Package ‘BayesMRA’
August 18, 2020

Type Package
Title Bayesian Multi-Resolution Gaussian Process Approximations
Version 1.0.0
Date 2020-08-11
Description Software for fitting sparse Bayesian multi-resolution spatial models using Markov Chain Monte Carlo.
License GPL (>= 3)
Depends R (>= 3.5.0)
Imports fields, igraph, Matrix, mvnfast, Rcpp (>= 1.0.4.6), spam
RoxygenNote 7.1.0
Suggests knitr, pkgdown, rmarkdown, testthat (>= 2.1.0), covr
URL https://github.com/jtipton25/BayesMRA
BugReports https://github.com/jtipton25/BayesMRA/issues
VignetteBuilder knitr
Encoding UTF-8
LinkingTo Rcpp, RcppArmadillo
NeedsCompilation yes
Author John Tipton [aut, cre]
Maintainer John Tipton <jrtipton@uark.edu>
Repository CRAN
Date/Publication 2020-08-18 09:52:11 UTC

R topics documented:

BayesMRA ................................................................. 2
make_Q_alpha_2d ......................................................... 2
make_Q_alpha_taus ..................................................... 3
mcmc_mra ............................................................... 4
mra_wendland_2d ....................................................... 6
Description

Software for fitting sparse multi-resolution spatial models

Author(s)

John Tipton

make_Q_alpha_2d

Generate CAR precision matrix

Description

A function for setting up a conditional autoregressive (CAR) or simultaneous autoregressive (SAR) precision matrix for use as a prior in Bayesian models

Usage

make_Q_alpha_2d(n_dims, phi, use_spam = TRUE, prec_model = "CAR")

Arguments

n_dims is a vector of length M that are the dimensions of the CAR/SAR matrix at each resolution
phi is a vector of length M with each element between -1 and 1 that defines the strength of the autoregressive process. Typically this will be set to 1 for use as a prior in penalized Bayesian models
use_spam is a boolean flag to determine whether the output is a list of spam matrix objects (use_spam = TRUE) or a an n × n sparse Matrix of class "dgCMatrix" use_spam = FALSE(see Matrix package for details)
prec_model is a string that takes the values "CAR" or "SAR" and defines the graphical structure for the precision matrix.
Value

a list of $n \times n$ sparse spam matrices or Matrix matrices of class "dgCMatrix" (see Matrix package for details)

Examples

```r
n_dims <- c(4, 8)
phi <- c(0.8, 0.9)
Q_alpha <- make_Q_alpha_2d(n_dims, phi)
## plot the precision matrix structure at each resolution
layout(matrix(1:2, 1, 2))
spam::display(Q_alpha[[1]])
spam::display(Q_alpha[[2]])
```

Description

Title

Usage

`make.Q.alpha_tau2(Q_alpha, tau2, use.spam = TRUE)`

Arguments

- `Q_alpha` a list of length M composed of matrices that are the correlation structure of the CAR prior on beta.
- `tau2` a vector of length M that contains the CAR prior precision matrices.
- `use.spam` a boolean that determines if the output matrix is of class "spam" (use.spam = TRUE) or of class "dgCMatrix" (use.spam = FALSE; see Matrix package for details).

Value

A sparse block diagonal matrix representing the precision matrices for all of the resolutions of the random effects.

Examples

```r
n_dims <- c(4, 8)
phi <- c(0.8, 0.9)
tau2 <- c(3, 4)
Q_alpha <- make.Q.alpha_2d(n_dims, phi)
Q_alpha_tau2 <- make.Q.alpha_tau2(Q_alpha, tau2)
## plot the full precision matrix structure
```
Description

this function runs Markov Chain Monte Carlo to estimate the Bayesian multi-resolution spatial regression model.

Usage

```r
mcmc_mra(
  y,
  X,
  locs,
  params,
  priors = NULL,
  M = 4,
  n_neighbors = 68,
  n_coarse_grid = 10,
  n_padding = 5L,
  n_cores = 1L,
  inits = NULL,
  config = NULL,
  verbose = FALSE,
  use_spam = TRUE,
  n_chain = 1
)
```

Arguments

- `y` is a `n` vector of Gaussian data.
- `X` is a `n x p` matrix of fixed effects (like latitude, elevation, etc)
- `locs` is a `n x 2` matrix of observation locations.
- `params` is a list of parameter settings. The list `params` must contain the following values:
  - `n_adapt`: A positive integer number of adaptive MCMC iterations.
  - `n_mcmc`: A positive integer number of total MCMC iterations post adaptation.
  - `n_thin`: A positive integer number of MCMC iterations per saved sample.
  - `n_message`: A positive integer number of frequency of iterations to output a progress message. For example, `n_message = 50` outputs progress messages every 50 iterations.
- `priors` is the list of prior settings.
M
n_neighbors
n_coarse_grid
n_padding
n_cores
n_chain
n_dims
n_dims_idx
n_dims[i]
dims_idx
Q_alpha
Q_alpha_tau2
A_constraint

The number of resolutions.
The expected number of neighbors for each interior basis function. This determines the basis radius parameter.
The number of basis functions in one direction (e.g. \( n_{\text{coarse\_grid}} = 10 \) results in a \( 10 \times 10 \) course grid which is further extended by the number of additional padding basis functions given by \( n_{\text{padding}} \).
The number of additional boundary points to add on each boundary. For example, \( n_{\text{padding}} = 5 \) will add 5 boundary knots to the both the left and right side of the grid).

is the number of cores for parallel computation using openMP.
is the list of initial values if the user wishes to specify initial values. If these values are not specified, then the initial values will be randomly sampled from the prior.
is the list of configuration values if the user wishes to specify initial values. If these values are not specified, then default a configuration will be used.
Should verbose output be printed? Typically this is only useful for troubleshooting.

is a boolean flag to determine whether the output is a list of spam matrix objects (\( \text{use\_spam} = \text{TRUE} \)) or an \( n \times n \) sparse Matrix of class "dgCMatrix" (\( \text{use\_spam} = \text{FALSE} \) (see spam and Matrix packages for details).
is the MCMC chain id. The default is 1.

Examples

```r
set.seed(111)
## generate the locations
locs <- matrix(runif(20), 10, 2)
## generate some covariates and regression coefficients
X <- cbind(1, matrix(rnorm(30), 10, 3))
beta <- rnorm(ncol(X))

## simulate the MRA process
M <- 2
MRA <- mra_wendland_2d(locs, M = M, n_coarse_grid = 4)
W <- do.call(cbind, MRA$W)
n_dims <- rep(NA, length(MRA$W))
dims_idx <- NULL
for (i in 1:M) {
  n_dims[i] <- ncol(MRA$W[[i]])
dims_idx <- c(dims_idx, rep(i, n_dims[i]))
}

## set up the process precision matrices
Q_alpha <- make_Q_alpha_2d(sqrt(n_dims), c(0.9, 0.8))
Q_alpha_tau2 <- make_Q_alpha_tau2(Q_alpha, tau2 = c(2, 4))

## add in constraints so each resolution has random effects that sum to 0
A_constraint <- sapply(1:M, function(i){
  # add constraint
  # ...
})
```

```r
# Code to construct the multi-resolution sparse basis function representation for fitting spatial processes

tmp = rep(0, sum(n_dims))
tmp[dims_idx == 1] <- 1
return(tmp)

a_constraint <- rep(0, M)
alpha <- as.vector(spam::rmvnorm.prec.const(
  n = 1,
  mu = rep(0, nrow(W)),
  Q = Q_alpha_tau2,
  A = t(A_constraint),
  a = a_constraint))

## define the data
y <- as.vector(X %*% beta + W %*% alpha + rnorm(10))

## define the params for MCMC fitting
params <- list(
  n_mcmc = 5,
  n_adapt = 5,
  n_thin = 1,
  n_message = 5)

## define the model priors
priors <- list(
  alpha_tau2 = 1,
  beta_tau2 = 1,
  alpha_sigma2 = 1,
  beta_sigma2 = 1,
  mu_beta = rep(0, ncol(X)),
  Sigma_beta = 5 * diag(ncol(X)))

## Fit the MRA model using MCMC
out <- mcmc_mra(
  y = y,
  X = X,
  locs = locs,
  params = params,
  priors = priors,
  M = 2,
  n_coarse_grid = 4,
  n_cores = 1L,
  verbose = FALSE
)
```
Description

Code to construct the multi-resolution sparse basis function representation for fitting spatial processes.

Usage

mra_wendland_2d(
  locs,
  M = 4,
  n_coarse_grid = 10,
  n_padding = 5L,
  n_neighbors = 68,
  use_spam = TRUE
)

Arguments

- **locs**: The location variables in 2 dimensions over which to construct the basis function representation.
- **M**: The number of resolutions.
- **n_coarse_grid**: The number of basis functions in one direction (e.g. \( n_{\text{coarse\_grid}} = 10 \) results in a \( 10 \times 10 \) course grid which is further extended by the number of additional padding basis functions given by \( n_{\text{padding}} \)).
- **n_padding**: The number of additional boundary points to add on each boundary. For example, \( n_{\text{padding}} = 5 \) will add 5 boundary knots to the both the left and right side of the grid).
- **n_neighbors**: The expected number of neighbors for each interior basis function. This determines the basis radius parameter.
- **use_spam**: is a boolean flag to determine whether the output is a list of `spam::spam` matrix objects (\( \text{use\_spam} = \text{TRUE} \)) or an \( n \times n \) sparse Matrix of class `Matrix::dgCMatrix` \( \text{use\_spam} = \text{FALSE} \) (see `spam` and `Matrix` packages for details).

Value

A list of objects including the MRA knots locations \( \text{locs\_grid} \), the Wendland basis representation matrix \( W \) at the observed locations, the basis radius \( \text{radius} \), the numbers of resolutions \( M \), the number of expected neighbors in the interior of each grid \( n_{\text{neighbors}} \), the number of interior basis functions in one direction \( n_{\text{coarse\_grid}} \), the number of additional padding basis functions given by \( n_{\text{padding}} \), and the setting \( \text{use\_spam} \) which determines whether the MRA output uses the `spam` format.

Examples

```r
set.seed(111)
locs <- matrix(runif(20), 10, 2)
MRA <- mra_wendland_2d(locs, M = 2, n_coarse_grid = 4)
## plot the MRA grid at different resolutions
layout(matrix(1:2, 1, 2))
```
mra_wendland_2d_pred

Code to construct the mutli-resolution sparse basis function representation for fitting spatial processes

Description

Code to construct the multi-resolution sparse basis function representation for fitting spatial processes

Usage

mra_wendland_2d_pred(locs, locs_pred, MRA, use_spam = TRUE)

Arguments

locos  The location variables in 2 dimensions over which to construct the basis function representation in the fitting stage.
locs_pred  The location variables in 2 dimensions over which to construct the basis function representation in the prediction stage.
MRA  The multi-resolution basis expansion at the observed locations. This object is the output of mra_wendland-2d() and is of class "mra_wendland_2d".
use_spam  is a boolean flag to determine whether the output is a list of spam matrix objects (use_spam = TRUE) or a an \( n \times n \) sparse Matrix of class "dgCMatrix" use_spam = FALSE (see spam and Matrix packages for details).

Value

A list of objects including the MRA knots locations locs_grid, the Wendland basis representation matrix W_pred at the prediction locations, and the basis radius radius

Examples

set.seed(111)
locs <- matrix(runif(20), 10, 2)
locs_pred <- matrix(runif(20), 10, 2)
MRA <- mra_wendland_2d(locs, M = 2, n_coarse_grid = 4)
MRA_pred <- mra_wendland_2d_pred(locs, locs_pred, MRA)

## plot the MRA prediction grid at different resolutions
layout(matrix(1:2, 1, 2))
plot(MRA_pred$locs_grid[[1]])
plot(MRA_pred$locs_grid[[2]])
rmvnarma

A function for sampling from conditional multivariate normal distributions with mean $A^{-1}b$ and covariance matrix $A^{-1}$.

**Description**

A function for sampling from conditional multivariate normal distributions with mean $A^{-1}b$ and covariance matrix $A^{-1}$.

**Usage**

```r
rmvnarma(A, b)
```

**Arguments**

- `A` A $d \times d$ matrix for the Gaussian full conditional distribution precision matrix.
- `b` A $d$ vector for the Gaussian full conditional distribution mean.

**Examples**

```r
set.seed(111)
A <- diag(4)
b <- rnorm(4)
sample <- rmvnarma(A, b)
```

rmvnarma_chol

A function for sampling from conditional multivariate normal distributions with mean $A^{-1}b$ and covariance matrix $A^{-1}$.

**Description**

A function for sampling from conditional multivariate normal distributions with mean $A^{-1}b$ and covariance matrix $A^{-1}$.

**Usage**

```r
rmvnarma_chol(A_chol, b)
```

**Arguments**

- `A_chol` A $d \times d$ matrix for the Gaussian full conditional distribution precision matrix Cholesky factor.
- `b` A $d$ vector for the Gaussian full conditional distribution mean.
rmvn arma scalar | A function for sampling from conditional multivariate normal distributions with mean $A^{-1}b$ and covariance matrix $A^{-1}$.

Description

A function for sampling from conditional multivariate normal distributions with mean $A^{-1}b$ and covariance matrix $A^{-1}$.

Usage

rmvn arma scalar(a, b)

Arguments

- a | a A scalar for the Gaussian full conditional distribution precision.
- b | b A d vector for the Gaussian full conditional distribution mean.

Examples

set.seed(111)
a <- 4
b <- rnorm(1)
sample <- rmvn arma scalar(a, b)

wendland basis | calculate the Wendland basis function

Description

 calculate the Wendland basis function

Usage

wendland basis(d, radius)
Arguments

d The distance over which to calculate the Wendland basis
radius The effective radius over which the Wendland basis is defined

Value

The output of the Wendland basis applied to the distance \( d \) for a given radius \( \text{radius} \).

Examples

```r
layout(matrix(1:2, 1, 2))
curve(wendland_basis(sqrt(x^2), radius = 1), from = -2, to = 2)
curve(wendland_basis(sqrt(x^2), radius = 2), from = -2, to = 2)
```
Index

BayesMRA, 2
make_Q_alpha_2d, 2
make_Q_alpha_tau2, 3
mcmc_mra, 4
mra_wendland_2d, 6
mra_wendland_2d_pred, 8
rmvn arma, 9
rmvn arma chol, 9
rmvn arma scalar, 10
wendland basis, 10