Package 'HLSM'

March 1, 2020

Type Package

Title Hierarchical Latent Space Network Model

Version 0.8.2

Date 2020-02-29

Author Samrachana Adhikari, Brian Junker, Tracy Sweet, Andrew C. Thomas

Maintainer Tracy Sweet <tsweet@umd.edu>

Description Implements Hierarchical Latent Space Network Model (HLSM) for ensemble of networks as described in Sweet, Thomas & Junker (2013). <DOI:10.3102/1076998612458702>.

Depends R (>= 3.0.0)

ByteCompile TRUE

License GPL (> 3)

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

LazyData yes

NeedsCompilation yes

Repository CRAN

Date/Publication 2020-03-01 07:00:06 UTC

R topics documented:

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. When returned to the console, a summary table of the diagnostics will be printed instead of the list representation of the object.

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 model (and HLSMfixedEF for fixed effects model)*

Description

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

Usage

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

HLSMfixedEF(Y,edgeCov=NULL, receiverCov = NULL, senderCov = NULL,
        FullX = NULL, initialVals = NULL, priors = NULL, tune = NULL,
        tuneIn = TRUE, TT = NULL,dd, niter)

getBeta(object, burnin = 0, thin = 1)
getIntercept(object, burnin = 0, thin = 1)
getAlpha(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 socio-matrixs 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.
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.
`alpha` is a numeric variable and is 0 for no-intervention model.

`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,MuAlpha,VarAlpha,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.
`MuAlpha` is a numeric variable specifying the mean of prior distribution of intervention effect. Default is 0.
`VarAlpha` is a numeric variable for the variance of the prior distribution of intervention effect. Default is 100.
`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(tuneAlpha,tuneBeta,tuneInt,tuneZ)`.
`tuneAlpha`, `tuneBeta` and `tuneInt` have the same structure as `beta`, `alpha` 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`.

TT	a vector of binaries to indicate treatment and control networks. If there is no intervention effect, TT = NULL (default).
dd	dimension of latent space.
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.

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

References

Tracy M. Sweet, Andrew C. Thomas and Brian W. Junker (2012), "Hierarchical Network Models for Education Research: Hierarchical Latent Space Models", Journal of Educational and Behavorial Statistics.

Examples

```
library(HLSM)

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

#Random effect HLSM on Pitt and Spillane data
random.fit = HLSMrandomEF(Y = ps.advice.mat,FullX = ps.edge.vars.mat,
initialVals = initialVals,priors = priors,
tune = tune,tuneIn = FALSE,dd = 2,niter = niter)


summary(random.fit)
names(random.fit)
```

```
#extract results without burning and thinning
Beta = getBeta(random.fit)
Intercept = getIntercept(random.fit)
LS = getLS(random.fit)
Likelihood = getLikelihood(random.fit)

##Same can be done for fixed effect model

#Fixed effect HLSM on Pitt and Spillane data

fixed.fit = HLSMfixedEF(Y = ps.advice.mat,FullX = ps.edge.vars.mat,
initialVals = initialVals,priors = priors,
tune = tune,tuneIn = FALSE,dd = 2,niter = niter)

summary(fixed.fit)
names(fixed.fit)
```

`plotDiagnostic` *built-in plot functions for HLSM object*

Description

A suite of functions for plotting HLSM model fits. HSLMcovplots is the most recent function to plot posterior distribution summaries. plotLikelihood( ) plots the likelihood, and plotDiagnostic( ) plots diagnostic-plot of posterior draws of the parameters from MCMC sample. plotHLSM.random.fit( ) and plotHLSM.fixed.fit( ) are functions to plot mean-results from fitted models, and plotHLSM.LS( ) is for plotting the mean latent position estimates.

Usage

```
plotLikelihood(object,burnin = 0, thin = 1)
plotDiagnostic(chain)
plotHLSM.random.fit(fitted.model,parameter,burnin=0,thin=1)
plotHLSM.fixed.fit(fitted.model, parameter,burnin=0,thin=1)
plotHLSM.LS(fitted.model,pdfname=NULL,burnin=0,thin=1,...)
HLSMcovplots(fitted.model, burnin=0, thin=1)
```

Arguments

`object`	object of class 'HLSM' obtained as an output from `HLSMrandomEF()` or `HLSMfixedEF()`
`fitted.model`	model fit from either HLSMrandomEF() or HLSMfixedEF()
`parameter`	parameter to plot; specified as `Beta` for slope coefficients, `Intercept` for intercept, and `Alpha` for intervention effect
`pdfname`	character to specify the name of the pdf to save the plot if desired. Default is NULL
`burnin`	numeric value to burn the chain for plotting the results from the 'HLSM'object

thin	a numeric thinning value
chain	a numeric vector of posterior draws of parameter of interest.
...	other options

Value

returns plot objects.

Author(s)

Sam Adhikari

Examples

```
#using advice seeking network of teachers in 15 schools
#to fit the data

#Random effect model#
priors = NULL
tune = NULL
initialVals = NULL
niter = 10

random.fit = HLSMrandomEF(Y = ps.advice.mat,FullX = ps.edge.vars.mat,
initialVals = initialVals,priors = priors,
tune = tune,tuneIn = FALSE,dd = 2,niter = niter)


HLSMcovplots(random.fit)


plotLikelihood(random.fit)
intercept = getIntercept(random.fit)
dim(intercept) ##is an array of dimension niter by 15
plotDiagnostic(intercept[,1])
plotHLSM.LS(random.fit)
plotHLSM.random.fit(random.fit,parameter = 'Beta')
plotHLSM.random.fit(random.fit,parameter = 'Intercept')
##look at the diagnostic plot of intercept for the first school
```

---

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.all.vars.mat
ps.edge.vars.mat
ps.school.vars.mat
ps.teacher.vars.mat
```

Format

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

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

Author(s)

Sam Adhikari

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. (2012). "Hierarchical Network Models for Education Research: Hierarchical Latent Space Models". Journal of Educational and Behavorial Statistics.

Index