Package ‘netcmc’

June 29, 2022

Type Package
Title Spatio-Network Generalised Linear Mixed Models for Areal Unit and Network Data
Version 1.0.1
Date 2022-06-28
Author George Gerogiannis, Mark Tranmer, Duncan Lee
Maintainer George Gerogiannis <g.gerogiannis.1@research.gla.ac.uk>
Description Implements a class of univariate and multivariate spatio-network generalised linear mixed models for areal unit and network data, with inference in a Bayesian setting using Markov chain Monte Carlo (MCMC) simulation. The response variable can be binomial, Gaussian, or Poisson. Spatial autocorrelation is modelled by a set of random effects that are assigned a conditional autoregressive (CAR) prior distribution following the Leroux model (Leroux et al. (2000) <doi:10.1007/978-1-4612-1284-3_4>). Network structures are modelled by a set of random effects that reflect a multiple membership structure (Browne et al. (2001) <doi:10.1177/1471082X0100100202>).
License GPL (>= 2)
Depends R (>= 4.0.0), MCMCpack
Imports Rcpp (>= 1.0.4), coda, ggplot2, mvtnorm, MASS
LinkingTo Rcpp, RcppArmadillo, RcppProgress
LazyLoad yes
Suggests testthat, igraph, magic
NeedsCompilation yes
Repository CRAN
Date/Publication 2022-06-29 00:20:09 UTC

R topics documented:

  netcmc-package .................................................. 2
  getAdjacencyMatrix ............................................. 2
  getMembershipMatrix .......................................... 4
  getTotalAltersByStatus ....................................... 6
An R Package for Bayesian Social Network Modelling

Description

Implements a class of univariate and multivariate spatio-network generalised linear mixed models, with inference in a Bayesian setting using Markov chain Monte Carlo (MCMC) simulation. The response variable can be binomial, Gaussian, and Poisson.

Details

Package: netcmc
Type: Package
Version: 1.0
Date: 2022-01-24
License: GPL (>= 2)

Author(s)

George Gerogiannis <g.gerogiannis.1@research.gla.ac.uk>

Examples

## See the examples in the function specific help files.

getAdjacencyMatrix A function that extracts valuable properties from a raw social network.
**getAdjacencyMatrix**

**Description**

This function transforms a network, which is a data.frame type in a specified format, into a resultant \( n \) by \( n \) adjacency matrix, where \( a_{ij} = 0 \) if vertex \( i \) and \( j \) \((i \neq j)\) are not adjacent i.e. vertex \( i \) and \( j \) are not the head/tail of an edge \( e \) and \( a_{ij} = 1 \) if vertex \( i \) and \( j \) \((i \neq j)\) are adjacent i.e. vertex \( i \) and \( j \) are the head/tail of an edge \( e \). \( a_{ij} = 0 \) when \( i = j \).

**Usage**

getAdjacencyMatrix(rawNetwork)

**Arguments**

- rawNetwork: The data.frame which encodes information about the network. The dimensions of the matrix are \( n \) by \((l+1)\). The data.frame contains one column corresponding to the labels for each of the \( n \) vertices in the network, the column name for this should be ‘labels’. The other \( l \) columns correspond to the vertices which are adjacent to each of the \( n \) vertices in the network. It is important to note that the label of a vertex should not be 0. The \( n \)th vertex can be adjacent to a maximum of \( l \) other vertices.

**Value**

- adjacencyMatrix: The resultant adjacency matrix for the rawNetwork data.frame.
- nonnominators: The individuals in the social network who are nominees of at least one other individual but were not in the set of individuals who did the nominating.
- vertexNoOutdegrees: The individuals in the social network that have an outdegree of 0.
- vertexNoIndegrees: The individuals in the social network that have an indegree of 0.
- vertexIsolates: The individuals in the social network that have an outdegree and indegree of 0.

**Author(s)**

George Gerogiannis

**Examples**

```r
rawNetwork = matrix(NA, 4, 3)
rawNetwork = as.data.frame(rawNetwork)
colnames(rawNetwork)[1] = "labels"
rawNetwork[, 1] = c("A", "B", "C", "D")
rawNetwork[, 2] = c(0, "C", "D", 0)
rawNetwork[, 3] = c("B", 0, "A", "C")
getAdjacencyMatrix(rawNetwork)

rawNetwork = matrix(NA, 4, 3)
```
getMembershipMatrix

A function that generates a data.frame that is the membership matrix of the network.
**getMembershipMatrix**

**Description**

A function that generates a data.frame that is the membership matrix of the network given individual IDs and the alters that they have nominated.

**Usage**

```r
getMembershipMatrix(individualID, alters)
```

**Arguments**

- `individualID`: A data.frame which stores the IDs of the individuals that nominate alters.
- `alters`: A data.frame which stores the alters of a given individual.

**Value**

`membershipMatrix`

The resultant data.frame.

**Author(s)**

George Gerogiannis

**Examples**

```r
individualID = data.frame(c(1, 2, 3))
alters = data.frame(c(5, 3, 2), c(5, 6, 1))
getMembershipMatrix(individualID, alters)

individualID = data.frame(c(1, 2, 3))
alters = data.frame(c(NA, 3, 2), c(NA, NA, 1))
getMembershipMatrix(individualID, alters)

individualID = data.frame(c(1, 2, 3))
alters = data.frame(c(NA, 3, NA), c(NA, NA, 1))
getMembershipMatrix(individualID, alters)

individualID = data.frame(c(1, 2, 3))
alters = data.frame(c(NA, 3, NA), c(6, NA, 1))
getMembershipMatrix(individualID, alters)
```
getTotalAltersByStatus

A function that generates a data.frame that stores the number of alters with a given level of a factor an individual has.

Description

This is a function that can be used to generate a data.frame that stores the number of alters with a given level of a factor an individual has.

Usage

getTotalAltersByStatus(individualID, status, alters)

Arguments

individualID  A data.frame which stores the IDs of the individuals that nominate alters.
status  A data.frame which stores the levels of a variable.
alters  A data.frame which stores the alters of a given individual.

Value

totalAltersByStatus  The resultant data.frame.

Author(s)

George Gerogiannis

Examples

individualID = data.frame(c(1, 2, 3, 4))
status = data.frame(c(10, 20, 30, 20))
alters = data.frame(c(4, 3, 2, 1), c(3, 4, 1, 2), c(2, 1, 4, 3))
totalAltersByStatus = getTotalAltersByStatus(individualID, status, alters)

individualID = data.frame(c(1, 2, 3, 4))
status = data.frame(c("RegularSmoke", "Nonsmoker", "CasualSmoker", "Nonsmoker"))
alters = data.frame(c(4, 3, 2, 1), c(3, 4, 1, 2), c(5, 1, 5, 3))
totalAltersByStatus = getTotalAltersByStatus(individualID, status, alters)

individualID = data.frame(c(1, 2, 3, 4))
status = data.frame(c(NA, "Nonsmoker", "CasualSmoker", "Nonsmoker"))
alters = data.frame(c(4, 3, 2, 1), c(3, 4, 1, 2), c(5, 1, 5, 3))
totalAltersByStatus = getTotalAltersByStatus(individualID, status, alters)

individualID = data.frame(c(10, 20))
status = data.frame(c(NA, "Nonsmoker"))
alters = data.frame(c(NA, 10), c(20, NA))
multiNet

A function that generates samples for a multivariate fixed effects and network model.

**Description**

This function that generates samples for a multivariate fixed effects and network model, which is given by

\[
Y_{i,s,r} | \mu_{i,s,r}, \sigma^2_{e_{r}} \sim f(y_{i,s,r}|\mu_{i,s,r},\sigma^2_{e_{r}}) \quad i = 1, \ldots, N_s, \ s = 1, \ldots, S, \ r = 1, \ldots, R,
\]

\[
g(\mu_{i,s,r}) = x_{i,s}^\top \beta_r + \sum_{j \in \text{net}(i_s)} w_{i,j} u_{j}^r + w_{i,s}^* u_{s}^r,
\]

\[
\beta_r \sim N(0, \alpha I)
\]

\[
u_j = (u_{1j}, \ldots, u_{Rj}) \sim N(0, \Sigma_u),
\]

\[
u^* = (u_{1}^*, \ldots, u_{R}^*) \sim N(0, \Sigma_u),
\]

\[
\Sigma_u \sim \text{Inverse-Wishart}(\xi_u, \Omega_u),
\]

\[
\sigma^2_{e_r} \sim \text{Inverse-Gamma}(\alpha_3, \xi_3).
\]

The covariates for the \(i\)th individual in the \(s\)th spatial unit or other grouping are included in a \(p \times 1\) vector \(x_{i,s}\). The corresponding \(p \times 1\) vector of fixed effect parameters relating to the \(r\)th response are denoted by \(\beta_r\), which has an assumed multivariate Gaussian prior with mean \(0\) and diagonal covariance matrix \(\alpha I\) that can be chosen by the user. A conjugate Inverse-Gamma prior is specified for \(\sigma^2_{e_r}\), and the corresponding hyperparameters (\(\alpha_3, \xi_3\)) can be chosen by the user.

The \(R \times 1\) vector of random effects for the \(j\)th alter is denoted by \(u_j = (u_{1j}, \ldots, u_{Rj})_{R \times 1}\), while the \(R \times 1\) vector of isolation effects for all \(R\) outcomes is denoted by \(u^* = (u_{1}^*, \ldots, u_{R}^*)\), and both are assigned multivariate Gaussian prior distributions. The unstructured covariance matrix \(\Sigma_u\) captures the covariance between the \(R\) outcomes at the network level, and a conjugate Inverse-Wishart prior is specified for this covariance matrix \(\Sigma_u\). The corresponding hyperparameters (\(\xi_u, \Omega_u\)) can be chosen by the user.

The exact specification of each of the likelihoods (binomial, Gaussian, and Poisson) are given below:

**Binomial:** \(Y_{i,s,r} \sim \text{Binomial}(n_{i,s,r}, \theta_{i,s,r})\) and \(g(\mu_{i,s,r}) = \ln(\theta_{i,s,r}/(1 - \theta_{i,s,r}))\),

**Gaussian:** \(Y_{i,s,r} \sim N(\mu_{i,s,r}, \sigma^2_{e_r})\) and \(g(\mu_{i,s,r}) = \mu_{i,s,r}\),

**Poisson:** \(Y_{i,s,r} \sim \text{Poisson}(\mu_{i,s,r})\) and \(g(\mu_{i,s,r}) = \ln(\mu_{i,s,r})\).
Usage

multiNet(formula, data, trials, family, W, numberOfSamples = 10, burnin = 0,
thin = 1, seed = 1, trueBeta = NULL, trueURandomEffects = NULL,
trueVarianceCovarianceU = NULL, trueSigmaSquaredE = NULL,
covarianceBetaPrior = 10^5, xi, omega, a3 = 0.001, b3 = 0.001,
centerURandomEffects = TRUE)

Arguments

formula A formula for the covariate part of the model using a similar syntax to that used
in the lm() function.
data An optional data.frame containing the variables in the formula.
trials A vector the same length as the response containing the total number of trials
$n_{i,r}$. Only used if family="binomial".
family The data likelihood model that must be “gaussian”, “poisson” or “binomial”.
W A matrix $W$ that encodes the social network structure and whose rows sum to
1.
numberOfSamples The number of samples to generate pre-thin.
burnin The number of MCMC samples to discard as the burn-in period.
thin The value by which to thin numberOfSamples.
seed A seed for the MCMC algorithm.
trueBeta If available, the true value of $\beta_1, \ldots, \beta_R$.
trueURandomEffects If available, the true values of $u_1, \ldots, u_J, u^*$.
trueVarianceCovarianceU If available, the true value of $\Sigma_u$.
trueSigmaSquaredE If available, the true value of $\sigma^2_{e1}, \ldots, \sigma^2_{eR}$. Only used if family="gaussian".
covarianceBetaPrior A scalar prior $\alpha$ for the covariance parameter of the beta prior, such that the
covariance is $\alpha I$.
xi The degrees of freedom parameter for the Inverse-Wishart distribution relating
to the network random effects $\xi_u$.
omega The scale parameter for the Inverse-Wishart distribution relating to the network
random effects $\Omega_u$.
a3 The shape parameter for the Inverse-Gamma distribution relating to the error
terms $\alpha_3$. Only used if family="gaussian".
b3 The scale parameter for the Inverse-Gamma distribution relating to the error
terms $\xi_3$. Only used if family="gaussian".
centerURandomEffects A choice to center the network random effects after each iteration of the MCMC
sampler.
**Value**

- `call` The matched call.
- `y` The response used.
- `X` The design matrix used.
- `standardizedX` The standardized design matrix used.
- `W` The network matrix used.
- `samples` The matrix of simulated samples from the posterior distribution of each parameter in the model (excluding random effects).
- `betaSamples` The matrix of simulated samples from the posterior distribution of $\beta_1, \ldots, \beta_R$ parameters in the model.
- `varianceCovarianceUSamples` The matrix of simulated samples from the posterior distribution of $\Sigma_u$ in the model.
- `uRandomEffectsSamples` The matrix of simulated samples from the posterior distribution of network random effects $u_1, \ldots, u_J, u^*$ in the model.
- `sigmaSquaredESamples` The vector of simulated samples from the posterior distribution of $\sigma^2 e_1, \ldots, \sigma^2 e_R$ in the model. Only used if `family`="gaussian".
- `acceptanceRates` The acceptance rates of parameters in the model from the MCMC sampling scheme.
- `uRandomEffectsAcceptanceRate` The acceptance rates of network random effects in the model from the MCMC sampling scheme.
- `timeTaken` The time taken for the model to run.
- `burnin` The number of MCMC samples to discard as the burn-in period.
- `thin` The value by which to thin `numberOfSamples`.
- `DBar` `DBar` for the model.
- `posteriorDeviance` The posterior deviance for the model.
- `posteriorLogLikelihood` The posterior log likelihood for the model.
- `pd` The number of effective parameters in the model.
- `DIC` The DIC for the model.

**Author(s)**

George Gerogiannis
multiNetLeroux

A function that generates samples for a multivariate fixed effects, spatial, and network model.

Description

This function that generates samples for a multivariate fixed effects, spatial, and network model, which is given by

\[
g(\mu_{isr}) = x_i^T \beta_r + \phi_{sr} + \sum_{r \in \text{net}(is)} w_{isr} u_{jr} + u_{is}^* u_{jr}^*,
\]

\[
\beta_r \sim N(0, \alpha I)
\]

\[
\phi_r = (\phi_{1r}, \ldots, \phi_{Sr}) \sim N(0, \tau_r^2 (\rho_r \text{diag}(A_1) - A) + (1 - \rho_r) I)^{-1},
\]

\[
u_j = (u_{1j}, \ldots, u_{Rj}) \sim N(0, \Sigma_u),
\]

\[
u^* = (u_{1}^*, \ldots, u_{R}^*) \sim N(0, \Sigma_u),
\]

\[
\tau_r^2 \sim \text{Inverse-Gamma}(a_1, b_1),
\]

\[
\rho_r \sim \text{Uniform}(0, 1),
\]

\[
\Sigma_u \sim \text{Inverse-Wishart}(\xi_u, \Omega_u),
\]

\[
\sigma_{err}^2 \sim \text{Inverse-Gamma}(\alpha_3, \xi_3).
\]

The covariates for the \(i\)th individual in the \(s\)th spatial unit or other grouping are included in a \(p \times 1\) vector \(x_{is}\). The corresponding \(p \times 1\) vector of fixed effect parameters relating to the \(r\)th response are denoted by \(\beta_r\), which has an assumed multivariate Gaussian prior with mean \(0\) and diagonal covariance matrix \(\alpha I\) that can be chosen by the user. A conjugate Inverse-Gamma prior is specified for \(\sigma_{err}^2\), and the corresponding hyperparamaterers \((\alpha_3, \xi_3)\) can be chosen by the user.

Spatial correlation in these areal unit level random effects is most often modelled by a conditional autoregressive (CAR) prior distribution. Using this model spatial correlation is induced into the random effects via a non-negative spatial adjacency matrix \(A = (a_{sl})_{S \times S}\), which defines how spatially close the \(S\) areal units are to each other. The elements of \(A_{S \times S}\) can be binary or non-binary, and the most common specification is that \(a_{sl} = 1\) if a pair of areal units \((G_s, G_l)\) share a common border or are considered neighbours by some other measure, and \(a_{sl} = 0\) otherwise. Note, \(a_{ss} = 0\) for all \(s\). \(\tau_r^2\) measures the variance of these random effects for the \(r\)th response, where a conjugate Inverse-Gamma prior is specified for \(\tau_r^2\) and the corresponding hyperparamaterers \((a_1, b_1)\) can be chosen by the user. \(\rho_r\) controls the level of spatial autocorrelation. A non-conjugate uniform prior is specified for \(\rho_r\).

The \(R \times 1\) vector of random effects for the \(j\)th alter is denoted by \(u_j = (u_{j1}, \ldots, u_{jR})_{R \times 1}\), while the \(R \times 1\) vector of isolation effects for all \(R\) outcomes is denoted by \(u^* = (u_1^*, \ldots, u_R^*)\), and both are assigned multivariate Gaussian prior distributions. The unstructured covariance matrix \(\Sigma_u\) captures the covariance between the \(R\) outcomes at the network level, and a conjugate Inverse-Wishart prior is specified for this covariance matrix \(\Sigma_u\). The corresponding hyperparamaterers \((\xi_u, \Omega_u)\) can be chosen by the user.
The exact specification of each of the likelihoods (binomial, Gaussian, and Poisson) are given below:

- **Binomial**: \( Y_{i,r} \sim \text{Binomial}(n_{i,r}, \theta_{i,r}) \) and \( g(\mu_{i,r}) = \ln(\theta_{i,r}/(1 - \theta_{i,r})) \),

- **Gaussian**: \( Y_{i,r} \sim N(\mu_{i,r}, \sigma^2_{er}) \) and \( g(\mu_{i,r}) = \mu_{i,r} \),

- **Poisson**: \( Y_{i,r} \sim \text{Poisson}(\mu_{i,r}) \) and \( g(\mu_{i,r}) = \ln(\mu_{i,r}) \).

**Usage**

```r
multiNetLeroux(formula, data, trials, family, squareSpatialNeighbourhoodMatrix, spatialAssignment, W, numberOfSamples = 10, burnin = 0, thin = 1, seed = 1, trueBeta = NULL, trueSpatialRandomEffects = NULL, trueURandomEffects = NULL, trueSpatialTauSquared = NULL, trueSpatialRho = NULL, trueVarianceCovarianceU = NULL, trueSigmaSquaredE = NULL, covarianceBetaPrior = 10^5, a1 = 0.001, b1 = 0.001, xi, omega, a3 = 0.001, b3 = 0.001, centerSpatialRandomEffects = TRUE, centerURandomEffects = TRUE)
```

**Arguments**

- `formula`: A formula for the covariate part of the model using a similar syntax to that used in the `lm()` function.
- `data`: An optional data.frame containing the variables in the formula.
- `trials`: A vector the same length as the response containing the total number of trials \( n_{i,r} \). Only used if `family` is “binomial”.
- `family`: The data likelihood model that must be “gaussian”, “poisson” or “binomial”.
- `squareSpatialNeighbourhoodMatrix`: An \( S \times S \) symmetric and non-negative neighbourhood matrix \( A = (a_{sl})_{S \times S} \).
- `W`: A matrix \( W \) that encodes the social network structure and whose rows sum to 1.
- `spatialAssignment`: The binary matrix of individual’s assignment to spatial area used in the model fitting process.
- `numberOfSamples`: The number of samples to generate pre-thin.
- `burnin`: The number of MCMC samples to discard as the burn-in period.
- `thin`: The value by which to thin `numberOfSamples`.
- `seed`: A seed for the MCMC algorithm.
- `trueBeta`: If available, the true value of \( \beta_1, \ldots, \beta_R \).
- `trueSpatialRandomEffects`: If available, the true values of \( \phi_1, \ldots, \phi_R \).
- `trueURandomEffects`: If available, the true values of \( u_1, \ldots, u_J, u^* \).
- `trueSpatialTauSquared`: If available, the true values of \( \tau_1^2, \ldots, \tau_R^2 \).
trueSpatialRho  If available, the true value of $\rho_1, \ldots, \rho_R$.

trueVarianceCovarianceU  
If available, the true value of $\Sigma_u$.

trueSigmaSquaredE  
If available, the true value of $\sigma^2_e, \ldots, \sigma^2_{eR}$. Only used if family="gaussian".

covarianceBetaPrior  
A scalar prior $\alpha$ for the covariance parameter of the beta prior, such that the covariance is $\alpha I$.

a1  
The shape parameter for the Inverse-Gamma distribution relating to the spatial random effects $\alpha_1$.

b1  
The scale parameter for the Inverse-Gamma distribution relating to the spatial random effects $\xi_1$.

xi  
The degrees of freedom parameter for the Inverse-Wishart distribution relating to the network random effects $\xi_u$.

omega  
The scale parameter for the Inverse-Wishart distribution relating to the network random effects $\Omega_u$.

a3  
The shape parameter for the Inverse-Gamma distribution relating to the error terms $\alpha_3$. Only used if family="gaussian".

b3  
The scale parameter for the Inverse-Gamma distribution relating to the error terms $\xi_3$. Only used if family="gaussian".

centerSpatialRandomEffects  
A choice to center the spatial random effects after each iteration of the MCMC sampler.

centerURandomEffects  
A choice to center the network random effects after each iteration of the MCMC sampler.

Value

call  The matched call.

y  The response used.

X  The design matrix used.

standardizedX  The standardized design matrix used.

squareSpatialNeighbourhoodMatrix  
The spatial neighbourhood matrix used.

spatialAssignment  
The spatial assignment matrix used.

W  The network matrix used.

samples  
The matrix of simulated samples from the posterior distribution of each parameter in the model (excluding random effects).

betaSamples  
The matrix of simulated samples from the posterior distribution of $\beta_1, \ldots, \beta_R$ parameters in the model.
spatialTauSquaredSamples
  Type: matrix. The matrix of simulated samples from the posterior distribution of $\tau_1^2, \ldots, \tau_R^2$ in the model.

spatialRhoSamples
  The vector of simulated samples from the posterior distribution of $\rho_1, \ldots, \rho_R$ in the model.

varianceCovarianceUSamples
  The matrix of simulated samples from the posterior distribution of $\Sigma_u$ in the model.

spatialRandomEffectsSamples
  The matrix of simulated samples from the posterior distribution of spatial random effects $\phi_1, \ldots, \phi_R$ in the model.

uRandomEffectsSamples
  The matrix of simulated samples from the posterior distribution of network random effects $u_1, \ldots, u_J, u^*$ in the model.

sigmaSquaredESamples
  The vector of simulated samples from the posterior distribution of $\sigma_e^2_1, \ldots, \sigma_e^2_R$ in the model. Only used if family="gaussian".

acceptanceRates
  The acceptance rates of parameters in the model from the MCMC sampling scheme.

spatialRandomEffectsAcceptanceRate
  The acceptance rates of spatial random effects in the model from the MCMC sampling scheme.

uRandomEffectsAcceptanceRate
  The acceptance rates of network random effects in the model from the MCMC sampling scheme.

timeTaken
  The time taken for the model to run.

burnin
  The number of MCMC samples to discard as the burn-in period.

thin
  The value by which to thin numberOfSamples.

DBar
  DBar for the model.

posteriorDeviance
  The posterior deviance for the model.

posteriorLogLikelihood
  The posterior log likelihood for the model.

pd
  The number of effective parameters in the model.

DIC
  The DIC for the model.

Author(s)

George Gerogiannis
multiNetRand

A function that generates samples for a multivariate fixed effects, grouping, and network model.

Description

This function that generates samples for a multivariate fixed effects, grouping, and network model, which is given by

\[
Y_{i,s,r} | \mu_{i,s,r} \sim f(y_{i,s,r} | \mu_{i,s,r}, \sigma_{er}^2) \quad i = 1, \ldots, N_s, \ s = 1, \ldots, S, \ r = 1, \ldots, R,
\]

\[
g(\mu_{i,s,r}) = x_i^T \beta_r v_{sr} + \sum_{j \in \text{net}(i_s)} w_{i,j} u_{jr} + w_{i,s}^* u_r,
\]

\[
\beta_r \sim \text{N}(0, \alpha I)
\]

\[
v_s = (v_{s1}, \ldots, v_{sR}) \sim \text{N}(0, \Sigma_v) v_s = (v_{s1}, \ldots, v_{sR}) \sim \text{N}(0, \Sigma_v),
\]

\[
u_j = (u_{1j}, \ldots, u_{Rj}) \sim \text{N}(0, \Sigma_u),
\]

\[
u^* = (u_{1}^*, \ldots, u_{R}^*) \sim \text{N}(0, \Sigma_u),
\]

\[
\Sigma_v \sim \text{Inverse-Wishart}(\xi_v, \Omega_v),
\]

\[
\Sigma_u \sim \text{Inverse-Wishart}(\xi_u, \Omega_u),
\]

\[
\sigma_{er}^2 \sim \text{Inverse-Gamma}(\alpha_3, \xi_3).
\]

The covariates for the \(i\)th individual in the \(s\)th spatial unit or other grouping are included in a \(p \times 1\) vector \(x_{i,s}\). The corresponding \(p \times 1\) vector of fixed effect parameters relating to the \(r\)th response are denoted by \(\beta_r\), which has an assumed multivariate Gaussian prior with mean \(0\) and diagonal covariance matrix \(\alpha I\) that can be chosen by the user. A conjugate Inverse-Gamma prior is specified for \(\sigma_{er}^2\), and the corresponding hyperparameters \((\alpha_3, \xi_3)\) can be chosen by the user.

The \(R \times 1\) vector of random effects for the \(s\)th group is denoted by \(v_s = (v_{s1}, \ldots, v_{sR})_{R \times 1}\), which is assigned a joint Gaussian prior distribution with an unstructured covariance matrix \(\Sigma_v\) that captures the covariance between the \(R\) outcomes. A conjugate Inverse-Wishart prior is specified for the random effects covariance matrix \(\Sigma_v\). The corresponding hyperparameters \((\xi_v, \Omega_v)\) can be chosen by the user.

The \(R \times 1\) vector of random effects for the \(j\)th alter is denoted by \(u_j = (u_{j1}, \ldots, u_{jR})_{R \times 1}\), while the \(R \times 1\) vector of isolation effects for all \(R\) outcomes is denoted by \(u^* = (u_{1}^*, \ldots, u_{R}^*)\), and both are assigned multivariate Gaussian prior distributions. The unstructured covariance matrix \(\Sigma_u\) captures the covariance between the \(R\) outcomes at the network level, and a conjugate Inverse-Wishart prior is specified for this covariance matrix \(\Sigma_u\). The corresponding hyperparameters \((\xi_u, \Omega_u)\) can be chosen by the user.

The exact specification of each of the likelihoods (binomial, Gaussian, and Poisson) are given below:

Binomial: \(Y_{i,s,r} \sim \text{Binomial}(n_{i,s,r}, \theta_{i,s,r})\) and \(g(\mu_{i,s,r}) = \ln(\theta_{i,s,r} / (1 - \theta_{i,s,r}))\),

Gaussian: \(Y_{i,s,r} \sim \text{N}(\mu_{i,s,r}, \sigma_{er}^2)\) and \(g(\mu_{i,s,r}) = \mu_{i,s,r}\),

Poisson: \(Y_{i,s,r} \sim \text{Poisson}(\mu_{i,s,r})\) and \(g(\mu_{i,s,r}) = \ln(\mu_{i,s,r})\).
multiNetRand

Usage
multiNetRand(formula, data, trials, family, V, W, numberOfSamples = 10, burnin = 0, thin = 1, seed = 1, trueBeta = NULL, trueVRandomEffects = NULL, trueURandomEffects = NULL, trueVarianceCovarianceV = NULL, trueVarianceCovarianceU = NULL, trueSigmaSquaredE = NULL, covarianceBetaPrior = 10^5, xiV, omegaV, xi, omega, a3 = 0.001, b3 = 0.001, centerVRandomEffects = TRUE, centerURandomEffects = TRUE)

Arguments

formula
A formula for the covariate part of the model using a similar syntax to that used in the lm() function.

data
An optional data.frame containing the variables in the formula.

trials
A vector the same length as the response containing the total number of trials \( n_{i,r} \). Only used if family="binomial".

family
The data likelihood model that must be “gaussian”, “poisson” or “binomial”.

V
The binary matrix of individual’s assignment to groups used in the model fitting process.

W
A matrix \( W \) that encodes the social network structure and whose rows sum to 1.

numberOfSamples
The number of samples to generate pre-thin.

burnin
The number of MCMC samples to discard as the burn-in period.

thin
The value by which to thin numberOfSamples.

seed
A seed for the MCMC algorithm.

trueBeta
If available, the true value of \( \beta_1, \ldots, \beta_R \).

trueVRandomEffects
If available, the true values of \( v_1, \ldots, v_S \).

trueURandomEffects
If available, the true values of \( u_1, \ldots, u_J, u^* \).

trueVarianceCovarianceV
If available, the true value of \( \Sigma_v \).

trueVarianceCovarianceU
If available, the true value of \( \Sigma_u \).

trueSigmaSquaredE
If available, the true value of \( \sigma^2_{e1}, \ldots, \sigma^2_{eR} \). Only used if family="gaussian".

covarianceBetaPrior
A scalar prior \( \alpha \) for the covariance parameter of the beta prior, such that the covariance is \( \alpha I \).

xiV
The degrees of freedom parameter for the Inverse-Wishart distribution relating to the grouping random effects \( \xi_v \).

omegaV
The scale parameter for the Inverse-Wishart distribution relating to the grouping random effects \( \Omega_v \).
The degrees of freedom parameter for the Inverse-Wishart distribution relating to the network random effects $\xi_u$.

The scale parameter for the Inverse-Wishart distribution relating to the network random effects $\Omega_u$.

The shape parameter for the Inverse-Gamma distribution relating to the error terms $\alpha_3$. Only used if family="gaussian".

The scale parameter for the Inverse-Gamma distribution relating to the error terms $\xi_3$. Only used if family="gaussian".

centerVRandomEffects  
A choice to center the spatial random effects after each iteration of the MCMC sampler.

centerURandomEffects  
A choice to center the network random effects after each iteration of the MCMC sampler.

Value

call  
The matched call.

y  
The response used.

X  
The design matrix used.

standardizedX  
The standardized design matrix used.

V  
The grouping assignment matrix used.

W  
The network matrix used.

samples  
The matrix of simulated samples from the posterior distribution of each parameter in the model (excluding random effects).

betaSamples  
The matrix of simulated samples from the posterior distribution of $\beta_1, \ldots, \beta_R$ parameters in the model.

varianceCovarianceVSamples  
The matrix of simulated samples from the posterior distribution of $\Sigma_v$ in the model.

varianceCovarianceUSamples  
The matrix of simulated samples from the posterior distribution of $\Sigma_u$ in the model.

vRandomEffectsSamples  
The matrix of simulated samples from the posterior distribution of spatial random effects $v_1, \ldots, v_S$ in the model.

uRandomEffectsSamples  
The matrix of simulated samples from the posterior distribution of network random effects $u_1, \ldots, u_J, u^*$ in the model.

sigmaSquaredESamples  
The vector of simulated samples from the posterior distribution of $\sigma^2_{\varepsilon_1}, \ldots, \sigma^2_{\varepsilon_R}$ in the model. Only used if family="gaussian".

acceptanceRates  
The acceptance rates of parameters in the model from the MCMC sampling scheme.
vRandomEffectsAcceptanceRate
   The acceptance rates of grouping random effects in the model from the MCMC sampling scheme.

uRandomEffectsAcceptanceRate
   The acceptance rates of network random effects in the model from the MCMC sampling scheme.

timeTaken
   The time taken for the model to run.

burnin
   The number of MCMC samples to discard as the burn-in period.

thin
   The value by which to thin numberOfSamples.

DBar
   DBar for the model.

posteriorDeviance
   The posterior deviance for the model.

posteriorLogLikelihood
   The posterior log likelihood for the model.

pd
   The number of effective parameters in the model.

DIC
   The DIC for the model.

Author(s)
   George Gerogiannis

plot.netcmc
   A function that plots visual MCMC diagnostics of the fitted model.

Description
   This function takes a netcmc object of samples from the posterior distribution of a parameter(s) and returns a visual convergence diaagnostics in the form of a density plot, trace plot, and ACF plot.

Usage
   ```
   ## S3 method for class 'netcmc'
   plot(x, ...)
   ```

Arguments
   
   x
      A netcmc object of samples from the posterior distribution of a parameter(s).

   ...  
      Ignored.

Value
   Returns a trace plot, density plot and ACF plot for the posterior distribution of a parameter(s) in a netcmc object.

Author(s)
   George Gerogiannis
### summary.netcmc

*Function that gets a summary of the fitted model.*

**Description**

This function takes a netcmc object and returns a summary of the fitted model. The summary includes, for selected parameters, posterior medians and 95 percent credible intervals, the effective number of independent samples and the Geweke convergence diagnostic in the form of a Z-score.

**Usage**

```r
## S3 method for class 'netcmc'
summary(object, ...)
```

**Arguments**

- `object` A netcmc fitted model object.
- `...` Ignored.

**Value**

Returns a model summary for a netcmc object.

**Author(s)**

George Gerogiannis

### uni

*Function that generates samples for a univariate fixed effects model.*

**Description**

This function generates samples for a univariate fixed effects model, which is given by

\[
Y_{is} | \mu_{is} \sim f(y_{is} | \mu_{is}, \sigma^2_e) \quad i = 1, \ldots, N_s, \quad s = 1, \ldots, S,
\]

\[
g(\mu_{is}) = x_{is}^T \beta,
\]

\[
\beta \sim N(0, \alpha I),
\]

\[
\sigma^2_e \sim \text{Inverse-Gamma}(\alpha_3, \xi_3).
\]

The covariates for the \(i\)th individual in the \(s\)th spatial unit or other grouping are included in a \(p \times 1\) vector \(x_{is}\). The corresponding \(p \times 1\) vector of fixed effect parameters are denoted by \(\beta\), which has an assumed multivariate Gaussian prior with mean \(0\) and diagonal covariance matrix \(\alpha I\) that can
be chosen by the user. A conjugate Inverse-Gamma prior is specified for $\sigma^2_e$, and the corresponding hyperparameters ($\alpha_3, \xi_3$) can be chosen by the user.

The exact specification of each of the likelihoods (binomial, Gaussian, and Poisson) are given below:

Binomial: $Y_{i_s} \sim \text{Binomial}(n_{i_s}, \theta_{i_s})$ and $g(\mu_{i_s}) = \ln(\theta_{i_s}/(1 - \theta_{i_s}))$,

Gaussian: $Y_{i_s} \sim \text{N}(\mu_{i_s}, \sigma^2_e)$ and $g(\mu_{i_s}) = \mu_{i_s}$,

Poisson: $Y_{i_s} \sim \text{Poisson}(\mu_{i_s})$ and $g(\mu_{i_s}) = \ln(\mu_{i_s})$.

**Usage**

`uni(formula, data, trials, family, numberOfSamples = 10, burnin = 0, thin = 1, seed = 1, trueBeta = NULL, trueSigmaSquaredE = NULL, covarianceBetaPrior = 10^5, a3 = 0.001, b3 = 0.001)`

**Arguments**

- `formula`: A formula for the covariate part of the model using a similar syntax to that used in the `lm()` function.
- `data`: An optional data.frame containing the variables in the formula.
- `trials`: A vector the same length as the response containing the total number of trials $n_{i_s}$. Only used if `family`="binomial".
- `family`: The data likelihood model that must be “gaussian”, “poisson” or “binomial".
- `numberOfSamples`: The number of samples to generate pre-thin.
- `burnin`: The number of MCMC samples to discard as the burn-in period.
- `thin`: The value by which to thin `numberOfSamples`.
- `seed`: A seed for the MCMC algorithm.
- `trueBeta`: If available, the true values of the $\beta$.
- `trueSigmaSquaredE`: If available, the true value of $\sigma^2_e$. Only used if `family`="gaussian".
- `covarianceBetaPrior`: A scalar prior $\alpha$ for the covariance parameter of the beta prior, such that the covariance is $\alpha I$.
- `a3`: The shape parameter for the Inverse-Gamma distribution $\alpha_3$. Only used if `family`="gaussian".
- `b3`: The scale parameter for the Inverse-Gamma distribution $\xi_3$. Only used if `family`="gaussian".

**Value**

- `call`: The matched call.
- `y`: The response used.
- `X`: The design matrix used.
The standardized design matrix used.

The matrix of simulated samples from the posterior distribution of each parameter in the model (excluding random effects).

The matrix of simulated samples from the posterior distribution of $\beta$ parameters in the model.

The vector of simulated samples from the posterior distribution of $\sigma^2$ in the model.

The acceptance rates of parameters in the model from the MCMC sampling scheme.

The time taken for the model to run.

The number of MCMC samples to discard as the burn-in period.

The value by which to thin numberOfSamples.

DBar for the model.

The posterior deviance for the model.

The posterior log likelihood for the model.

The number of effective parameters in the model.

The DIC for the model.

Author(s)

George Gerogiannis

Examples

```r
#### Run the model on simulated data
 observations <- 100
 X <- matrix(rnorm(2 * observations), ncol = 2)
 colnames(X) <- c("x1", "x2")
 beta <- c(2, -2, 2)
 logit <- cbind(rep(1, observations), X) %*% beta
 prob <- exp(logit) / (1 + exp(logit))
 trials <- rep(50, observations)
 Y <- rbinom(n = observations, size = trials, prob = prob)
 data <- data.frame(cbind(Y, X))

#### Generate the covariates and response data

#### Run the model
 formula <- Y ~ x1 + x2

# Not run: model <- uni(formula = formula, data = data, family="binomial",
 trials = trials, numberOfSamples = 10000,
 burnin = 10000, thin = 10, seed = 1)
```
uniNet

A function that generates samples for a univariate network model.

Description

This function generates samples for a univariate network model, which is given by

\[ Y_{is} | \mu_{is} \sim f(y_{is} | \mu_{is}, \sigma^2_e) \quad i = 1, \ldots, N_s, \ s = 1, \ldots, S, \]

\[ g(\mu_{is}) = x_{is}^T \beta + \sum_{j \in \text{net}(i_s)} w_{is} j u_j + w^*_{is} u^*, \]

\[ \beta \sim N(0, \alpha I), \]

\[ u_j \sim N(0, \sigma^2_u), \]

\[ u^* \sim N(0, \sigma^2_u), \]

\[ \sigma^2_u \sim \text{Inverse-Gamma}(\alpha_2, \xi_2), \]

\[ \sigma^2_e \sim \text{Inverse-Gamma}(\alpha_3, \xi_3). \]

The covariates for the \( i \)-th individual in the \( s \)-th spatial unit or other grouping are included in a \( p \times 1 \) vector \( x_{is} \). The corresponding \( p \times 1 \) vector of fixed effect parameters are denoted by \( \beta \), which has an assumed multivariate Gaussian prior with mean 0 and diagonal covariance matrix \( \alpha I \) that can be chosen by the user. A conjugate Inverse-Gamma prior is specified for \( \sigma^2_e \), and the corresponding hyperparameters \((\alpha_3, \xi_3)\) can be chosen by the user.

The \( J \times 1 \) vector of alter random effects are denoted by \( u = (u_1, \ldots, u_J)_{J \times 1} \) and modelled as independently Gaussian with mean zero and a constant variance, and due to the row standardised nature of \( W \), \( \sum_{j \in \text{net}(i_s)} w_{is} j u_j \) represents the average (mean) effect that the peers of individual \( i \) in spatial unit or group \( s \) have on that individual. \( w^*_{is} \) \( u^* \) is an isolation effect, which is an effect for individuals who don’t nominate any friends. This is achieved by setting \( w^*_{is} = 1 \) if individual \( i_s \) nominates no peers and \( w^*_{is} = 0 \) otherwise, and if \( w^*_{is} = 1 \) then clearly \( \sum_{j \in \text{net}(i_s)} w_{is} j u_j = 0 \) as \( \text{net}(i_s) \) is the empty set. A conjugate Inverse-Gamma prior is specified for the random effects variance \( \sigma^2_u \), and the corresponding hyperparameters \((\alpha_2, \xi_2)\) can be chosen by the user.

The exact specification of each of the likelihoods (binomial, Gaussian, and Poisson) are given below:

**Binomial:** \( Y_{is} \sim \text{Binomial}(n_{is}, \theta_{is}) \) and \( g(\mu_{is}) = \ln(\theta_{is}/(1 - \theta_{is})) \),

**Gaussian:** \( Y_{is} \sim N(\mu_{is}, \sigma^2_e) \) and \( g(\mu_{is}) = \mu_{is} \),

**Poisson:** \( Y_{is} \sim \text{Poisson}(\mu_{is}) \) and \( g(\mu_{is}) = \ln(\mu_{is}) \).
uniNet(formula, data, trials, family, W, numberOfSamples = 10, burnin = 0, thin = 1, seed = 1, trueBeta = NULL, trueURandomEffects = NULL, trueSigmaSquaredU = NULL, trueSigmaSquaredE = NULL, covarianceBetaPrior = 10^5, a2 = 0.001, b2 = 0.001, a3 = 0.001, b3 = 0.001, centerURandomEffects = TRUE)

Arguments

formula  A formula for the covariate part of the model using a similar syntax to that used in the lm() function.
data  An optional data.frame containing the variables in the formula.
trials  A vector the same length as the response containing the total number of trials $n_i$. Only used if family="binomial".
family  The data likelihood model that must be “gaussian”, “poisson” or “binomial”.
W  A matrix $W$ that encodes the social network structure and whose rows sum to 1.
numberOfSamples  The number of samples to generate pre-thin.
burnin  The number of MCMC samples to discard as the burn-in period.
thin  The value by which to thin numberOfSamples.
seed  A seed for the MCMC algorithm.
trueBeta  If available, the true value of $\beta$.
trueURandomEffects  If available, the true value of $u$.
trueSigmaSquaredU  If available, the true value $\sigma_u^2$.
trueSigmaSquaredE  If available, the true value $\sigma_e^2$.
covarianceBetaPrior  A scalar prior $\alpha$ for the covariance parameter of the beta prior, such that the covariance is $\alpha I$.
a2  The shape parameter for the Inverse-Gamma distribution relating to the network random effects $\alpha_2$.
b2  The scale parameter for the Inverse-Gamma distribution relating to the network random effects $\xi_2$.
a3  The shape parameter for the Inverse-Gamma distribution relating to the error terms $\alpha_3$. Only used if family="gaussian".
b3  The scale parameter for the Inverse-Gamma distribution relating to the error terms $\xi_3$. Only used if family="gaussian".
centerURandomEffects  A choice to center the network random effects after each iteration of the MCMC sampler.
Value

call The matched call.
y The response used.
X The design matrix used.
standardizedX The standardized design matrix used.
W The network matrix used.
samples The matrix of simulated samples from the posterior distribution of each parameter in the model (excluding random effects).
betaSamples The matrix of simulated samples from the posterior distribution of $\mathbf{\beta}$ parameters in the model.
sigmaSquaredUSamples The vector of simulated samples from the posterior distribution of $\sigma^2_u$ in the model.
sigmaSquaredESamples The vector of simulated samples from the posterior distribution of $\sigma^2_e$ in the model.
uRandomEffectsSamples The matrix of simulated samples from the posterior distribution of network random effects $u$ in the model.
acceptanceRates The acceptance rates of parameters in the model (excluding random effects) from the MCMC sampling scheme.
uRandomEffectsAcceptanceRate The acceptance rates of network random effects in the model from the MCMC sampling scheme.
timeTaken The time taken for the model to run.
burnin The number of MCMC samples to discard as the burn-in period.
thin The value by which to thin numberOfSampling.
DBar DBar for the model.
pD The posterior deviance for the model.
pDBar The posterior log likelihood for the model.
pd The number of effective parameters in the model.
DIC The DIC for the model.

Author(s)

George Gerogiannis
### Examples

#### Run the model on simulated data

#### Load other libraries required

```r
library(MCMCpack)
```

#### Set up a network

```r
observations <- 200
numberOfMultipleClassifications <- 50
W <- matrix(rbinom(observations * numberOfMultipleClassifications, 1, 0.05),
ncol = numberOfMultipleClassifications)
numberofActorsWithNoPeers <- sum(apply(W, 1, function(x) { sum(x) == 0 })))
peers <- sample(1:numberOfMultipleClassifications, numberOfActorsWithNoPeers,
TRUE)
actorsWithNoPeers <- which(apply(W, 1, function(x) { sum(x) == 0 })))
for(i in 1:numberofActorsWithNoPeers) {
  W[actorsWithNoPeers[i], peers[i]] <- 1
}
W <- t(apply(W, 1, function(x) { x / sum(x) })))
```

#### Generate the covariates and response data

```r
X <- matrix(rnorm(2 * observations), ncol = 2)
colnames(X) <- c("x1", "x2")
beta <- c(1, -0.5, 0.5)
sigmaSquaredU <- 1
uRandomEffects <- rnorm(numberOfMultipleClassifications, mean = 0,
sd = sqrt(sigmaSquaredU))
logTheta <- cbind(rep(1, observations), X) %*% beta + W %*% uRandomEffects
Y <- rpois(n = observations, lambda = exp(logTheta))
data <- data.frame(cbind(Y, X))
```

#### Run the model

```r
formula <- Y ~ x1 + x2
## Not run: model <- uniNet(formula = formula, data = data, family="poisson",
W = W, numberOfSamples = 10000, burnin = 10000,
thin = 10, seed = 1)
## End(Not run)
```

---

### uniNetLeroux

A function that generates samples for a univariate network Leroux model.

### Description

This function generates samples for a univariate network Leroux model, which is given by

\[ Y_i | \mu_i, \sim f(y_i | \mu_i, \sigma^2_c) \quad i = 1, \ldots, N_s, \quad s = 1, \ldots, S, \]
\[ g(\mu_i) = \mathbf{x}_{i}^{\top} \beta + \phi_s + \sum_{j \in \text{net}(i_s)} w_{i,j} u_j + w_{i,*} u^*, \]

\[ \beta \sim N(0, \alpha I), \]

\[ \phi_s | \phi_{-s} \sim N \left( \frac{\rho \sum_{l=1}^{S} a_{sl} \phi_l}{\rho \sum_{l=1}^{S} a_{sl} + 1 - \rho}, \frac{\tau^2}{\rho \sum_{l=1}^{S} a_{sl} + 1 - \rho} \right), \]

\[ u_j \sim N(0, \sigma^2_u), \]

\[ u^* \sim N(0, \sigma^2_u), \]

\[ \tau^2 \sim \text{Inverse-Gamma}(\alpha_1, \xi_1), \]

\[ \rho \sim \text{Uniform}(0, 1), \]

\[ \sigma^2_u \sim \text{Inverse-Gamma}(\alpha_2, \xi_2), \]

\[ \sigma^2_\tau \sim \text{Inverse-Gamma}(\alpha_3, \xi_3). \]

The covariates for the \( i \)th individual in the \( s \)th spatial unit or other grouping are included in a \( p \times 1 \) vector \( \mathbf{x}_{i,s} \). The corresponding \( p \times 1 \) vector of fixed effect parameters are denoted by \( \beta \), which has an assumed multivariate Gaussian prior with mean 0 and diagonal covariance matrix \( \alpha I \) that can be chosen by the user. A conjugate Inverse-Gamma prior is specified for \( \sigma^2_u \), and the corresponding hyperparameters \((\alpha_3, \xi_3)\) can be chosen by the user.

Spatial correlation in these areal unit level random effects is most often modelled by a conditional autoregressive (CAR) prior distribution. Using this model spatial correlation is induced into the random effects via a non-negative spatial adjacency matrix \( A = (a_{sl})_{S \times S} \), which defines how spatially close the \( S \) areal units are to each other. The elements of \( A_{S \times S} \) can be binary or non-binary, and the most common specification is that \( a_{sl} = 1 \) if a pair of areal units \((G_s, G_l)\) share a common border or are considered neighbours by some other measure, and \( a_{sl} = 0 \) otherwise. Note, \( a_{ss} = 0 \) for all \( s \).

\( \phi_{-s} = (\phi_1, \ldots, \phi_{s-1}, \phi_{s+1}, \ldots, \phi_S) \). Here \( \tau^2 \) is a measure of the variance relating to the spatial random effects \( \phi \), while \( \rho \) controls the level of spatial autocorrelation, with values close to one and zero representing strong autocorrelation and independence respectively. A non-conjugate uniform prior on the unit interval is specified for the single level of spatial autocorrelation \( \rho \). In contrast, a conjugate Inverse-Gamma prior is specified for the random effects variance \( \tau^2 \), and corresponding hyperparameters \((\alpha_1, \xi_1)\) can be chosen by the user.

The \( J \times 1 \) vector of alter random effects are denoted by \( \mathbf{u} = (u_1, \ldots, u_J)_{J \times 1} \) and modelled as independently Gaussian with mean zero and a constant variance, and due to the row standardised nature of \( W \), \( \sum_{j \in \text{net}(i_s)} w_{i,j} u_j \) represents the average (mean) effect that the peers of individual \( i \) in spatial unit or group \( s \) have on that individual. \( w_{i,*} u^* \) is an isolation effect, which is an effect for individuals who don’t nominate any friends. This is achieved by setting \( w_{i,*} u^* = 1 \) if individual \( i_s \) nominates no peers and \( w_{i,*} u^* = 0 \) otherwise, and if \( w_{i,*} = 1 \) then clearly \( \sum_{j \in \text{net}(i_s)} w_{i,j} u_j = 0 \) as \( \text{net}(i_s) \) is the empty set. A conjugate Inverse-Gamma prior is specified for the random effects variance \( \sigma^2_u \), and the corresponding hyperparameters \((\alpha_2, \xi_2)\) can be chosen by the user.

The exact specification of each of the likelihoods (binomial, Gaussian, and Poisson) are given below:

**Binomial:** \( Y_{i,s} \sim \text{Binomial}(n_{i,s}, \theta_{i,s}) \) and \( g(\mu_i) = \ln(\theta_{i,s}/(1 - \theta_{i,s})) \),

**Gaussian:** \( Y_{i,s} \sim \text{N}(\mu_{i,s}, \sigma^2_\tau) \) and \( g(\mu_{i,s}) = \mu_{i,s} \),

**Poisson:** \( Y_{i,s} \sim \text{Poisson}(\mu_{i,s}) \) and \( g(\mu_{i,s}) = \ln(\mu_{i,s}) \).
uniNetLeroux(formula, data, trials, family, squareSpatialNeighbourhoodMatrix, spatialAssignment, W, numberOfSamples = 10, burnin = 0, thin = 1, seed = 1, trueBeta = NULL, trueSpatialRandomEffects = NULL, trueURandomEffects = NULL, trueSpatialTauSquared = NULL, trueSpatialRho = NULL, trueSigmaSquaredU = NULL, trueSigmaSquaredE = NULL, covarianceBetaPrior = 10^5, a1 = 0.001, b1 = 0.001, a2 = 0.001, b2 = 0.001, a3 = 0.001, b3 = 0.001, centerSpatialRandomEffects = TRUE, centerURandomEffects = TRUE)

Arguments

- **formula**: A formula for the covariate part of the model using a similar syntax to that used in the `lm()` function.
- **data**: An optional data.frame containing the variables in the formula.
- **trials**: A vector the same length as the response containing the total number of trials \( n_i \). Only used if `family` = “binomial”.
- **family**: The data likelihood model that must be “gaussian”, “poisson” or “binomial”.
- **squareSpatialNeighbourhoodMatrix**: An \( S \times S \) symmetric and non-negative neighbourhood matrix \( A = (a_{ij})_{S \times S} \).
- **W**: A matrix \( W \) that encodes the social network structure and whose rows sum to 1.
- **spatialAssignment**: The binary matrix of individual’s assignment to spatial area used in the model fitting process.
- **numberOfSamples**: The number of samples to generate pre-thin.
- **burnin**: The number of MCMC samples to discard as the burn-in period.
- **thin**: The value by which to thin `numberOfSamples`.
- **seed**: A seed for the MCMC algorithm.
- **trueBeta**: If available, the true value of \( \beta \).
- **trueSpatialRandomEffects**: If available, the true value of \( \phi \).
- **trueURandomEffects**: If available, the true value of \( u \).
- **trueSpatialTauSquared**: If available, the true value of \( \tau^2 \).
- **trueSpatialRho**: If available, the true value of \( \rho \).
- **trueSigmaSquaredU**: If available, the true value of \( \sigma^2_u \).
- **trueSigmaSquaredE**: If available, the true value of \( \sigma^2_e \).
- **covarianceBetaPrior**: A scalar prior \( \alpha \) for the covariance parameter of the beta prior, such that the covariance is \( \alpha I \).
a1  The shape parameter for the Inverse-Gamma distribution relating to the spatial random effects $\alpha_1$.

b1  The scale parameter for the Inverse-Gamma distribution relating to the spatial random effects $\xi_1$.

a2  The shape parameter for the Inverse-Gamma distribution relating to the network random effects $\alpha_2$.

b2  The scale parameter for the Inverse-Gamma distribution relating to the network random effects $\xi_2$.

a3  The shape parameter for the Inverse-Gamma distribution relating to the error terms $\alpha_3$. Only used if family="gaussian".

b3  The scale parameter for the Inverse-Gamma distribution relating to the error terms $\xi_3$. Only used if family="gaussian".

centerSpatialRandomEffects
   A choice to center the spatial random effects after each iteration of the MCMC sampler.

centerURandomEffects
   A choice to center the network random effects after each iteration of the MCMC sampler.

Value

call  The matched call.

y  The response used.

X  The design matrix used.

standardizedX  The standardized design matrix used.

squareSpatialNeighbourhoodMatrix  The spatial neighbourhood matrix used.

spatialAssignment  The spatial assignment matrix used.

W  The network matrix used.

samples  The matrix of simulated samples from the posterior distribution of each parameter in the model (excluding random effects).

betaSamples  The matrix of simulated samples from the posterior distribution of $\beta$ parameters in the model.

spatialTauSquaredSamples  The vector of simulated samples from the posterior distribution of $\tau^2$ in the model.

spatialRhoSamples  The vector of simulated samples from the posterior distribution of $\rho$ in the model.

sigmaSquaredUSamples  The vector of simulated samples from the posterior distribution of $\sigma_u^2$ in the model.
The vector of simulated samples from the posterior distribution of $\sigma^2_e$ in the model.

The matrix of simulated samples from the posterior distribution of spatial/grouping random effects $\phi$ in the model.

The matrix of simulated samples from the posterior distribution of network random effects $u$ in the model.

The acceptance rates of parameters in the model (excluding random effects) from the MCMC sampling scheme.

The acceptance rates of spatial/grouping random effects in the model from the MCMC sampling scheme.

The acceptance rates of network random effects in the model from the MCMC sampling scheme.

The time taken for the model to run.

The number of MCMC samples to discard as the burn-in period.

The value by which to thin `numberOfSamples`.

$DBar$ for the model.

The posterior deviance for the model.

The posterior log likelihood for the model.

The number of effective parameters in the model.

The DIC for the model.

Author(s)

George Gerogiannis

Examples

```r
#############################################################
### Run the model on simulated data
#############################################################
### Load other libraries required
library(MCMCpack)

### Set up a network
observations <- 200
numberOfMultipleClassifications <- 50
W <- matrix(rbinom(observations * numberOfMultipleClassifications, 1, 0.05),
ncol = numberOfMultipleClassifications)
numberOfActorsWithNoPeers <- sum(apply(W, 1, function(x) { sum(x) == 0 })))
```
peers <- sample(1:numberOfMultipleClassifications, numberofActorsWithNoPeers, TRUE)
actorsWithNoPeers <- which(apply(W, 1, function(x) { sum(x) == 0 })))
for(i in 1:numberOfActorsWithNoPeers) {
W[actorsWithNoPeers[i], peers[i]] <- 1
}
W <- t(apply(W, 1, function(x) { x / sum(x) }))

#### Set up a spatial structure
numberOfSpatialAreas <- 100
factor = sample(1:numberOfSpatialAreas, observations, TRUE)
spatialAssignment = matrix(NA, ncol = numberOfSpatialAreas, nrow = observations)
for(i in 1:length(factor)){
for(j in 1:numberOfSpatialAreas){
if(factor[i] == j){
   spatialAssignment[i, j] = 1
} else {
   spatialAssignment[i, j] = 0
}
}
}

gridAxis = sqrt(numberOfSpatialAreas)
easting = 1:gridAxis
northing = 1:gridAxis
grid = expand.grid(easting, northing)
numberOfRowsInGrid = nrow(grid)
distance = as.matrix(dist(grid))
squareSpatialNeighbourhoodMatrix = array(0, c(numberOfRowsInGrid, numberOfRowsInGrid))
squareSpatialNeighbourhoodMatrix[distance==1] = 1

#### Generate the covariates and response data
X <- matrix(rnorm(2 * observations), ncol = 2)
colnames(X) <- c("x1", "x2")
beta <- c(2, -2, 2)

spatialRho <- 0.5
spatialTauSquared <- 2
spatialPrecisionMatrix = spatialRho * 
   (diag(apply(squareSpatialNeighbourhoodMatrix, 1, sum)) - 
   squareSpatialNeighbourhoodMatrix) + (1 - spatialRho) * 
   diag(rep(1, numberOfSpatialAreas))
spatialCovarianceMatrix = solve(spatialPrecisionMatrix)
spatialPhi = mvrnorm(n = 1, mu = rep(0, numberOfSpatialAreas), 
   Sigma = (spatialTauSquared * spatialCovarianceMatrix))

sigmaSquaredU <- 2
uRandomEffects <- rnorm(numberOfMultipleClassifications, mean = 0, 
   sd = sqrt(sigmaSquaredU))

logit <- cbind(rep(1, observations), X) %*% beta +
uniNetRand

A function that generates samples for a univariate network group model.

Description

This function generates samples for a univariate network group model, which is given by

\[ Y_{is} | \mu_{is} \sim f(y_{is} | \mu_{is}, \sigma^2_e) \quad i = 1, \ldots, N_s, \ s = 1, \ldots, S, \]

\[ g(\mu_{is}) = x_{is}^\top \beta + v_s + \sum_{j \in \text{net}(is)} w_{isj} u_j + w^*_{is} u^*, \]

\[ \beta \sim N(0, \alpha I), \]

\[ v_s \sim N(0, \tau^2), \]

\[ u_j \sim N(0, \sigma^2_u), \]

\[ u^* \sim N(0, \sigma^2_u), \]

\[ \tau^2 \sim \text{Inverse-Gamma}(\alpha_1, \xi_1), \]

\[ \sigma^2_e \sim \text{Inverse-Gamma}(\alpha_2, \xi_2), \]

\[ \sigma^2_u \sim \text{Inverse-Gamma}(\alpha_3, \xi_3). \]

The covariates for the \( i \)th individual in the \( s \)th spatial unit or other grouping are included in a \( p \times 1 \) vector \( x_{is} \). The corresponding \( p \times 1 \) vector of fixed effect parameters are denoted by \( \beta \), which has an assumed multivariate Gaussian prior with mean 0 and diagonal covariance matrix \( \alpha I \) that can be chosen by the user. A conjugate Inverse-Gamma prior is specified for \( \sigma^2_e \), and the corresponding hyperparameters \( (\alpha_3, \xi_3) \) can be chosen by the user.

The \( S \times 1 \) vector of random effects for the groups are collectively denoted by \( v = (v_1, \ldots, v_S)_{S \times 1} \), and each element is assigned an independent zero-mean Gaussian prior distribution with a constant
variance $\tau^2$. A conjugate Inverse-Gamma prior is specified for $\tau^2$. The corresponding hyperparameters $(\alpha_1, \xi_1)$ can be chosen by the user.

The $J \times 1$ vector of alter random effects are denoted by $u = (u_1, \ldots, u_J)_{J \times 1}$ and modelled as independently Gaussian with mean zero and a constant variance, and due to the row standardised nature of $W$, $\sum_{j \in \text{net}(is)} w_{is,j} u_j$ represents the average (mean) effect that the peers of individual $i$ in spatial unit or group $s$ have on that individual. $w^*_{is} u^*$ is an isolation effect, which is an effect for individuals who don’t nominate any friends. This is achieved by setting $w^*_{is} = 1$ if individual $i_s$ nominates no peers and $w^*_{is} = 0$ otherwise, and if $w^*_{is} = 1$ then clearly $\sum_{j \in \text{net}(is)} w_{is,j} u_{jr} = 0$ as net$(is)$ is the empty set. A conjugate Inverse-Gamma prior is specified for the random effects variance $\sigma^2_u$, and the corresponding hyperparameters $(\alpha_2, \xi_2)$ can be chosen by the user.

The exact specification of each of the likelihoods (binomial, Gaussian, and Poisson) are given below:

- **Binomial**: $Y_{is} \sim \text{Binomial}(n_{is}, \theta_{is})$ and $g(\mu_{is}) = \ln(\theta_{is}/(1 - \theta_{is}))$,
- **Gaussian**: $Y_{is} \sim \text{N}(\mu_{is}, \sigma^2_e)$ and $g(\mu_{is}) = \mu_{is}$,
- **Poisson**: $Y_{is} \sim \text{Poisson}(\mu_{is})$ and $g(\mu_{is}) = \ln(\mu_{is})$.

**Usage**

```r
uniNetRand(formula, data, trials, family, groupAssignment, W, numberOfSamples = 10, burnin = 0, thin = 1, seed = 1, trueBeta = NULL, trueGroupRandomEffects = NULL, trueURandomEffects = NULL, trueTauSquared = NULL, trueSigmaSquaredU = NULL, trueSigmaSquaredE = NULL, covarianceBetaPrior = 10^5, a1 = 0.001, b1 = 0.001, a2 = 0.001, b2 = 0.001, a3 = 0.001, b3 = 0.001, centerGroupRandomEffects = TRUE, centerURandomEffects = TRUE)
```

**Arguments**

- **formula**: A formula for the covariate part of the model using a similar syntax to that used in the `lm()` function.
- **data**: An optional data.frame containing the variables in the formula.
- **trials**: A vector the same length as the response containing the total number of trials $n_{is}$. Only used if `family`="binomial".
- **family**: The data likelihood model that must be “gaussian”, “poisson” or “binomial”.
- **W**: A matrix $W$ that encodes the social network structure and whose rows sum to 1.
- **groupAssignment**: The binary matrix of individual’s assignment to groups used in the model fitting process.
- **numberOfSamples**: The number of samples to generate pre-thin.
- **burnin**: The number of MCMC samples to discard as the burn-in period.
- **thin**: The value by which to thin `numberOfSamples`. 
seed
A seed for the MCMC algorithm.

trueBeta
If available, the true value of $\beta$.

trueGroupRandomEffects
If available, the true value of $\nu$.

trueURandomEffects
If available, the true value of $u$.

trueTauSquared
If available, the true value $\tau^2$.

trueSigmaSquaredU
If available, the true value $\sigma^2_u$.

trueSigmaSquaredE
If available, the true value $\sigma^2_e$.

covarianceBetaPrior
A scalar prior $\alpha$ for the covariance parameter of the beta prior, such that the covariance is $\alpha I$.

a1
The shape parameter for the Inverse-Gamma distribution relating to the group random effects $\alpha_1$.

b1
The shape parameter for the Inverse-Gamma distribution relating to the group random effects $\xi_1$.

a2
The shape parameter for the Inverse-Gamma distribution relating to the network random effects $\alpha_2$.

b2
The scale parameter for the Inverse-Gamma distribution relating to the network random effects $\xi_2$.

a3
The shape parameter for the Inverse-Gamma distribution relating to the error terms $\alpha_3$. Only used if `family`="gaussian".

b3
The scale parameter for the Inverse-Gamma distribution relating to the error terms $\xi_3$. Only used if `family`="gaussian".

centerGroupRandomEffects
A choice to center the group random effects after each iteration of the MCMC sampler.

centerURandomEffects
A choice to center the network random effects after each iteration of the MCMC sampler.

Value

call
The matched call.

y
The response used.

X
The design matrix used.

standardizedX
The standardized design matrix used.

groupAssignment
The group assignment matrix used.

W
The network matrix used.

samples
The matrix of simulated samples from the posterior distribution of each parameter in the model (excluding random effects).
betasSamples  The matrix of simulated samples from the posterior distribution of $\beta$ parameters in the model.

tauSquaredSamples  The vector of simulated samples from the posterior distribution of $\tau^2$ in the model.

sigmaSquaredUSamples  The vector of simulated samples from the posterior distribution of $\sigma_u^2$ in the model.

sigmaSquaredESamples  The vector of simulated samples from the posterior distribution of $\sigma_e^2$ in the model.

groupRandomEffectsSamples  The matrix of simulated samples from the posterior distribution of spatial/grouping random effects $v$ in the model.

uRandomEffectsSamples  The matrix of simulated samples from the posterior distribution of network random effects $u$ in the model.

acceptanceRates  The acceptance rates of parameters in the model (excluding random effects) from the MCMC sampling scheme.

groupRandomEffectsAcceptanceRate  The acceptance rates of spatial/grouping random effects in the model from the MCMC sampling scheme.

uRandomEffectsAcceptanceRate  The acceptance rates of network random effects in the model from the MCMC sampling scheme.

timeTaken  The time taken for the model to run.

burnin  The number of MCMC samples to discard as the burn-in period.

thin  The value by which to thin numberOfSamples.

DBar  DBar for the model.

posteriorDeviance  The posterior deviance for the model.

posteriorLogLikelihood  The posterior log likelihood for the model.

pd  The number of effective parameters in the model.

DIC  The DIC for the model.

Author(s)

George Gerogiannis

Examples

```bash
#############################################################
#### Run the model on simulated data
#############################################################
```
### Load other libraries required
library(MCMCpack)

### Set up a network
observations <- 200
numberOfMultipleClassifications <- 50
W <- matrix(rbinom(observations * numberOfMultipleClassifications, 1, 0.05),
            ncol = numberOfMultipleClassifications)
numberOfActorsWithNoPeers <- sum(apply(W, 1, function(x) { sum(x) == 0 })))
peers <- sample(1:numberOfMultipleClassifications, numberOfActorsWithNoPeers, TRUE)
actorsWithNoPeers <- which(apply(W, 1, function(x) { sum(x) == 0 })))
for(i in 1:numberOfActorsWithNoPeers) {
  W[actorsWithNoPeers[i], peers[i]] <- 1
}
W <- t(apply(W, 1, function(x) { x / sum(x) })))

### Set up a single level classification
numberOfSingleClassifications <- 20
factor = sample(1:numberOfSingleClassifications, observations, TRUE)
V = matrix(NA, ncol = numberOfSingleClassifications, nrow = observations)
for(i in 1:length(factor)) {
  for(j in 1:numberOfSingleClassifications) {
    if(factor[i] == j) {
      V[i, j] = 1
    } else {
      V[i, j] = 0
    }
  }
}

### Generate the covariates and response data
X <- matrix(rnorm(2 * observations), ncol = 2)
colnames(X) <- c("x1", "x2")
beta <- c(1, -0.5, 0.5)
tauSquared <- 0.5
vRandomEffects <- rnorm(numberOfSingleClassifications, mean = 0,
                          sd = sqrt(tauSquared))
sigmaSquaredU <- 1
uRandomEffects <- rnorm(numberOfMultipleClassifications, mean = 0,
                         sd = sqrt(sigmaSquaredU))

logTheta <- cbind(rep(1, observations), X) %*% beta + V %*% vRandomEffects + W %*% uRandomEffects
Y <- rpois(n = observations, lambda = exp(logTheta))
data <- data.frame(cbind(Y, X))

### Run the model
formula <- Y ~ x1 + x2
# Not run: model <- uniNetRand(formula = formula, data = data, family="poisson",
#                           W = W, groupAssignment = V,
#                           numberOfSamples = 10000, burnin = 10000,
#                           thin = 10, seed = 1)
## End (Not run)
Index

getAdjacencyMatrix, 2
getMembershipMatrix, 4
getTotalAltersByStatus, 6

multiNet, 7
multiNetLeroux, 10
multiNetRand, 14

netcmc (netcmc-package), 2
netcmc-package, 2

plot.netcmc, 17

summary.netcmc, 18

uni, 18
uniNet, 21
uniNetLeroux, 24
uniNetRand, 30