Package 'spbridge'

November 29, 2025

Title Marginally interpretable spatial logistic regression with bridge processes

Version 1.0.0.1

Description Provides functions for marginally interpretable spatial logistic regression with bridge processes. For details, see Lee and dunson (2025+) <doi:10.48550/arXiv.2412.04744>.

License GPL (>= 3)

URL https://github.com/changwoo-lee/spbridge

BugReports https://github.com/changwoo-lee/spbridge/issues

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Imports BayesLogit,
coda,
fields,
lme4,
Matrix,
matrixStats,
methods,
spam,
stats,
utils

Contents

gambia *Gambia Malaria Data*

Description

Malaria prevalence in children recorded at villages in The Gambia, Africa. This dataset and documentation is imported from R package GeoR version 1.9-6, inherting the GPL-3 license.

Usage

```
data(gambia)
```

Format

Two objects are made available:

1. `gambia`
   A data frame with 2035 observations on the following 8 variables.
   - **x** x-coordinate of the village (UTM).
   - **y** y-coordinate of the village (UTM).
   - **pos** presence (1) or absence (0) of malaria in a blood sample taken from the child.
   - **age** age of the child, in days
   - **netuse** indicator variable denoting whether (1) or not (0) the child regularly sleeps under a bed-net.
   - **treated** indicator variable denoting whether (1) or not (0) the bed-net is treated (coded 0 if netuse=0).
   - **green** satellite-derived measure of the green-ness of vegetation in the immediate vicinity of the village (arbitrary units).
   - **phc** indicator variable denoting the presence (1) or absence (0) of a health center in the village.
2. `gambia.borders`
   A data frame with 2 variables:
   - **x** x-coordinate of the country borders.
   - **y** y-coordinate of the country borders.

References

Thomson, M., Connor, S., D Alessandro, U., Rowlingson, B., Diggle, P., Cresswell, M. & Greenwood, B. (1999). Predicting malaria infection in Gambian children from satellite data and bednet use surveys: the importance of spatial correlation in the interpretation of results. *American Journal of Tropical Medicine and Hygiene* 61: 2–8.

Diggle, P., Moyeed, R., Rowlingson, B. & Thomson, M. (2002). Childhood malaria in The Gambia: a case-study in model-based geostatistics, *Applied Statistics*.

splogi_bridge *Spatial logistic model with bridge process random effect (empirical Bayes)*

Description

$$\text{logit}\left[\Pr(y_{ij}=1 \mid X_{ij}, u(s_i))\right] = X_{ij}^{\top}\beta + u(s_i)$$

where u(s) ~ Bridge process with Matern correlation kernel, i=1,...n corresponds to n spatial locations and j=1,... N_i correspond to n_i responses at location i, resulting data of size N = N_1 + ... N_n.

Usage

```
splogi_bridge(
  y,
  X,
  id,
  coords,
 priors = list(beta_intercept_scale = 10, beta_scale = 2.5, beta_df = Inf, logpriorphi =
    NULL, rho_lb = NULL, rho_ub = NULL),
  smoothness = 1.5,
  nburn = 100,
  nsave = 1000,
  nthin = 1,
  verbose = TRUE
)
```

Arguments

y	N x 1 binary vector
X	N x p fixed-effect design matrix, including intercept
id	N x 1 vector of spatial location id. When N=n, it is point-referenced data
coords	n x 2 matrix of spatial coordinates
priors	list of prior hyperparameters, see details
smoothness	postive numeric, Matern smoothness parameter
nburn	number of burn-in interation
nsave	number of posterior samples
nthin	thin-in rate
verbose	logical, whether to print progress

Details

Priors are specified by "priors" argument, which is a list of hyperparameters. Specifically, we set zero centered normal or t prior for beta, uniform prior for Matern range parameter rho (see fields::Matern), and for phi, we use prior that induces half-Cauchy prior on the standard deviation of u.

priors is a named list with the following possible elements:

beta_intercept_scale scale of intercept parameter (default 10)

beta_scale scale of other beta parameters (default 2.5)

beta_df degrees of freedom for t prior on beta (default Inf, normal prior)

rho_lb lower bound for range parameter rho (default min distance between coords)

rho_ub upper bound for range parameter rho (default max distance between coords)

Value

Returns list of

post_save	a matrix of posterior samples (coda::mcmc) with nsave rows
u_save	a matrix of posterior samples (coda::mcmc) of random effects, with nsave rows
betam_save	a matrix of posterior samples (coda::mcmc) of population-averaged log odds, with nsave rows
loglik_save	a nsave x n matrix of pointwise log-likelihood values, can be used for WAIC calculation.
priors	list of hyperprior information
nsave	number of MCMC samples
t_mcmc	wall-clock time for running MCMC
t_premcmc	wall-clock time for preprocessing before MCMC
y	response vector
X	fixed effect design matrix
coords	a n x 2 matrix of Euclidean coordinates

list, including posterior samples of beta, phi, rho, and u saved in "post_save". Population-averaged coefficient, which is beta*phi, is saved separately as "betam_save"

Examples

```
library(spbridge)
data(gambia)
N = length(gambia$pos) # 2035
y = gambia$pos # binary response, N = 2035 by 1 vector
# define id based on spatial coords unique values
id = as.numeric(factor(paste(gambia$x, gambia$y)))
n = length(unique(id)) # 65
coords = unique(cbind(gambia$x, gambia$y)/1000) # n by 2 matrix, in km
# standardized covariates following Gelman et al (2008)
intercept = rep(1,N) # intercept
age = scale(gambia$age/365, scale = 2*sd(gambia$age/365)) # in years
netuse = gambia$netuse - mean(gambia$netuse)
treated = gambia$treated - mean(gambia$treated)
green = scale(gambia$green, scale = 2*sd(gambia$green))
green2 = scale(gambia$green^2, scale = 2*sd(gambia$green^2))
healthctr = gambia$phc - mean(gambia$phc)

X = cbind(intercept, age, netuse, treated, green, green2, healthctr)
colnames(X) = c("(Intercept)", "age", "netuse", "treated", "green", "green2", "healthctr")

centers = c(attr(age, "scaled:center"), mean(gambia$netuse), mean(gambia$treated),
          attr(green, "scaled:center"), attr(green2, "scaled:center"), mean(gambia$phc))
```

```
scales = c(attr(age, "scaled:scale"), 1, 1, attr(green, "scaled:scale"), attr(green2, "scaled:scale"), 1)
fit_bridge = spbridge::splogi_bridge(y = y,
                                     X = X,
                                     id = id,
                                     priors = list(beta_intercept_scale = 10,
                                                   beta_scale = 2.5, beta_df = Inf,
                                                   rho_lb = 0.01, rho_ub = 100),
                                     coords = coords,
                                  smoothness = 0.5, nburn = 1000, nsave = 10000, nthin = 1)
```

`splogi_bridge_fb` *Spatial logistic model with bridge process random effect (fully Bayes)*

Description

$$\text{logit}\left[\Pr(y_{ij} = 1 \mid X_{ij}, u(s_i))\right] = X_{ij}^{\top}\beta + u(s_i)$$

where u(s) ~ Bridge process with Matern correlation kernel, i=1,...n corresponds to n spatial locations and j=1,... N_i correspond to n_i responses at location i, resulting data of size N = N_1 + ... N_n.

Usage

```
splogi_bridge_fb(
  y,
  X,
  id,
  coords,
 priors = list(beta_intercept_scale = 10, beta_scale = 2.5, beta_df = Inf, logpriorphi =
    NULL, rho_lb = NULL, rho_ub = NULL),
  smoothness = 1.5,
  nburn = 100,
  nsave = 1000,
  nthin = 1,
  nparticle = 20,
  verbose = TRUE
)
```

Arguments

`y`	N x 1 binary vector
`X`	N x p fixed-effect design matrix, including intercept
`id`	N x 1 vector of spatial location id. When N=n, it is point-referenced data
`coords`	n x 2 matrix of spatial coordinates
`priors`	list of prior hyperparameters, see details
`smoothness`	postive numeric, Matern smoothness parameter
`nburn`	number of burn-in interation

nsave	number of posterior samples
nthin	thin-in rate
nparticle	number of particles in particle marginal Metropolis-Hastings
verbose	logical, whether to print progress

Details

Priors are specified by "priors" argument, which is a list of hyperparameters. Specifically, we set zero centered normal or t prior for beta, uniform prior for Matern range parameter rho (see fields::Matern), and for phi, we use prior that induces half-Cauchy prior on the standard deviation of u.

beta_intercept_scale: scale of intercept parameter (default 10) beta_scale: scale of other beta parameters (default 2.5) beta_df: degrees of freedom for t prior on beta (default Inf, normal prior) logpriorphi: function for log-prior on phi (default prior that induces half-Cauchy prior on the standard deviation of u.) rho_lb: lower bound for range parameter rho (default min distance between coords) rho_ub: upper bound for range parameter rho (default max distance between coords)

Value

Returns list of

post_save	a matrix of posterior samples (coda::mcmc) with nsave rows
u_save	a matrix of posterior samples (coda::mcmc) of random effects, with nsave rows
betam_save	a matrix of posterior samples (coda::mcmc) of population-averaged log odds, with nsave rows
loglik_save	a nsave x n matrix of pointwise log-likelihood values, can be used for WAIC calculation.
priors	list of hyperprior information
nsave	number of MCMC samples
t_mcmc	wall-clock time for running MCMC
t_premcmc	wall-clock time for preprocessing before MCMC
y	response vector
X	fixed effect design matrix
coords	a n x 2 matrix of Euclidean coordinates

list, including posterior samples of beta, phi, rho, and u saved in "post_save". Population-averaged coefficient, which is beta*phi, is saved separately as "betam_save"

Examples

```
library(spbridge)
data(gambia)
N = length(gambia$pos) # 2035
y = gambia$pos # binary response, N = 2035 by 1 vector
# define id based on spatial coords unique values
id = as.numeric(factor(paste(gambia$x, gambia$y)))
n = length(unique(id)) # 65
coords = unique(cbind(gambia$x, gambia$y)/1000) # n by 2 matrix, in km
# standardized covariates following Gelman et al (2008)
intercept = rep(1,N) # intercept
```

```
age = scale(gambia$age/365, scale = 2*sd(gambia$age/365)) # in years
netuse = gambia$netuse - mean(gambia$netuse)
treated = gambia$treated - mean(gambia$treated)
green = scale(gambia$green, scale = 2*sd(gambia$green))
green2 = scale(gambia$green^2, scale = 2*sd(gambia$green^2))
healthctr = gambia$phc - mean(gambia$phc)

X = cbind(intercept, age, netuse, treated, green, green2, healthctr)
colnames(X) = c("(Intercept)", "age", "netuse", "treated", "green", "green2", "healthctr")

centers = c(attr(age, "scaled:center"), mean(gambia$netuse), mean(gambia$treated),
          attr(green, "scaled:center"), attr(green2, "scaled:center"), mean(gambia$phc))
scales = c(attr(age, "scaled:scale"), 1, 1, attr(green, "scaled:scale"), attr(green2, "scaled:scale"), 1)
fit_bridge = spbridge::splogi_bridge_fb(y = y,
                                        X = X,
                                        id = id,
                                        priors = list(beta_intercept_scale = 10,
                                                      beta_scale = 2.5, beta_df = Inf,
                                                      rho_lb = 0.01, rho_ub = 100),
                                        coords = coords,
                                    smoothness = 0.5, nburn = 1000, nsave = 10000, nthin = 1)
```

---

splogi_bridge_fblowrank *Spatial logistic model with bridge process random effect with low-rank structure (fully Bayes)*

Description

$$\text{logit}\left[\Pr(y_{ij} = 1 \mid X_{ij}, u(s_i))\right] = X_{ij}^{\top}\beta + u(s_i)$$

where u(s) ~ Bridge process with Matern correlation kernel with low-rank structure, i=1,...n corresponds to n spatial locations and j=1,... N_i correspond to n_i responses at location i, resulting data of size N = N_1 + ... N_n.

Usage

```
splogi_bridge_fblowrank(
  y,
  X,
  id,
  coords,
  coords_knot,
 priors = list(beta_intercept_scale = 10, beta_scale = 2.5, beta_df = Inf, logpriorphi =
    NULL, rho_lb = NULL, rho_ub = NULL),
  smoothness = 1.5,
  nburn = 100,
  nsave = 1000,
  nthin = 1,
```

```
  nparticle = 20,
  verbose = TRUE
)
```

Arguments

y	N x 1 binary vector
X	N x p fixed-effect design matrix, including intercept
id	N x 1 vector of spatial location id. When N=n, it is point-referenced data
coords	n x 2 matrix of spatial coordinates
coords_knot	q x 2 matrix of knot coordinates
priors	list of prior hyperparameters, see details
smoothness	postive numeric, Matern smoothness parameter
nburn	number of burn-in interation
nsave	number of posterior samples
nthin	thin-in rate
nparticle	number of particles in particle marginal Metropolis-Hastings
verbose	logical, whether to print progress

Details

Priors are specified by "priors" argument, which is a list of hyperparameters. Specifically, we set zero centered normal or t prior for beta, uniform prior for Matern range parameter rho (see fields::Matern), and for phi, we use prior that induces half-Cauchy prior on the standard deviation of u.

priors is a named list with the following possible elements:

beta_intercept_scale scale of intercept parameter (default 10)

beta_scale scale of other beta parameters (default 2.5)

beta_df degrees of freedom for t prior on beta (default Inf, normal prior)

logpriorphi function for log-prior on phi (default prior that induces half-Cauchy prior on the standard deviation of u.)

rho_lb lower bound for range parameter rho (default min distance between coords)

rho_ub upper bound for range parameter rho (default max distance between coords)

Value

list, including posterior samples of beta, phi, rho, and u saved in "post_save". Population-averaged coefficient, which is beta*phi, is saved separately as "betam_save"

Examples

```
library(spbridge)
data(gambia)
N = length(gambia$pos) # 2035
y = gambia$pos # binary response, N = 2035 by 1 vector
# define id based on spatial coords unique values
id = as.numeric(factor(paste(gambia$x, gambia$y)))
n = length(unique(id)) # 65
```

```
coords = unique(cbind(gambia$x, gambia$y)/1000) # n by 2 matrix, in km
# standardized covariates following Gelman et al (2008)
intercept = rep(1,N) # intercept
age = scale(gambia$age/365, scale = 2*sd(gambia$age/365)) # in years
netuse = gambia$netuse - mean(gambia$netuse)
treated = gambia$treated - mean(gambia$treated)
green = scale(gambia$green, scale = 2*sd(gambia$green))
green2 = scale(gambia$green^2, scale = 2*sd(gambia$green^2))
healthctr = gambia$phc - mean(gambia$phc)

X = cbind(intercept, age, netuse, treated, green, green2, healthctr)
colnames(X) = c("(Intercept)", "age", "netuse", "treated", "green", "green2", "healthctr")

centers = c(attr(age, "scaled:center"), mean(gambia$netuse), mean(gambia$treated),
          attr(green, "scaled:center"), attr(green2, "scaled:center"), mean(gambia$phc))
scales = c(attr(age, "scaled:scale"), 1, 1, attr(green, "scaled:scale"), attr(green2, "scaled:scale"), 1)
fit_bridge = spbridge::splogi_bridge_fblowrank(y = y,
                                        X = X,
                                        id = id,
                                        priors = list(beta_intercept_scale = 10,
                                                      beta_scale = 2.5, beta_df = Inf,
                                                      rho_lb = 0.01, rho_ub = 100),
                                        coords = coords,
                                    smoothness = 0.5, nburn = 1000, nsave = 10000, nthin = 1)
```

---

splogi_bridge_lowrank *Spatial logistic model with bridge process random effect with low-rank structure (empirical Bayes)*

Description

$$\text{logit}\left[\Pr(y_{ij} = 1 \mid X_{ij}, u(s_i))\right] = X_{ij}^{\top}\beta + u(s_i)$$

where u(s) ~ Bridge process with Matern correlation kernel with low-rank structure, i=1,...n corresponds to n spatial locations and j=1,... N_i correspond to n_i responses at location i, resulting data of size N = N_1 + ... N_n.

Usage

```
splogi_bridge_lowrank(
  y,
  X,
  id,
  coords,
  coords_knot,
 priors = list(beta_intercept_scale = 10, beta_scale = 2.5, beta_df = Inf, logpriorphi =
    NULL, rho_lb = NULL, rho_ub = NULL),
  smoothness = 1.5,
  nburn = 100,
  nsave = 1000,
```

```
  nthin = 1,
  verbose = TRUE
)
```

Arguments

y	N x 1 binary vector
X	N x p fixed-effect design matrix, including intercept
id	N x 1 vector of spatial location id. When N=n, it is point-referenced data
coords	n x 2 matrix of spatial coordinates
coords_knot	q x 2 matrix of knot coordinates
priors	list of prior hyperparameters, see details
smoothness	postive numeric, Matern smoothness parameter
nburn	number of burn-in interation
nsave	number of posterior samples
nthin	thin-in rate
verbose	logical, whether to print progress

Details

Priors are specified by "priors" argument, which is a list of hyperparameters. Specifically, we set zero centered normal or t prior for beta, uniform prior for Matern range parameter rho (see fields::Matern), and for phi, we use prior that induces half-Cauchy prior on the standard deviation of u.

priors is a named list with the following possible elements:

beta_intercept_scale scale of intercept parameter (default 10)

beta_scale scale of other beta parameters (default 2.5)

beta_df degrees of freedom for t prior on beta (default Inf, normal prior)

rho_lb lower bound for range parameter rho (default min distance between coords)

rho_ub upper bound for range parameter rho (default max distance between coords)

Value

list, including posterior samples of beta, phi, rho, and u saved in "post_save". Population-averaged coefficient, which is beta*phi, is saved separately as "betam_save"

Examples

```
library(spbridge)
data(gambia)
N = length(gambia$pos) # 2035
y = gambia$pos # binary response, N = 2035 by 1 vector
# define id based on spatial coords unique values
id = as.numeric(factor(paste(gambia$x, gambia$y)))
n = length(unique(id)) # 65
coords = unique(cbind(gambia$x, gambia$y)/1000) # n by 2 matrix, in km
# standardized covariates following Gelman et al (2008)
intercept = rep(1,N) # intercept
age = scale(gambia$age/365, scale = 2*sd(gambia$age/365)) # in years
```

```
netuse = gambia$netuse - mean(gambia$netuse)
treated = gambia$treated - mean(gambia$treated)
green = scale(gambia$green, scale = 2*sd(gambia$green))
green2 = scale(gambia$green^2, scale = 2*sd(gambia$green^2))
healthctr = gambia$phc - mean(gambia$phc)

X = cbind(intercept, age, netuse, treated, green, green2, healthctr)
colnames(X) = c("(Intercept)", "age", "netuse", "treated", "green", "green2", "healthctr")

centers = c(attr(age, "scaled:center"), mean(gambia$netuse), mean(gambia$treated),
          attr(green, "scaled:center"), attr(green2, "scaled:center"), mean(gambia$phc))
scales = c(attr(age, "scaled:scale"), 1, 1, attr(green, "scaled:scale"), attr(green2, "scaled:scale"), 1)
fit_bridge = spbridge::splogi_gaussian_lowrank(y = y,
                                        X = X,
                                        id = id,
                                        priors = list(beta_intercept_scale = 10,
                                                      beta_scale = 2.5, beta_df = Inf,
                                                      rho_lb = 0.01, rho_ub = 100),
                                        coords = coords,
                                    smoothness = 0.5, nburn = 1000, nsave = 10000, nthin = 1)
```

---

splogi_gaussian *Spatial logistic model with Gaussian process random effect*

Description

$$\text{logit}\left[\Pr(y_{ij} = 1 \mid X_{ij}, u(s_i))\right] = X_{ij}^{\top}\beta + u(s_i)$$

where u(s) ~ Gaussian process with Matern correlation kernel, i=1,...n corresponds to n spatial locations and j=1,... N_i correspond to n_i responses at location i, resulting data of size N = N_1 + ... N_n.

Usage

```
splogi_gaussian(
  y,
  X,
  id,
  coords,
 priors = list(beta_intercept_scale = 10, beta_scale = 2.5, beta_df = Inf, logpriorsigu2
    = NULL, rho_lb = NULL, rho_ub = NULL),
  smoothness = 1.5,
  nburn = 100,
  nsave = 1000,
  nthin = 1,
  verbose = TRUE
)
```

Arguments

y	N x 1 binary vector
X	N x p fixed-effect design matrix, including intercept
id	N x 1 vector of spatial location id. When N=n, it is point-referenced data
coords	n x 2 matrix of spatial coordinates
priors	list of prior hyperparameters, see details
smoothness	postive numeric, Matern smoothness parameter
nburn	number of burn-in interation
nsave	number of posterior samples
nthin	thin-in rate
verbose	logical, whether to print progress

Details

Priors are specified by "priors" argument, which is a list of hyperparameters. Specifically, we set zero centered normal or t prior for beta, uniform prior for Matern range parameter rho (see fields::Matern), and for sigu2, we use prior that induces half-Cauchy prior on the standard deviation of u.

priors is a named list with the following possible elements:

beta_intercept_scale scale of intercept parameter (default 10)

beta_scale scale of other beta parameters (default 2.5)

beta_df degrees of freedom for t prior on beta (default Inf, normal prior)

logpriorsigu2 function for log-prior on sigu2 (default prior that induces half-Cauchy prior on the standard deviation of u.)

rho_lb lower bound for range parameter rho (default min distance between coords)

rho_ub upper bound for range parameter rho (default max distance between coords)

Value

list, including posterior samples of beta, sigu2, rho, and u saved in "post_save".

Examples

```
library(spbridge)
data(gambia)
N = length(gambia$pos) # 2035
y = gambia$pos # binary response, N = 2035 by 1 vector
# define id based on spatial coords unique values
id = as.numeric(factor(paste(gambia$x, gambia$y)))
n = length(unique(id)) # 65
coords = unique(cbind(gambia$x, gambia$y)/1000) # n by 2 matrix, in km
# standardized covariates following Gelman et al (2008)
intercept = rep(1,N) # intercept
age = scale(gambia$age/365, scale = 2*sd(gambia$age/365)) # in years
netuse = gambia$netuse - mean(gambia$netuse)
treated = gambia$treated - mean(gambia$treated)
green = scale(gambia$green, scale = 2*sd(gambia$green))
green2 = scale(gambia$green^2, scale = 2*sd(gambia$green^2))
healthctr = gambia$phc - mean(gambia$phc)
```

```
X = cbind(intercept, age, netuse, treated, green, green2, healthctr)
colnames(X) = c("(Intercept)", "age", "netuse", "treated", "green", "green2", "healthctr")

centers = c(attr(age, "scaled:center"), mean(gambia$netuse), mean(gambia$treated),
          attr(green, "scaled:center"), attr(green2, "scaled:center"), mean(gambia$phc))
scales = c(attr(age, "scaled:scale"), 1, 1, attr(green, "scaled:scale"), attr(green2, "scaled:scale"), 1)
fit_bridge = spbridge::splogi_gaussian(y = y,
                                       X = X,
                                       id = id,
                                       priors = list(beta_intercept_scale = 10,
                                                     beta_scale = 2.5, beta_df = Inf,
                                                     rho_lb = 0.01, rho_ub = 100),
                                       coords = coords,
                                  smoothness = 0.5, nburn = 1000, nsave = 10000, nthin = 1)
```

---

splogi_gaussian_lowrank *Spatial logistic model with Gaussian process random effect with low-rank structure*

Description

$$\text{logit}\left[\Pr(y_{ij} = 1 \mid X_{ij}, u(s_i))\right] = X_{ij}^{\top}\beta + u(s_i)$$

where u(s) ~ Gaussian process with Matern correlation kernel with low-rank structure, i=1,...n corresponds to n spatial locations and j=1,... N_i correspond to n_i responses at location i, resulting data of size N = N_1 + ... N_n.

Usage

```
splogi_gaussian_lowrank(
  y,
  X,
  id,
  coords,
  coords_knot,
 priors = list(beta_intercept_scale = 10, beta_scale = 2.5, beta_df = Inf, logpriorsigu2
    = NULL, rho_lb = NULL, rho_ub = NULL),
  smoothness = 1.5,
  nburn = 100,
  nsave = 1000,
  nthin = 1,
  verbose = TRUE
)
```

Arguments

y	N x 1 binary vector
X	N x p fixed-effect design matrix, including intercept

id	N x 1 vector of spatial location id. When N=n, it is point-referenced data
coords	n x 2 matrix of spatial coordinates
coords_knot	q x 2 matrix of knot coordinates
priors	list of prior hyperparameters, see details
smoothness	postive numeric, Matern smoothness parameter
nburn	number of burn-in interation
nsave	number of posterior samples
nthin	thin-in rate
verbose	logical, whether to print progress

Details

Priors are specified by "priors" argument, which is a list of hyperparameters. Specifically, we set zero centered normal or t prior for beta, uniform prior for Matern range parameter rho (see fields::Matern), and for sigu2, we use prior that induces half-Cauchy prior on the standard deviation of u.

priors is a named list with the following possible elements:

beta_intercept_scale scale of intercept parameter (default 10)

beta_scale scale of other beta parameters (default 2.5)

beta_df degrees of freedom for t prior on beta (default Inf, normal prior)

logpriorsigu2 function for log-prior on sigu2 (default prior that induces half-Cauchy prior on the standard deviation of u.)

rho_lb lower bound for range parameter rho (default min distance between coords)

rho_ub upper bound for range parameter rho (default max distance between coords)

Value

list, including posterior samples of beta, sigu2, rho, and u saved in "post_save".

Examples

```
library(spbridge)
data(gambia)
N = length(gambia$pos) # 2035
y = gambia$pos # binary response, N = 2035 by 1 vector
# define id based on spatial coords unique values
id = as.numeric(factor(paste(gambia$x, gambia$y)))
n = length(unique(id)) # 65
coords = unique(cbind(gambia$x, gambia$y)/1000) # n by 2 matrix, in km
# standardized covariates following Gelman et al (2008)
intercept = rep(1,N) # intercept
age = scale(gambia$age/365, scale = 2*sd(gambia$age/365)) # in years
netuse = gambia$netuse - mean(gambia$netuse)
treated = gambia$treated - mean(gambia$treated)
green = scale(gambia$green, scale = 2*sd(gambia$green))
green2 = scale(gambia$green^2, scale = 2*sd(gambia$green^2))
healthctr = gambia$phc - mean(gambia$phc)

X = cbind(intercept, age, netuse, treated, green, green2, healthctr)
colnames(X) = c("(Intercept)", "age", "netuse", "treated", "green", "green2", "healthctr")
```

```
centers = c(attr(age, "scaled:center"), mean(gambia$netuse), mean(gambia$treated),
          attr(green, "scaled:center"), attr(green2, "scaled:center"), mean(gambia$phc))
scales = c(attr(age, "scaled:scale"), 1, 1, attr(green, "scaled:scale"), attr(green2, "scaled:scale"), 1)
fit_bridge = spbridge::splogi_gaussian_lowrank(y = y,
                                        X = X,
                                        id = id,
                                        priors = list(beta_intercept_scale = 10,
                                                      beta_scale = 2.5, beta_df = Inf,
                                                      rho_lb = 0.01, rho_ub = 100),
                                        coords = coords,
                                    smoothness = 0.5, nburn = 1000, nsave = 10000, nthin = 1)
```

Index