Package 'HLSM'

May 7, 2026

Encoding UTF-8

Type Package

Title Hierarchical Latent Space Network Model

Version 0.9.2

Date 2025-06-04

Description Fits latent space models for single networks and hierarchical latent space models for ensembles of networks as described in Sweet, Thomas & Junker (2013).

Depends R (>= 4.4.0)

ByteCompile TRUE

License GPL (>= 2)

Imports MASS, coda, igraph, grDevices, graphics, methods, abind, stats

LazyData yes

NeedsCompilation yes

RoxygenNote 7.3.2

Author Samrachana Adhikari [aut],
Tracy Sweet [aut, cre]

Maintainer Tracy Sweet <tsweet@umd.edu>

Repository CRAN

Date/Publication 2025-06-04 15:30:09 UTC

Contents

`HLSMcovplots` *Function plot posterior summaries (boxplots) for regression coefficients*

Description

Function plot posterior summaries (boxplots) for regression coefficients

Usage

```
HLSMcovplots(fitted.model, burnin=0, thin=1, verbose=TRUE)
```

Arguments

`fitted.model`	Model fit using HLSM fitting function; should be a HLSM or LSM object
`burnin`	Amount of burnin to remove
`thin`	Amount to thin each chain
`verbose`	logical to indicate whether message about order of covariates is included with plots

Details

The plots show posterior means and 50 and 95 percent equal-tailed credible intervals.

Value

No return value, makes a plot in plotting window

Author(s)

Sam Adhikari & Tracy Sweet

References

Tracy M. Sweet, Andrew C. Thomas and Brian W. Junker (2013), "Hierarchical Network Models for Education Research: Hierarchical Latent Space Models", Journal of Educational and Behavorial Statistics.

HLSMdiag *Function to conduct diagnostics the MCMC chain from a random effect HLSM (and HLSMfixedEF for fixed effects model)*

Description

Function to compute and report diagnostic plots and statistics for a single or multiple HLSM objects.

Usage

```
HLSMdiag(object, burnin = 0,
         diags = c('psrf', 'raftery', 'traceplot', 'autocorr'),
         col = 1:6, lty = 1)
```

Arguments

object	object or list of objects of class 'HLSM' returned by `HLSMrandomEf()` or `HLSMfixedEF()`
burnin	numeric value to burn the chain while extracting results from the 'HLSM'object. Default is `burnin = 0`.
diags	a character vector that is a subset of `c('psrf', 'raftery', 'traceplot', 'autocorr')`. Default returns all diagnostics. If only a single chain is supplied in object, `'psrf'` throws a warning if explicitly requested by user.
col	a character or integer vector specifying the colors for the traceplot and autocorr plot
lty	a character or integer vector specifying the linetype for the traceplot and autocorr plot

Value

Returns an object of class "HLSMdiag". It is a list that contains variable-level diagnostic tables from either or both of the raftery diagnostic and psrf diagnostic.

call	the matched call.
raftery	list of matrices of suggested niters, burnin, and thinning for each chain.
psrf	list containing `psrf`, a matrix of psrf estimates and upper limits for variable, and `mpsrf` the multivariate psrf estimate.

Author(s)

Christian Meyer

HLSMrandomEF *Function to run the MCMC sampler in random effects latent space model, HLSMfixedEF for fixed effects model, or LSM for single network latent space model)*

Description

Function to run the MCMC sampler to draw from the posterior distribution of intercept, slopes, and latent positions. HLSMrandomEF( ) fits random effects model; HLSMfixedEF( ) fits fixed effects model; LSM( ) fits single network model.

Usage

```
HLSMrandomEF(Y,edgeCov=NULL, receiverCov = NULL, senderCov = NULL,
        FullX = NULL,initialVals = NULL, priors = NULL, tune = NULL,
        tuneIn = TRUE,dd=2, niter, verbose=TRUE)

HLSMfixedEF(Y,edgeCov=NULL, receiverCov = NULL, senderCov = NULL,
        FullX = NULL, initialVals = NULL, priors = NULL, tune = NULL,
        tuneIn = TRUE,dd=2, estimate.intercept=FALSE, niter, verbose=TRUE)

LSM(Y,edgeCov=NULL, receiverCov = NULL, senderCov = NULL,
        FullX = NULL,initialVals = NULL, priors = NULL, tune = NULL,
        tuneIn = TRUE,dd=2, estimate.intercept=FALSE, niter, verbose=TRUE)

getBeta(object, burnin = 0, thin = 1)
getIntercept(object, burnin = 0, thin = 1)
getLS(object, burnin  = 0, thin = 1)
getLikelihood(object, burnin = 0, thin = 1)
```

Arguments

Y	input outcome for different networks. Y can either be (i). list of sociomatrices for K different networks (Y[[i]] must be a matrix with named rows and columns) (ii). list of data frame with columns Sender, Receiver and Outcome for K different networks (iii). a dataframe with columns named as follows: id to identify network, Receiver for receiver nodes, Sender for sender nodes and finally, Outcome for the edge outcome. Note that for LSM, Y must be an adjacency matrix.
edgeCov	data frame to specify edge level covariates with (i). a column for network id named id, (ii). a column for sender node named Sender, (iii). a column for receiver nodes named Receiver, and (iv). columns for values of each edge level covariates.

`receiverCov` a data frame to specify nodal covariates as edge receivers with
(i.) a column for network id named `id`,
(ii.) a column `Node` for node names, and
(iii). the rest for respective node level covariates.

`senderCov` a data frame to specify nodal covariates as edge senders with
(i). a column for network id named `id`,
(ii). a column `Node` for node names, and
(iii). the rest for respective node level covariates.

`FullX` list of numeric arrays of dimension `n` by `n` by `p` of covariates for K different networks. When FullX is provided to the function, edgeCov, receiverCov and senderCov must be specified as NULL.

`initialVals` an optional list of values to initialize the chain. If `NULL` default initialization is used, else `initialVals = list(ZZ, beta, intercept, alpha)`.
For fixed effect model `beta` is a vector of length `p` and `intercept` is a vector of length 1.
For random effect model `beta` is an array of dimension `K` by `p`, and `intercept` is a vector of length `K`, where `p` is the number of covariates and `K` is the number of network.
`ZZ` is an array of dimension `NN` by `dd`, where `NN` is the sum of nodes in all `K` networks.

`priors` an optional list to specify the hyper-parameters for the prior distribution of the paramters. If priors = `NULL`, default value is used. Else,
`priors=`
`list(MuBeta,VarBeta,MuZ,VarZ,PriorA,PriorB)`
`MuBeta` is a numeric vector of length PP + 1 specifying the mean of prior distribution for coefficients and intercept
`VarBeta` is a numeric vector for the variance of the prior distribution of coefficients and intercept. Its length is same as that of MuBeta.
`MuZ` is a numeric vector of length same as the dimension of the latent space, specifying the prior mean of the latent positions.
`VarZ` is a numeric vector of length same as the dimension of the latent space, specifying diagonal of the variance covariance matrix of the prior of latent positions.
`PriorA, PriorB` is a numeric variable to indicate the rate and scale parameters for the inverse gamma prior distribution of the hyper parameter of variance of slope and intercept

`tune` an optional list of tuning parameters for tuning the chain. If tune = `NULL`, default tuning is done. Else,
`tune = list(tuneBeta, tuneInt,tuneZ)`.
`tuneBeta` and `tuneInt` have the same structure as `beta` and `intercept` in `initialVals`.
`ZZ` is a vector of length `NN`.

`tuneIn` a logical to indicate whether tuning is needed in the MCMC sampling. Default is `FALSE`.

dd	dimension of latent space.
estimate.intercept	When TRUE, the intercept will be estimated. If the variance of the latent positions are of interest, intercept=FALSE will allow users to obtain a unique variance. The intercept can also be inputed by the user.
niter	number of iterations for the MCMC chain.
object	object of class 'HLSM' returned by HLSM() or HLSMfixedEF()
burnin	numeric value to burn the chain while extracting results from the 'HLSM'object. Default is burnin = 0.
thin	numeric value by which the chain is to be thinned while extracting results from the 'HLSM' object. Default is thin = 1.
verbose	logical value; TRUE results in messages during MCMC tuning

Details

The HLSMfixedEF and HLSMrandomEF functions will not automatically assess thinning and burn-in. To ensure appropriate inference, see HLSMdiag. See also LSM for fitting network data from a single network.

Value

Returns an object of class "HLSM". It is a list with following components:

draws	list of posterior draws for each parameters.
acc	list of acceptance rates of the parameters.
call	the matched call.
tune	final tuning values

Author(s)

Sam Adhikari & Tracy Sweet

References

Tracy M. Sweet, Andrew C. Thomas and Brian W. Junker (2013), "Hierarchical Network Models for Education Research: Hierarchical Latent Space Models", Journal of Educational and Behavorial Statistics.

Examples

```
library(HLSM)
data(schoolsAdviceData)

#Set values for the inputs of the function
priors = NULL
tune = NULL
initialVals = NULL
niter = 10
```

```
lsm.fit = LSM(Y=School9Network,edgeCov=School9EdgeCov,
senderCov=School9NodeCov, receiverCov=School9NodeCov, estimate.intercept=0, niter = niter)
```

---

schoolsAdviceData *HLSM: Included Data Sets*

Description

Data set included with the HLSM package: network variables from Pitts and Spillane (2009).

Usage

```
ps.advice.mat
ps.advice.df
ps.all.vars.mat
ps.edge.vars.mat
ps.edge.df
ps.school.vars.mat
ps.teacher.vars.mat
ps.node.df
School9Network
School9NodeCov
School9EdgeCov
```

Format

ps.advice.mat: a list of 15 sociomatrices of advice seeking network, one for each school.

ps.advice.df: a data frame of all ties.

ps.all.vars.mat: a list of 15 arrays of all the covariates, one for each school. edge.vars.mat: a list of edge level covariates for 15 different school.

ps.edge.df: a dataframe of all edge covariates.

ps.school.vars.mat: a list of school level covariates for all 15 schools.

ps.teacher.vars.mat: a list of node level covariates for all 15 schools.

ps.node.df: a dataframe of all node covariates.

ps.all.vars.mat: a single list of length 15 containing the covariates mentioned above.

School9Network: a single adjacency matrix from School 9.

School9NodeCov: a dataframe with node covariates

School9EdgeCov: a dataframe with dyad-level covariates.

Author(s)

Sam Adhikari and Tracy Sweet

References

Pitts, V., & Spillane, J. (2009). "Using social network methods to study school leadership".International Journal of Research & Method in Education, 32, 185-207

Sweet, T.M., Thomas, A.C., and Junker, B.W. (2013). "Hierarchical Network Models for Education Research: Hierarchical Latent Space Models". Journal of Educational and Behavorial Statistics.

Index