Package 'bspme'

January 13, 2024

Type Package

Title Bayesian Spatial Measurement Error Models

Version 1.0.1

Author Changwoo Lee[aut, cre], Eun Sug Park[aut]

Maintainer Changwoo Lee <c.lee@stat.tamu.edu>

Description Scalable methods for fitting Bayesian linear and generalized linear models in the presence of spatial exposure measurement error. These models typically arise from a two-stage Bayesian analysis of environmental exposures and health outcomes. From a first-stage model, predictions of the covariate of interest ("exposure") and their uncertainty information (typically contained in MCMC samples) are used to form a multivariate normal prior distribution for exposure in a second-stage regression model. This package also provides implementation of the methods used in Lee et al. (2024) <https://arxiv.org/abs/2401.00634>.

License GPL (>= 3)

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Imports BayesLogit,
coda,
fields,
spam,
spNNGP

Depends Matrix,
R (>= 2.10)

URL https://changwoo-lee.github.io/bspme/

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

Suggests knitr,
rmarkdown

VignetteBuilder knitr

R topics documented:

bglm_me 2
blm_me 5

health_sim 8
NO2_Jan2012 8
vecchia_cov 9

Index **11**

`bglm_me` *Bayesian generalized linear models with spatial exposure measurement error.*

Description

This function fits a Bayesian generalized linear model in the presence of spatial exposure measurement error for covariate(s) X. One of the most important features of this function is that it allows a sparse matrix input for the prior precision matrix of X for scalable computation. As of version 1.0.0, only the Bayesian logistic regression model is supported among GLMs, and function `bglm_me()` runs a Gibbs sampler to carry out posterior inference using Polya-Gamma augmentation (Polson et al., 2013). See the below "Details" section below for the model description and Lee et al. (2024) for an application example in environmental epidemiology.

Usage

```
bglm_me(
  Y,
  X_mean,
  X_prec,
  Z,
  family = binomial(link = "logit"),
  nburn = 5000,
  nsave = 5000,
  nthin = 1,
  prior = NULL,
  saveX = FALSE
)
```

Arguments

`Y`	*vector<int>*, n by 1 binary response vector.
`X_mean`	*vector<num>*, n by 1 prior mean vector μ_X. When there are q multiple exposures subject to measurement error, it can be a length q list of n by 1 vectors.
`X_prec`	*matrix<num>*, n by n prior precision matrix Q_X, which allows sparse format from Matrix package. When there are q multiple exposures subject to measurement error, it can be a length q list of n by n matrices.
`Z`	*matrix<num>*, n by p matrix containing p covariates that are not subject to measurement error.
`family`	*class family*, a description of the error distribution and the link function to be used in the model. Currently, it only supports binomial(link = "logit").
`nburn`	*integer*, number of burn-in iterations (default=5000).
`nsave`	*integer*, number of posterior samples (default=5000). Total number of MCMC iteration is `nburn + nsave * nthin`.

`nthin`	*integer*, thin-in rate (default=1).
`prior`	*list*, list of prior parameters of the regression model. Default is `list(var_beta = 100)`.
`saveX`	*logical*, default FALSE, whether save posterior samples of X (exposure).

Details

Let Y_i be a binary response, X_i be a $q \times 1$ covariate vector that is subject to spatial exposure measurement error, and Z_i be a $p \times 1$ covariate vector without measurement error. Consider a logistic regression model, independently for each $i = 1, \ldots, n$,

$$\log(Pr(Y_i = 1)/Pr(Y_i = 0)) = \beta_0 + X_i^\top \beta_X + Z_i^\top \beta_Z.$$

Spatial exposure measurement error of X_i (for $i = 1, \ldots, n$) is incorporated into the model using a multivariate normal prior. For example, when $q = 1$, we have an $n-$dimensional multivariate normal prior on $X = (X_1, \ldots, X_n)^\top$,

$$(X_1, \ldots, X_n) \sim N_n(\mu_X, Q_X^{-1}).$$

Most importantly, it allows a sparse matrix input for the prior precision matrix Q_X for scalable computation, which can be obtained by Vecchia approximation. When $q > 1$, q independent $n-$dimensional multivariate normal priors are assumed.

We consider normal priors for regression coefficients,

$$\beta_0 \sim N(0, V_\beta), \quad \beta_{X,j} \overset{iid}{\sim} N(0, V_\beta), \quad \beta_{Z,k} \overset{iid}{\sim} N(0, V_\beta)$$

where `var_beta` corresponds to V_β.

Value

List of the following:

posterior `nsave` by (q+p) matrix of posterior samples of β_x and β_z as a coda::mcmc object.

time time taken for running MCMC (in seconds)

X_save (if `saveX = TRUE`) posterior samples of X

References

Polson, N. G., Scott, J. G., & Windle, J. (2013). Bayesian inference for logistic models using Pólya–Gamma latent variables. Journal of the American statistical Association, 108(504), 1339-1349.

Lee, C. J., Symanski, E., Rammah, A., Kang, D. H., Hopke, P. K., & Park, E. S. (2024). A scalable two-stage Bayesian approach accounting for exposure measurement error in environmental epidemiology. arXiv preprint arXiv:2401.00634.

Examples

```
## Not run:
library(bspme)
data(NO2_Jan2012)
data(health_sim)
library(fields)
library(maps)
# Obtain the predicted exposure mean and covariance at simulated health subject locations
```

```
# based on NO2 data obtained on Jan 10, 2012
# using a Gaussian process prior with mean zero and exponential covariance kernel
# with a fixed range 8 (in km) and standard deviation 1.

# exposure data
data_jan10 = NO2_Jan2012[NO2_Jan2012$date == as.POSIXct("2012-01-10"),]
coords_monitor = cbind(data_jan10$lon, data_jan10$lat)

# health data
coords_health = cbind(health_sim$lon, health_sim$lat)

distmat_xx <- rdist.earth(coords_monitor, miles = F)
distmat_xy <- rdist.earth(coords_monitor, coords_health, miles = F)
distmat_yy <- rdist.earth(coords_health, miles = F)

a = 8; sigma = 1; # assume known

Sigmaxx = fields::Matern(distmat_xx, smoothness = 0.5, range = a, phi = sigma^2)
Sigmaxy = fields::Matern(distmat_xy, smoothness = 0.5, range = a, phi = sigma^2)
Sigmayy = fields::Matern(distmat_yy, smoothness = 0.5, range = a, phi = sigma^2)

# posterior predictive mean and covariance of exposure at health subject locations
X_mean <- t(Sigmaxy) %*% solve(Sigmaxx, data_jan10$lnNO2)
X_cov <- Sigmayy - t(Sigmaxy) %*% solve(Sigmaxx,Sigmaxy) # n_y by n_y

# visualize
# monitoring station exposure data
quilt.plot(cbind(data_jan10$lon, data_jan10$lat),
           data_jan10$lnNO2, main = "NO2 exposures (in log) at 21 monitoring stations",
         xlab = "longitude", ylab= "latitude", xlim = c(-96.5, -94.5), ylim = c(29, 30.5))
maps::map("county", "Texas", add = T)

# posterior predictive mean of exposure at health subject locations
quilt.plot(cbind(health_sim$lon, health_sim$lat),
        X_mean, main = "posterior predictive mean of exposure at health subject locations",
         xlab = "longitude", ylab= "latitude", xlim = c(-96.5, -94.5), ylim = c(29, 30.5))
maps::map("county", "Texas", add = T)

# posterior predictive sd of exposure at health subject locations
quilt.plot(cbind(health_sim$lon, health_sim$lat),
       sqrt(diag(X_cov)), main = "posterior predictive sd of exposure at health subject locations",
         xlab = "longitude", ylab= "latitude", xlim = c(-96.5, -94.5), ylim = c(29, 30.5))
maps::map("county", "Texas", add = T)


# vecchia approximation
run_vecchia = vecchia_cov(X_cov, coords = cbind(health_sim$lon, health_sim$lat),
                          n.neighbors = 10)
Q_sparse = run_vecchia$Q
run_vecchia$cputime

# fit the model, binary response
fit = bglm_me(Y = health_sim$Ybinary,
                X_mean = X_mean,
                X_prec = Q_sparse, # sparse precision matrix
                Z = health_sim$Z,
                family = binomial(link = "logit"),
```

```
                nburn = 5000,
                nsave = 5000,
                nthin = 1)
fit$cputime
summary(fit$posterior)
library(bayesplot)
bayesplot::mcmc_trace(fit$posterior)

## End(Not run)
```

`blm_me` *Bayesian linear regression models with spatial exposure measurement error.*

Description

This function fits a Bayesian linear regression model in the presence of spatial exposure measurement error for covariate(s) X. One of the most important features of this function is that it allows a sparse matrix input for the prior precision matrix of X for scalable computation. Function `blm_me()` runs a Gibbs sampler to carry out posterior inference; see the below "Details" section below for the model description, and Lee et al. (2024) for an application example in environmental epidemiology.

Usage

```
blm_me(
  Y,
  X_mean,
  X_prec,
  Z,
  nburn = 5000,
  nsave = 5000,
  nthin = 1,
  prior = NULL,
  saveX = FALSE
)
```

Arguments

`Y`	*vector<int>*, n by 1 continuous response vector.
`X_mean`	*vector<num>*, n by 1 prior mean vector μ_X. When there are q multiple exposures subject to measurement error, it can be a length q list of n by 1 vectors.
`X_prec`	*matrix<num>*, n by n prior precision matrix Q_X, which allows sparse format from Matrix package. When there are q multiple exposures subject to measurement error, it can be a length q list of n by n matrices.
`Z`	*matrix<num>*, n by p matrix containing p covariates that are not subject to measurement error.
`nburn`	*integer*, number of burn-in iterations (default=5000).
`nsave`	*integer*, number of posterior samples (default=5000). Total number of MCMC iteration is `nburn + nsave * nthin`.

nthin	*integer*, thin-in rate (default=1).
prior	*list*, list of prior parameters of the regression model. Default is `list(var_beta = 100, a_Y = 0.01, b_Y = 0.01)`.
saveX	*logical*, default FALSE, whether save posterior samples of X (exposure).

Details

Let Y_i be a continuous response, X_i be a $q \times 1$ covariate vector that is subject to spatial exposure measurement error, and Z_i be a $p \times 1$ covariate vector without measurement error. Consider a normal linear regression model,

$$Y_i = \beta_0 + X_i^\top \beta_X + Z_i^\top \beta_Z + \epsilon_i, \quad \epsilon_i \overset{iid}{\sim} N(0, \sigma_Y^2), \quad i = 1, \ldots, n.$$

Spatial exposure measurement error of X_i for $i = 1, \ldots, n$ is incorporated into the model using a multivariate normal prior. For example when $q = 1$, we have an $n-$dimensional multivariate normal prior on $X = (X_1, \ldots, X_n)^\top$,

$$(X_1, \ldots, X_n) \sim N_n(\mu_X, Q_X^{-1}).$$

Most importantly, it allows a sparse matrix input for the prior precision matrix Q_X for scalable computation, which can be obtained by Vecchia approximation. When $q > 1$, q independent $n-$dimensional multivariate normal priors are assumed.

We consider semiconjugate priors for regression coefficients and error variance,

$$\beta_0 \sim N(0, V_\beta), \quad \beta_{X,j} \overset{iid}{\sim} N(0, V_\beta), \quad \beta_{Z,k} \overset{iid}{\sim} N(0, V_\beta), \quad \sigma_Y^2 \sim IG(a_Y, b_Y).$$

where `var_beta` corresponds to V_β, and `a_Y` and `b_Y` correspond to hyperparameters of an inverse gamma prior for σ_Y^2.

Value

list of the following:

posterior `nsave` by (q + p + 1) matrix of posterior samples of β_X, β_Z, σ_Y^2 as a coda::mcmc object.

time time taken for running MCMC (in seconds)

X_save (if `saveX = TRUE`) posterior samples of X

References

Lee, C. J., Symanski, E., Rammah, A., Kang, D. H., Hopke, P. K., & Park, E. S. (2024). A scalable two-stage Bayesian approach accounting for exposure measurement error in environmental epidemiology. arXiv preprint arXiv:2401.00634.

Examples

```
## Not run:
library(bspme)
data(NO2_Jan2012)
data(health_sim)
library(fields)
library(maps)
# Obtain the predicted exposure mean and covariance at simulated health subject locations
# based on NO2 data obtained on Jan 10, 2012
# using a Gaussian process prior with mean zero and exponential covariance kernel
```

```
# with a fixed range 8 (in km) and standard deviation 1.

# exposure data
data_jan10 = NO2_Jan2012[NO2_Jan2012$date == as.POSIXct("2012-01-10"),]
coords_monitor = cbind(data_jan10$lon, data_jan10$lat)

# health data
coords_health = cbind(health_sim$lon, health_sim$lat)

distmat_xx <- rdist.earth(coords_monitor, miles = F)
distmat_xy <- rdist.earth(coords_monitor, coords_health, miles = F)
distmat_yy <- rdist.earth(coords_health, miles = F)

a = 8; sigma = 1; # assume known

Sigmaxx = fields::Matern(distmat_xx, smoothness = 0.5, range = a, phi = sigma^2)
Sigmaxy = fields::Matern(distmat_xy, smoothness = 0.5, range = a, phi = sigma^2)
Sigmayy = fields::Matern(distmat_yy, smoothness = 0.5, range = a, phi = sigma^2)

# posterior predictive mean and covariance of exposure at health subject locations
X_mean <- t(Sigmaxy) %*% solve(Sigmaxx, data_jan10$lnNO2)
X_cov <- Sigmayy - t(Sigmaxy) %*% solve(Sigmaxx,Sigmaxy) # n_y by n_y

# visualize
# monitoring station exposure data
quilt.plot(cbind(data_jan10$lon, data_jan10$lat),
           data_jan10$lnNO2, main = "NO2 exposures (in log) at 21 monitoring stations",
         xlab = "longitude", ylab= "latitude", xlim = c(-96.5, -94.5), ylim = c(29, 30.5))
maps::map("county", "Texas", add = T)

# posterior predictive mean of exposure at health subject locations
quilt.plot(cbind(health_sim$lon, health_sim$lat),
        X_mean, main = "posterior predictive mean of exposure at health subject locations",
         xlab = "longitude", ylab= "latitude", xlim = c(-96.5, -94.5), ylim = c(29, 30.5))
maps::map("county", "Texas", add = T)

# posterior predictive sd of exposure at health subject locations
quilt.plot(cbind(health_sim$lon, health_sim$lat),
       sqrt(diag(X_cov)), main = "posterior predictive sd of exposure at health subject locations",
         xlab = "longitude", ylab= "latitude", xlim = c(-96.5, -94.5), ylim = c(29, 30.5))
maps::map("county", "Texas", add = T)


# vecchia approximation
run_vecchia = vecchia_cov(X_cov, coords = cbind(health_sim$lon, health_sim$lat),
                          n.neighbors = 10)
Q_sparse = run_vecchia$Q
run_vecchia$cputime

# fit the model, continuous response
fit = blm_me(Y = health_sim$Y,
                X_mean = X_mean,
                X_prec = Q_sparse, # sparse precision matrix
                Z = health_sim$Z,
                nburn = 5000,
                nsave = 5000,
                nthin = 1)
```

```
fit$cputime
summary(fit$posterior)
library(bayesplot)
bayesplot::mcmc_trace(fit$posterior)

## End(Not run)
```

`health_sim` *Simulated health data*

Description

Simulated health data associated with ln(NO2) concentration on Jan 10, 2012. For details, see `health_sim.R`.

Usage

```
data(health_sim)
```

Format

A data frame with n = 2000 rows and 6 variables:

Y simulated continuous health outcome

Ybinary simulated binary health outcome

lon simulated health subject longitude

lat simulated health subject latitude

Z simulated covariate (p=1) that is not subject to measurement error

X_true true ln(NO2) exposure used for simulating health outcome

---

`NO2_Jan2012` *Daily average NO2 concentrations in and around the Harris County, Texas, in Jan 2012*

Description

This dataset contains daily average NO2 (nitrogen dioxide) concentrations obtained from 21 monitoring stations in and around Harris County, Texas, in January 2012.

Usage

```
data(NO2_Jan2012)
```

Format

A data frame with 651 (21 sites x 31 days) rows and 5 variables:

date date in POSIXct format

site_name monitoring station name

lon monitoring station longitude

lat monitoring station latitude

lnNO2 natural logarithm of daily average NO2 concentrations, measured in parts per billion by volume (ppbv)

`vecchia_cov` *Run Vecchia approximation given a covariance matrix*

Description

Given a multivariate normal (MVN) distribution with covariance matrix Σ, this function finds a sparse precision matrix (inverse covariance) Q based on the Vecchia approximation (Vecchia 1988, Katzfuss and Guinness 2021), where $N(\mu, Q^{-1})$ is the sparse MVN that approximates the original MVN $N(\mu, \Sigma)$. The algorithm is based on the pseudocode 2 of Finley et al. (2019).

Usage

```
vecchia_cov(Sigma, coords, n.neighbors, ord = NULL, KLdiv = FALSE)
```

Arguments

`Sigma`	*matrix<num>*, n by n covariance matrix
`coords`	*matrix<num>*, n by 2 coordinate matrix for nearest neighborhood search
`n.neighbors`	*integer*, the number of nearest neighbors (k) to determine conditioning set of Vecchia approximation
`ord`	*vector<int>*, length n vector, ordering of data. If NULL, ordering based on the first coordinate will be used.
`KLdiv`	*logical*, If TRUE, return KL divergence $D_{KL}(p\|\|\tilde{p})$ where p is multivariate normal with original covariance matrix and $\tilde{p}$ is the approximated multivariate normal with sparse precision matrix.

Value

list of the following:

Q n by n sparse precision matrix in Matrix format

ord ordering used for Vecchia approximation

cputime time taken to run Vecchia approximation

KLdiv (if `KLdiv = TRUE`) KL divergence $D_{KL}(p||\tilde{p})$ where p is the multivariate normal with original covariance matrix and $\tilde{p}$ is the approximated multivariate normal with a sparse precision matrix.

References

Vecchia, A. V. (1988). Estimation and model identification for continuous spatial processes. Journal of the Royal Statistical Society Series B: Statistical Methodology, 50(2), 297-312.

Katzfuss, M., & Guinness, J. (2021). A General Framework for Vecchia Approximations of Gaussian Processes. Statistical Science, 36(1).

Finley, A. O., Datta, A., Cook, B. D., Morton, D. C., Andersen, H. E., & Banerjee, S. (2019). Efficient algorithms for Bayesian nearest neighbor Gaussian processes. Journal of Computational and Graphical Statistics, 28(2), 401-414.

Zhang, L., (2020), public Github repository https://github.com/LuZhangstat/NNGP_STAN.

Examples

```
n = 1000
coords = cbind(runif(n), runif(n))
Sigma = fields::Exp.cov(coords, aRange = 1)
fit5 = vecchia_cov(Sigma, coords, n.neighbors = 5, KLdiv = TRUE)
fit5$KLdiv
fit10 = vecchia_cov(Sigma, coords, n.neighbors = 10, KLdiv = TRUE)
fit10$KLdiv
```

Index

bglm_me, 2
blm_me, 5

family, *2*

health_sim, 8

Matrix, *2*, *5*, *9*
mcmc, *3*, *6*

NO2_Jan2012, 8

vecchia_cov, 9