Package 'rgm'

March 21, 2024

Type Package

Title Advanced Inference with Random Graphical Models

Version 1.0.4

Description Implements state-of-the-art Random Graphical Models (RGMs) for multivariate data analysis across multiple environments, offering tools for exploring network interactions and structural relationships. Capabilities include joint inference across environments, integration of external covariates, and a Bayesian framework for uncertainty quantification. Applicable in various fields, including microbiome analysis. Methods based on Vinciotti, V., Wit, E., & Richter, F. (2023). ``Random Graphical Model of Microbiome Interactions in Related Environments." <arXiv:2304.01956>.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 3.5.0)

Imports truncnorm, BDgraph, MASS, huge, ggplot2, stats, pROC, reshape2

LinkingTo Rcpp

RoxygenNote 7.3.0

BugReports https://github.com/franciscorichter/rgm/issues

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation yes

Author Francisco Richter [aut, cre],
Veronica Vinciotti [ctb],
Ernst Wit [ctb]

Maintainer Francisco Richter <richtf@usi.ch>

Repository CRAN

Date/Publication 2024-03-21 17:40:02 UTC

R topics documented:

`bpr` *Bayesian Probit Regression (BPR)*

Description

Performs Bayesian Probit Regression given the predictors and response.

Usage

```
bpr(y, X, offset = 0, theta, theta_0 = c(0, 0, 0), N_sim = 1)
```

Arguments

`y`	Vector of binary responses.
`X`	Matrix of predictors.
`offset`	Optional offset for the linear predictor.
`theta`	Initial values for the regression coefficients.
`theta_0`	Prior mean for the regression coefficients.
`N_sim`	Number of simulations to perform.

Value

A matrix of simulated values for the regression coefficients.

Gmcmc *Graph MCMC Sampler*

Description

Performs Markov Chain Monte Carlo (MCMC) sampling on a graph model.

Usage

```
Gmcmc(
  G,
  X = NULL,
  iter = 1000,
  alpha = NULL,
  theta = NULL,
  loc = NULL,
  burnin = 0
)
```

Arguments

G	Graph adjacency matrix.
X	Optional matrix of covariates.
iter	Number of MCMC iterations to perform.
alpha	Initial values for alpha parameters.
theta	Initial values for theta parameters.
loc	Initial locations for nodes in the graph.
burnin	Number of burn-in iterations.

Value

A list containing samples of alpha, loc, and possibly theta.

post_processing_rgm *Post-Processing for RGM (Random Graph Model)*

Description

This function performs post-processing on simulated data and results from a Random Graph Model (RGM). It calculates mean posterior estimates, compares true and estimated edge probabilities, generates various diagnostic plots, and returns a list of these plots.

Usage

```
post_processing_rgm(simulated_data, results)
```

Arguments

simulated_data	A list containing simulated data from an RGM.
results	A list containing results from fitting an RGM to 'simulated_data'.

Value

A list containing ggplot objects for different diagnostics: - 'rgm_recovery': A plot comparing true and estimated probit values. - 'estimation_of_alpha': A plot comparing true and estimated alpha values. - 'posterior_distribution': A density plot of the posterior distribution of the beta parameter. - 'beta_convergence': A trace plot of the beta parameter across MCMC iterations. - 'roc_plot': A ROC plot for graph recovery performance. - 'edge_prob': A heatmap of posterior edge probabilities for each environment.

---

rgm *Random Graphical Model*

Description

The function implements Bayesian inference of a random graphical model for multivariate data across multiple environments. The random graph prior assumes that there exists an underlying 2D latent space where the environments are located. Their vicinity in this space relates to structural similaries between the conditions. The model estimates these latent positions, the sparsity levels for each network, the regression coefficients of edge covariates associated to the propensity of two nodes ot connect (if available) and the network structures for each environment.

Usage

```
rgm(data, X=NULL, iter = 1000, burnin = 0,
initial.graphs = NULL, D = 2, initial.loc = NULL,
initial.alpha = NULL, initial.theta = NULL,
bd.iter = 20, bd.jump = 10,
method = c("ggm", "gcgm"), gcgm.dwpar = NULL)
```

Arguments

data	a list of B multivariate datasets measuring p variables across B number of environments.
X	an n.edge x ncol(X) data matrix for the edge covariates. Default is NULL, corresponding to the absence of edge covariates.
iter	number of iterations for the MCMC sampler. Default is 1000.
burnin	number of burn-in iterations to discard. Default is 0.

initial.graphs	an optional matrix of binary adjacency matrices for the initial graphs, with dimension n.edge x B. Default is NULL and in this case the initial graphs are constructed using graphical lasso (function huge).
D	number of dimensions in the latent space. Default is 2.
initial.loc	initial values for the B x D matrix of latent node positions. Default is NULL and in this case the initial values are drawn from a N(0,1) distribution.
initial.alpha	initial values for the B-dimensional intercepts. Default is NULL and in this case they are set to 0.
initial.theta	initial values for the regression coefficients associated to the covariates in X. Default is NULL and in this case they are set via a probit regression of the initial graphs on the edge covariates.
bd.iter	number of iterations for the BDgraph function. Default is 20.
bd.jump	number of links to be updated simulateneously for the BDgraph function. Default is 10.
method	method used for network estimation. Options are "ggm" (Gaussian graphical model) or "gcgm" (Gaussian copula graphical model). Default is "ggm".
gcgm.dwpar	a list of B elements, each containing the parameters of the discrete Weibull marginal fitting within each environment. This input is required only for method "gcgm" and is passed on to the function "sample.data". Default is NULL.

Details

rgm is a Bayesian random graphical model that infers the location of each environment in a 2-dimensional latent space. The probability of a link between two nodes in one environment is related to the distance of this environment to the other environments in the latent space as well as to the presence of an edge in the related environments. The model also allows for network-specific intercepts and regression coefficients for covariates measured at the edge level.

The function first initializes the latent positions, intercepts, regression coefficients and the initial graphs, if not provided. It then loops through the iterations and updates the latent positions, regression coefficients and intercepts using the Gmcmc function. Next, it calculates the probability of edge connections for each condition. Finally, it updates the network structure for each condition using the BDgraph package. The function returns the posterior samples of the parameters after discarding the burn-in period.

Value

A list containing the posterior samples of the model parameters. The list includes:

sample.alpha	a B x (iter - burnin) matrix of the alpha posterior samples of the network-specific intercepts
sample.theta	an ncol(X) x (iter - burnin) matrix of the posterior samples of the regression coefficients for the covariates in X. This is only returned if X is not NULL.
sample.loc	a B x D x (iter - burnin) array of the posterior samples of the latent positions of the conditions.
sample.graphs	an n.edge x B x (iter - burnin) array of the posterior samples of the network structures.

sample.K	an (n.edge+p) x B x (iter - burnin) array of the posterior samples of the precision matrices.
sample.pi	an n.edge x B x (iter - burnin) array of the posterior edge probabilities in each network.
pi.probit	an n.edge x B x (iter - burnin) array of the estimated probit probabilities of the edge connections in each network.

Author(s)

Veronica Vinciotti, Ernst C. Wit and Francisco Richter

Examples

```
# simulate data
sim_data <- sim.rgm(n = 10, D = 2, p = 7, B = 5)

# run inference
rgm(sim_data$data,X=sim_data$X,iter=1000)
```

---

rot *Rotate Locations*

Description

Rotates locations to align with the mean vector direction.

Usage

```
rot(loc)
```

Arguments

loc	Matrix of locations to rotate.

Value

Matrix of rotated locations.

Examples

```
# Example usage with a 2-column matrix representing locations.
loc <- matrix(rnorm(20), ncol = 2)
rotated_loc <- rot(loc)
```

sample.data *Sample Data*

Description

This function generates sample data based on the provided parameters and truncation points.

Usage

```
sample.data(data, K, tpoints)
```

Arguments

data	A list of matrices representing the data.
K	A list of matrices representing the precision matrices for each data matrix in 'data'.
tpoints	A list containing two lists of matrices for lower and upper truncation points, respectively.

Value

A list of matrices with the sampled data.

sim.rgm *Simulate Data from a Random Graphical Model*

Description

This function simulates data from a random graphical model. The graphical model is a Gaussian graphical model, with a mean zero vector and condition-specific precision matrices. The random graph model is a latent probit model, which includes condition-specific intercepts, a 2D latent space model and an edge specific covariate.

Usage

```
#sim.rgm(n = 1000, D = 2, p = 81, B = 10,
#seed = 123, mcmc_iter = 50, alpha = NULL,
#theta = NULL, loc = NULL, X = NULL)
```

Arguments

n	The number of observations for each environment. Default is 1000.
D	The dimension of the latent space. Default is 2.
p	The number of nodes in each graph. Default is 81.
B	The number of conditions. Default is 10.
seed	The random seed. Default is 123.
mcmc_iter	The number of MCMC sampling for the generation of the graphs from the joint random graph distribution. Default is 50.
alpha	The true values of the condition-specific intercepts. If NULL, these are drawn from a N(-2,1) distribution.
theta	The true values of the regression coefficients associated to the covariates in X. If NULL, this is set to 2.5.
loc	The true coordinates of the B locations in the latent space. If NULL, these are drawn from a N(0,0.3) distribution.
X	The edge specific covariates. If NULL, the data for one covariates is drawn from a Uniform(-0.5,0.5) distribution.

Value

A list with the following elements:

data	A list of B elements, where each element contains an n x p matrix of simulated Gaussian data.
X	An n.edge x ncol(X) data matrix of edge covariates.
loc	A B x D matrix of the true condition-specific coordinates.
alpha	A B-dimensional vector of the true condition-specific intercepts.
theta	A vector of the true regression coefficients associated to the covariates in X.
G	An n.edge x B matrix of the true graphs.
diagnostic	The sparsity of the graphs generated across the mcmc_iter iterations, as a diagnostic tool for convergence.

Examples

```
sim_data <- sim.rgm(n = 10, D = 2, p = 7, B = 5)
```

Index