Package ‘SAGM’

January 20, 2025

Type Package

Title Spatial Autoregressive Graphical Model

Imports fastmatrix, GIGrvg, stats, utils, mvtnorm

Version 1.0.0

Maintainer Sjoerd Hermes <sjoerd.hermes@wur.nl>

Description Implements the methodological developments found in Hermes, van Heerwaarden, and Behrouzi (2023) <doi:10.48550/arXiv.2308.04325>, and allows for the statistical modeling of asymmetric between-location effects, as well as within-location effects using spatial autoregressive graphical models. The package allows for the generation of spatial weight matrices to capture asymmetric effects for strip-type intercropping designs, although it can handle any type of spatial data commonly found in other sciences.

License GPL-3

Encoding UTF-8

LazyData true

Depends R (>= 3.10)

NeedsCompilation no

Author Sjoerd Hermes [aut, cre],
Joost van Heerwaarden [ctb],
Pariya Behrouzi [ctb]

Repository CRAN

Date/Publication 2024-02-06 11:40:02 UTC

Contents

`intercrop` *Intercropping data*

Description

This is a generated dataset containing of 4 different variables, measured across 40 plots on a strip-type intercropping design consisting of 2 crops.

Usage

```
data("intercrop")
```

Format

The format is: 40 by 4 matrix

Details

Contains generated data similar to the data used in the Hermes et al. (2023) paper, except that this data consists of a a single row of alternating crops.

Source

Generated

References

1. Hermes, S., van Heerwaarden, J., and Behrouzi, P. (2023). A Spatial Autoregressive Graphical Model with Applications in Intercropping. arXiv preprint, arXiv:2308.04325.

Examples

```
data(intercrop)
```

`make_weights` *make_weights*

Description

Create 2 weight matrices to capture asymmetric spatial effects for strip-type intercropping designs.

Usage

```
make_weights(n)
```

Arguments

`n`	Number of observations.

Value

`W_BA`	A $n \times n$ spatial weight matrix capturing the locations of type A that are adjacent to locations of type B.
`W_AB`	A $n \times n$ spatial weight matrix capturing the locations of type B that are adjacent to locations of type A.

Author(s)

Sjoerd Hermes, Joost van Heerwaarden and Pariya Behrouzi
Maintainer: Sjoerd Hermes <sjoerd.hermes@wur.nl>

References

1. Hermes, S., van Heerwaarden, J., and Behrouzi, P. (2023). A Spatial Autoregressive Graphical Model with Applications in Intercropping. arXiv preprint, arXiv:2308.04325.

Examples

```
make_weights(20)
```

`SAGM` *SAGM*

Description

This function applies the spatial autoregressive graphical model on a given dataset and array of spatial weight matrices. Different identifiability constraints can be imposed to estimate the Ψ_k. The method allows for both normal and normal-gamma priors, where the values for the hyperparameters can be specified by the user. Returns posterior samples for Θ_E and the Ψ_k.

Usage

```
SAGM(X, W, prior, constraint, triangular, idx_mat, zeta, kappa, b0, b1,
nBurnin, nIter, verbose)
```

Arguments

`X`	A $n \times p$ matrix, where n is the number of observations and p is the number of variables.
`W`	A $n \times n \times 2$ array, where the 2 generated spatial matrices are stacked. Note that the order in which the weight matrices are stacked corresponds to the order of the estimated spatial effect matrices.

prior	Prior choice on the spatial effects. Either normal ("normal") or normal-gamma ("ng").
constraint	Identifiability constraint on the spatial effects. Either symmetric ("symmetric"), triangular ("triangular") or informative ("informative").
triangular	Type of triangular restriction. Can be upper-triangular, or lower-triangular, or both, e.g. triangular = c("upper","upper"). Only has an effect whenever constraint = "triangular"
idx_mat	A nknown $\times$ $5\times$ matrix, where nknown is the number of known spatial effects. This matrix contains the indices, means and standard deviations of the known spatial effects that is specified by the user. The matrix only needs to be specified whenever constraint = "informative" is entered.
zeta	Value of hyperparameter ζ.
kappa	Value of hyperparameter κ.
b0	Value of hyperparameter b_0.
b1	Value of hyperparameter b_1.
nBurnin	Number of burnin samples.
nIter	Number of post-burnin Gibbs samples.
verbose	Return progress of parameter estimation (True) or not (False).

Value

Theta	A $p \times p \times$ nIter array consisting of the post-burnin samples for the within-plot dependencies.
Psi	A $p \times p \times 2 \times$ nIter array consisting of the post-burnin samples for the between-plot effects. The order of the third dimension of the array corresponds to that of the W.
lambda_sq	A $p \times p \times$ nIter array consisting of the post-burnin samples for Λ^2. This output is of secondary interest.
tau_sq	A vector of length nIter consisting of the post-burnin samples for τ^2. This output is of secondary interest.
accpt_rate	Value of the acceptance rate of the Metropolis Hastings step.

Author(s)

Sjoerd Hermes, Joost van Heerwaarden and Pariya Behrouzi
Maintainer: Sjoerd Hermes <sjoerd.hermes@wur.nl>

References

1. Hermes, S., van Heerwaarden, J., and Behrouzi, P. (2023). A Spatial Autoregressive Graphical Model with Applications in Intercropping. arXiv preprint, arXiv:2308.04325.

Examples

```
data(intercrop)
n <- nrow(intercrop)

W <- make_weights(n)

# Suppose we have 16 known effects. Here we assign informative normal
# priors to these effects
idx_mat <- matrix(NA, 16, 5)
idx_mat[1,] <- c(1,1,1,1, 0.1)
idx_mat[2,] <- c(1,2,1,1, 0.1)
idx_mat[3,] <- c(1,3,1,1, 0.1)
idx_mat[4,] <- c(1,1,2,1, 0.1)
idx_mat[5,] <- c(1,2,2,1, 0.1)
idx_mat[6,] <- c(1,3,2,1, 0.1)
idx_mat[7,] <- c(4,1,1,-1, 0.1)
idx_mat[8,] <- c(4,2,1,-1, 0.1)
idx_mat[9,] <- c(4,3,1,-1, 0.1)
idx_mat[10,] <- c(4,4,1,-1, 0.1)
idx_mat[11,] <- c(4,1,2,-1, 0.1)
idx_mat[12,] <- c(4,2,2,-1, 0.1)
idx_mat[13,] <- c(4,3,2,-1, 0.1)
idx_mat[14,] <- c(4,4,2,-1, 0.1)
idx_mat[15,] <- c(2,3,1,-1, 0.1)
idx_mat[16,] <- c(2,3,2,-1, 0.1)

W <- array(c(W$W_BA, W$W_AB), dim = c(n,n,2))
est <- SAGM(X = intercrop, W = W, prior = "normal", constraint = "informative",
triangular = c("upper","upper"), idx_mat = idx_mat, zeta = 0.1, kappa = 0.1,
b0 = 0.01, b1 = 0.01, nBurnin = 1000, nIter = 1000, verbose = TRUE)
```

Index