Package 'CARME'

January 9, 2023

Title CAR-MM Modelling in Stan

Version 0.1

Description 'Stan' based functions to estimate CAR-MM models. These models allow to estimate Generalised Linear Models with CAR (conditional autoregressive) spatial random effects for spatially and temporally misaligned data, provided a suitable Multiple Membership matrix. The main references are Gramatica, Liverani and Congdon (2022) <doi:10.48550/arXiv.2208.06738>, Petrof, Neyens, Nuyts, Nackaerts, Nemery and Faes (2020) <doi:10.1002/sim.8697> and Gramatica, Congdon and Liverani <doi:10.1111/rssc.12480>.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.1.2

Biarch true

Depends R (>= 3.5.0)

Imports methods, Rcpp (>= 0.12.0), rstan (>= 2.18.1), MASS, expm, stats

LinkingTo BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), rstan (>= 2.18.1), StanHeaders (>= 2.18.0)

SystemRequirements GNU make

NeedsCompilation yes

Author Marco Gramatica [aut, cre] (<https://orcid.org/0000-0002-6374-0933>), Silvia Liverani [aut] (<https://orcid.org/0000-0002-1870-3017>), Peter Congdon [aut] (<https://orcid.org/0000-0003-1934-9205>)

Maintainer Marco Gramatica <gramaticamarco@gmail.com>

Repository CRAN

Date/Publication 2023-01-09 10:33:22 UTC

R topics documented:

`CARME-package` *The 'CARME' package.*

Description

CAR-MM modelling in Stan

References

Stan Development Team (NA). RStan: the R interface to Stan. R package version 2.26.11. https://mc-stan.org

---

`car_mm` *CAR-MM prior model*

Description

CAR-MM prior model

Usage

```
car_mm(d_list, ...)
```

Arguments

`d_list`	List of data inputs for the stan model.
`...`	Arguments passed to `rstan::sampling` (e.g. iter, chains).

Value

An object of class `stanfit` returned by `rstan::sampling`

Examples

```
set.seed(455)

#---- Load data
data(W_sel)
## Number of areas
n <- nrow(W_sel)
## Number of memberships
m <- 153

#---- Simulate covariates
X <- cbind(rnorm(nrow(W_sel)), rnorm(nrow(W_sel)))
## Min-max normalisation
X_cent <- apply(X, 2, function(x) (x - min(x))/diff(range(x)))

#---- Simulate MM matrix
w_ord <- c(.5, .35, .15) # Weight of each neighbours orders
ord <- length(w_ord) - 1 # Order of neighbours to include
H_sel_sim <- sim_MM_matrix(
  W = W_sel, m = m, ord = ord, w_ord = w_ord, id_vec = rep(1, nrow(W_sel))
)

#---- Simulate outcomes
## Linear term parameters
gamma <- -.5 # Intercept
beta <- c(1, .5) # Covariates coefficients
## CAR random effects
phi_car <- sim_car(W = W_sel, alpha = .9, tau = 5)
# Areal log relative risks
l_RR <- X_cent %*% beta + phi_car
## Membership log relative risks
l_RR_mm <- as.numeric(apply(H_sel_sim, 1, function(x) x %*% l_RR))
## Expected rates
exp_rates <- rpois(m, lambda = 20)
## Outcomes
y <- rpois(m, lambda = exp_rates*exp(l_RR_mm))

#---- Create dataset for stan function
d_sel <- list(
  # Number of areas
  n = nrow(W_sel),
  # Covariates
  k = ncol(X_cent),
  X_cov = X_cent,
  # Adjacency
  W_n = sum(W_sel) / 2,
  # Number of neighbour pairs
  W = W_sel,
  # Memberships
  m = nrow(H_sel_sim),
  H = H_sel_sim,
```

```
  # Outcomes
  y = y,
  log_offset = log(exp_rates),
  # Prior parameters
  ## Intercept (mean and sd of normal prior)
  mu_gamma = 0, sigma_gamma = 1,
  ## Covariates (mean and sd of normal prior)
  mu_beta = 0, sigma_beta = 1,
  ## Marginal precision gamma prior
  tau_shape = 2,
  tau_rate = 0.2
)

#---- HMC parameters
niter <- 1E4
nchains <- 4

#---- Stan sampling
fit <- car_mm(
  d_list = d_sel,
  # arguments passed to sampling
  iter = niter, chains = nchains, refresh = 500,
  control = list(adapt_delta = .99, max_treedepth = 15)
)
```

`sim_car` *Simulation of proper CAR random effects*

Description

`sim_car` returns a vector of CAR distributed random effects

Usage

```
sim_car(W, alpha = 0.5, tau = 5)
```

Arguments

`W`	Symmetric adjacency matrix of size `n`
`alpha`	properness parameter between 0 and 1. Defaults to 0.5
`tau`	marginal precision. Defaults to 5

Value

a vector of length `n`

Examples

```
data(W_sel)
sim_car(W = W_sel, alpha = .9, tau = 5)
```

`sim_MM_matrix` *Simulation of MM matrix based*

Description

`sim_MM_matrix` returns a multiple membership matrix simulated based on an adjacency matrix according to the method described in

Usage

```
sim_MM_matrix(W, m, ord = 3, w_ord, id_vec, excess_areas = FALSE, red_areas)
```

Arguments

`W`	Symmetric adjacency matrix of size `n`
`m`	Integer. Number of membership to simulate
`ord`	Integer. Maximum order of neighbours to be used to simulate the memberships based on the adjacency matrix `W`
`w_ord`	A vector of length `ord` that specifies the weights of each order of neighbours
`id_vec`	Vector of zeros and ones of length `n`. Defaults to a vector of ones. It indicates whether an area is included in the simulation of a membership
`excess_areas`	if different from FALSE it indicates the indices of the areas to reuse in simulating memberships, whenever `m` > `n`. It defaults to FALSE, and if omitted randomly selects without replacement (if `m` - `n` <= `n`, otherwise with replacement) a subset of areas
`red_areas`	vector of indices of areas to use if `m` < `n`

Value

an m x n matrix of weights

Examples

```
set.seed(455)

#---- Load data
data(W_sel)
## Number of areas
n <- nrow(W_sel)
## Number of memberships
m <- 153
```

```
#---- Simulate MM matrix
w_ord <- c(.5, .35, .15) # Weight of each neighbours orders
ord <- length(w_ord) - 1 # Order of neighbours to include
H_sel_sim <- sim_MM_matrix(
  W = W_sel, m = m, ord = ord, w_ord = w_ord, id_vec = rep(1, nrow(W_sel))
)
```

`W_sel` *Adjacency matrix for the South East London set of MSOAs*

Description

Adjacency matrix of 152 MSOAs in South East London, used for the data analysis in the paper "Structure induced by a multiple membership transformation on the Conditional Autoregressive model". Column and rows names indicate the MSOA code.

Usage

```
data(W_sel)
```

Format

A 152x152 symmetric matrix

References

Structure induced by a multiple membership transformation on the Conditional Autoregressive model (2022) (arXiv)

Index