, net of the main effect. Either int.eff or joint must be set to TRUE for function to work.

at.controls   a vector of character strings containing the names of control variables to hold at prespecified values.

control_vals   a vector of numeric values to hold at.controls 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:

marginal.effects

a list containing the indirect, total, and partial effects for each level of at.2

summary.stats   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:

partial.marginal.effects

the marginal effects for each level of at.2 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, Assistant Professor, University of North Carolina–Chapel Hill, Department of Sociology.

References

Duxbury, Scott W. Forthcoming. "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)
```

---

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, Assistant Professor, University of North Carolina–Chapel Hill, Department of Sociology.

References

Duxbury, Scott W. 2018. "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