Package 'GERGM'

November 17, 2015

Type Package

Title Estimation and Fit Diagnostics for Generalized Exponential Random Graph Models

Version 0.4.2

Date 2015-11-13

Author Matthew J. Denny <mdenny@psu.edu>, James D. Wilson <jdwilson1212@gmail.com>, Skyler Cranmer <cranmer.12@osu.edu >, Bruce A. Desmarais <bdesmarais@psu.edu>, Shankar Bhamidi <bhamidi@email.unc.edu>

Maintainer Matthew J. Denny <mdenny@psu.edu>

Description Estimation and diagnosis of the convergence of Generalized Exponential Random Graph Models (GERGM) via Gibbs sampling or Metropolis Hastings with exponential down weighting.

URL https://github.com/matthewjdenny/GERGM

License GPL-3

Imports Rcpp, ggplot2, methods, stringr, igraph

Depends R (>= 3.2.1)

LinkingTo BH, Rcpp, RcppArmadillo

RoxygenNote 5.0.0

NeedsCompilation yes

Repository CRAN

Date/Publication 2015-11-17 14:40:34

R topics documented:

Estimate_Plot *Generate parameter estimate plot with 95 percent CI's from a GERGM object.*

Description

Generate parameter estimate plot with 95 percent CI's from a GERGM object.

Usage

```
Estimate_Plot(GERGM_Object)
```

Arguments

GERGM_Object	The object returned by the estimation procedure using the GERGM function.

Value

A parameter estimate plot.

gergm *A Function to estimate a GERGM.*

Description

The main function provided by the package.

Usage

```
gergm(formula, covariate_data = NULL, normalization_type = c("log",
  "division"), network_is_directed = c(TRUE, FALSE),
  use_MPLE_only = c(FALSE, TRUE), transformation_type = c("Cauchy",
  "LogCauchy", "Gaussian", "LogNormal"), estimation_method = c("Gibbs",
  "Metropolis"), maximum_number_of_lambda_updates = 10,
  maximum_number_of_theta_updates = 100,
  number_of_networks_to_simulate = 500, thin = 1, proposal_variance = 0.1,
  downweight_statistics_together = TRUE, MCMC_burnin = 100, seed = 123,
  convergence_tolerance = 0.01, MPLE_gain_factor = 0,
  acceptable_fit_p_value_threshold = 0.05, force_x_theta_updates = 1,
  output_directory = NULL, output_name = NULL, generate_plots = TRUE,
  using_correlation_network = FALSE)
```

Arguments

formula	A formula object that specifies the relationship between statistics and the observed network. Currently, the following statistics can be specified: c("out2star", "in2star", "ctriads", "recip", "ttriads", "edgeweigt").
covariate_data	A data frame containing node level covariates the user wished to transform into sender or reciever effects. It must have row names that match every entry in colnames(raw_network), should have descriptive column names. If left NULL, then no sender or reciever effects will be added.
normalization_type	If only a raw_network is provided then, the function will automatically check to determine if all edges fall in the [0,1] interval. If edges are determined to fall outside of this interval, then a trasformation onto the interval may be specified. If "division" is selected, then the data will have a value added to them such that the minimum value is atleast zero (if necessary) and then all edge values will be divided by the maximum to ensure that the maximum value is in [0,1]. If "log" is selected, then the data will have a value added to them such that the minimum value is atleast zero (if necessary), then 1 will be added to all edge values before they are logged and then divided by the largest value, again ensuring that the resulting network is on [0,1]. Defaults to "log" and need not be set to NULL if providing covariates as it will be ignored.
network_is_directed	Logical specifying whether or not the observed network is directed. Default is TRUE.
use_MPLE_only	Logical specifying whether or not only the maximum pseudo likelihood estimates should be obtained. In this case, no simulations will be performed. Default is FALSE.
transformation_type	Specifies how covariates are transformed onto the raw network. When working with heavly tailed data that are not strictly positive, select "Cauchy" to transform the data using a Cauchy distribution. If data are strictly positive and heavy tailed (such as financial data) it is suggested the user select "LogCauchy" to perform a Log-Cauchy transformation of the data. For a tranformation of the data using a Gaussian distribution, select "Gaussian" and for strictly positive raw networks, select "LogNormal". The Default value is "Cauchy".
estimation_method	Simulation method for MCMC estimation. Default is "Gibbs" which will generally be faster with well behaved networks but will not allow for exponential downweighting.
maximum_number_of_lambda_updates	Maximum number of iterations of outer MCMC loop which alternately estimates transform parameters and ERGM parameters. In the case that data_transformation = NULL, this argument is ignored. Default is 10.
maximum_number_of_theta_updates	Maximum number of iterations within the MCMC inner loop which estimates the ERGM parameters. Default is 100.
number_of_networks_to_simulate	Number of simulations generated for estimation via MCMC. Default is 500.

`thin` The proportion of samples that are kept from each simulation. For example, thin = 1/200 will keep every 200th network in the overall simulated sample. Default is 1.

`proposal_variance`
The variance specified for the Metropolis Hastings simulation method. This parameter is inversely proportional to the average acceptance rate of the M-H sampler and should be adjusted so that the average acceptance rate is approximately 0.25. Default is 0.1.

`downweight_statistics_together`
Logical specifying whether or not the weights should be applied inside or outside the sum. Default is TRUE and user should not select FALSE under normal circumstances.

`MCMC_burnin` Number of samples from the MCMC simulation procedure that will be discarded before drawing the samples used for estimation. Default is 100.

`seed` Seed used for reproducibility. Default is 123.

`convergence_tolerance`
Threshold designated for stopping criterion. If the difference of parameter estimates from one iteration to the next all have a p -value (under a paired t-test) greater than this value, the parameter estimates are declared to have converged. Default is 0.01.

`MPLE_gain_factor`
Multiplicative constant between 0 and 1 that controls how far away the initial theta estimates will be from the standard MPLEs via a one step Fisher update. In the case of strongly dependent data, it is suggested to use a value of 0.10. Default is 0.

`acceptable_fit_p_value_threshold`
A p-value threshold for how closely statistics of observed network conform to statistics of networks simulated from GERGM parameterized by converged final parameter estimates. Default value is 0.05.

`force_x_theta_updates`
Defaults to 1 where theta estimation is not allowed to converge until thetas have updated for x iterations . Useful when model is not degenerate but simulated statistics do not match observed network well when algorithm stops after first x updates.

`output_directory`
The directory where you would like output generated by the GERGM estimation proceedure to be saved (if output_name is specified). This includes, GOF, trace, and parameter estimate plots, as well as a summary of the estimation proceedure and an .Rdata file containing the GERGM object returned by this function. May be left as NULL if the user would prefer all plots be printed to the graphics device.

`output_name` The common name stem you would like to assign to all objects output by the gergm function. Default value of NULL will not save any output directly to .pdf files, it will be printed to the console instead. Must be a character string or NULL. For example, if "Test" is supplied as the output_name, then 4 files will be output: "Test_GOF.pdf", "Test_Parameter_Estim ates.pdf", "Test_GERGM_Object.Rdata", "Test_Estimation_Log.txt", and "Test_Trace_Plot.pdf"

generate_plots Defaults to TRUE, if FALSE, then no diagnostic or parameter plots are generated.

using_correlation_network
Defaults to FALSE. Experimental.

Value

A gergm object containing parameter estimates.

Examples

```
## Not run:
set.seed(12345)
net <- matrix(rnorm(100,0,20),10,10)
colnames(net) <- rownames(net) <- letters[1:10]
formula <- net ~ recip + edges

test <- gergm(formula,
              normalization_type = "division",
              network_is_directed = TRUE,
              use_MPLE_only = FALSE,
              estimation_method = "Metropolis",
              maximum_number_of_lambda_updates = 1,
              maximum_number_of_theta_updates = 5,
              number_of_networks_to_simulate = 40000,
              thin = 1/10,
              proposal_variance = 0.5,
              downweight_statistics_together = TRUE,
              MCMC_burnin = 10000,
              seed = 456,
              convergence_tolerance = 0.01,
              MPLE_gain_factor = 0,
              force_x_theta_update = 4)

## End(Not run)
```

GOF *Generate Goodness Of Fit plot from a GERGM object.*

Description

Generate Goodness Of Fit plot from a GERGM object.

Usage

```
GOF(GERGM_Object)
```

Arguments

GERGM_Object The object returned by the estimation procedure using the GERGM function.

Value

A GOF plot.

`plot_network` *Plots of value-edged networks.*

Description

Generates a visualization of a value-edged network.

Usage

```
plot_network(sociomatrix, threshold = 0.5, save_pdf = FALSE,
  pdf_name = "Test.pdf", output_directory = "./")
```

Arguments

`sociomatrix`	A square numeric matrix (socimatrix) with real valued edges (no NA's).
`threshold`	The threshold for removing edges from the network in order to calculate the positions for the nodes using the futcherman reingold algorithm. The value is multiplied against max(abs(sociomatrix)) to determine the threshold. Defaults to 0.5.
`save_pdf`	Logical indicating whether the plot should be saved to a PDF.
`pdf_name`	The name we would like to give to the output file. Be sure to include a ".pdf" extension.
`output_directory`	The directory where the user would like to output the PDF if save_pdf == TRUE.

Examples

```
set.seed(12345)
sociomatrix <- matrix(rnorm(400,0,20),20,20)
colnames(sociomatrix) <- rownames(sociomatrix) <- letters[1:20]
plot_network(sociomatrix)
```

simulate_networks *A Function to simulate networks from a GERGM with given theta parameters.*

Description

Simulates networks from a GERGM for a given set of parameter values.

Usage

```
simulate_networks(formula, edges = 0, recip = 0, ttriads = 0,
  ctriads = 0, in2star = 0, out2star = 0,
  simulation_method = c("Metropolis", "Gibbs"),
  network_is_directed = c(TRUE, FALSE),
  number_of_networks_to_simulate = 500, thin = 1, proposal_variance = 0.1,
  downweight_statistics_together = TRUE, MCMC_burnin = 100, seed = 123,
  simulate_correlation_network = FALSE)
```

Arguments

formula	A formula object that specifies which statistics the user would like to include while simulating the network, and the network the user is providing as the initial network. Currently, the following statistics can be specified: c("out2star", "in2star", "ctriads", "recip", "ttriads", "edgeweight").
edges	The theta value provided for the edges parameter, defaults to 0. Only statistics for structural terms included in formula will be used.
recip	The theta value provided for the reciprocity parameter, defaults to 0. Only statistics for structural terms included in formula will be used.
ttriads	The theta value provided for the transitive triads parameter, defaults to 0. Only statistics for structural terms included in formula will be used.
ctriads	The theta value provided for the cyclic triads parameter, defaults to 0. Only statistics for structural terms included in formula will be used.
in2star	The theta value provided for the in 2-stars parameter, defaults to 0. Only statistics for structural terms included in formula will be used.
out2star	The theta value provided for the out 2-starts parameter, defaults to 0. Only statistics for structural terms included in formula will be used.
simulation_method	Default is "Metropolis" which allows for exponential downweighting, can also be "Gibbs".
network_is_directed	Logical specifying whether or not the observed network is directed. Default is TRUE.
number_of_networks_to_simulate	Number of simulations generated for estimation via MCMC. Default is 500.

`thin` The proportion of samples that are kept from each simulation. For example, thin = 1/200 will keep every 200th network in the overall simulated sample. Default is 1.

`proposal_variance` The variance specified for the Metropolis Hastings simulation method. This parameter is inversely proportional to the average acceptance rate of the M-H sampler and should be adjusted so that the average acceptance rate is approximately 0.25. Default is 0.1.

`downweight_statistics_together` Logical specifying whether or not the weights should be applied inside or outside the sum. Default is TRUE and user should not select FALSE under normal circumstances.

`MCMC_burnin` Number of samples from the MCMC simulation procedure that will be discarded before drawing the samples used for estimation. Default is 100.

`seed` Seed used for reproducibility. Default is 123.

`simulate_correlation_network` Defaults to FALSE. Experimental.

Value

A list object containing simulated networks and parameters used to specify the simulation. See the $MCMC_Output field for simulated networks.

Examples

```
set.seed(12345)
net <- matrix(runif(100),10,10)
diag(net) <- 0
colnames(net) <- rownames(net) <- letters[1:10]
formula <- net ~ edges + ttriads + in2star

test <- simulate_networks(formula,
 edges = 0.2,
 ttriads = 0.6,
 in2star = -0.8,
 network_is_directed = TRUE,
 simulation_method = "Metropolis",
 number_of_networks_to_simulate = 10000,
 thin = 1/10,
 proposal_variance = 0.5,
 downweight_statistics_together = TRUE,
 MCMC_burnin = 1000,
 seed = 456)
```

`Trace_Plot` *Generate trace plot of network density from a GERGM object.*

Description

Generate trace plot of network density from a GERGM object.

Usage

```
Trace_Plot(GERGM_Object)
```

Arguments

`GERGM_Object`	The object returned by the estimation procedure using the GERGM function.

Value

A trace plot of network density.

Index