Package: BRISC (via r-universe)

October 28, 2026

Type Package

Title Fast Inference for Large Spatial Datasets using BRISC

Version 1.0.6

Maintainer Arkajyoti Saha <arkajyotisaha93@gmail.com>

Author Arkajyoti Saha [aut, cre], Abhirup Datta [aut], Jorge Nocedal [ctb], Naoaki Okazaki [ctb], Lukas M. Weber [ctb]

Depends R (>= 3.3.0), RANN, parallel, stats, rdist, matrixStats, pbapply, graphics

Description Fits bootstrap with univariate spatial regression models using Bootstrap for Rapid Inference on Spatial Covariances (BRISC) for large datasets using nearest neighbor Gaussian processes detailed in Saha and Datta (2018) <doi:10.1002/sta4.184>.

License GPL (>= 2)

URL https://github.com/ArkajyotiSaha/BRISC

BugReports https://github.com/ArkajyotiSaha/BRISC/issues

Encoding UTF-8

NeedsCompilation yes

RoxygenNote 6.1.1

Repository https://cran.r-universe.dev

Date/Publication 2024-10-02 04:30:11 UTC

RemoteUrl https://github.com/cran/BRISC

RemoteRef HEAD

RemoteSha f9734a6cbde7e3605ccd8e11ae79916c97d7fde7

Contents

BRISC_bootstrap 2
BRISC_correlation 4

BRISC_decorrelation 6
BRISC_estimation 9
BRISC_neighbor 13
BRISC_order 14
BRISC_prediction 16
BRISC_simulation 18
BRISC_variogram.ci 20

Index **22**

`BRISC_bootstrap` *Function for performing bootstrap with BRISC*

Description

The function `BRISC_bootstrap` performs bootstrap to provide confidence intervals for parameters of univariate spatial regression models using outputs of `BRISC_estimation`. The details of the bootstrap method can be found in BRISC (Saha & Datta, 2018). The optimization is performed with C library of limited-memory BFGS libLBFGS: a library of Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS), http://www.chokkan.org/software/liblbfgs/ (Naoaki Okazaki). For user convenience the soure codes of the package libLBFGS are provided in the package. Some code blocks are borrowed from the R package: spNNGP: Spatial Regression Models for Large Datasets using Nearest Neighbor Gaussian Processes
https://CRAN.R-project.org/package=spNNGP .

Usage

```
BRISC_bootstrap(BRISC_Out, n_boot = 100, h = 1, n_omp = 1,
                init = "Initial", verbose = TRUE,
                nugget_status = 1)
```

Arguments

`BRISC_Out`	an object of class `BRISC_Out`, obtained as an output of `BRISC_estimation`.
`n_boot`	number of bootstrap samples. Default value is 100.
`h`	number of core to be used in parallel computing setup for bootstrap samples. If `h = 1`, there is no parallelization. Default value is 1.
`n_omp`	number of threads to be used, value can be more than 1 if source code is compiled with OpenMP support. Default is 1.
`init`	keyword that specifies initialization scheme to be used. Supported keywords are: `"Initial"` and `"Estimate"` for initialization of parameter values for bootstrap samples with initial values used in `BRISC_estimate` and estimated values of parameters in `BRISC_estimate` respectively.
`verbose`	if `TRUE`, model specifications along with information regarding OpenMP support and progress of the algorithm is printed to the screen. Otherwise, nothing is printed to the screen. Default value is `TRUE`.

`nugget_status` if `nugget_status = 0`, `tau.sq` is fixed to 0, if `nugget_status = 1` `tau.sq` is estimated. Default value is 1.

Value

A list comprising of the following:

`boot.Theta` estimates of spatial covariance parameters corresponding to bootstrap samples.

`boot.Beta` estimates of beta corresponding to bootstrap samples.

`confidence.interval` confidence intervals corresponding to the parameters.

`boot.time` time (in seconds) required to perform the bootstrapping after preprocessing data in R, reported using `proc.time()`.

Author(s)

Arkajyoti Saha <arkajyotisaha93@gmail.com>,
Abhirup Datta <abhidatta@jhu.edu>

References

Saha, A., & Datta, A. (2018). BRISC: bootstrap for rapid inference on spatial covariances. Stat, e184, DOI: 10.1002/sta4.184.

Okazaki N. libLBFGS: a library of Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS), http://www.chokkan.org/software/liblbfgs/ .

Andrew Finley, Abhirup Datta and Sudipto Banerjee (2017). spNNGP: Spatial Regression Models for Large Datasets using Nearest Neighbor Gaussian Processes. R package version 0.1.1. https://CRAN.R-project.org/package=spNNGP

Examples

```
rmvn <- function(n, mu = 0, V = matrix(1)){
  p <- length(mu)
  if(any(is.na(match(dim(V),p))))
    stop("Dimension not right!")
  D <- chol(V)
  t(matrix(rnorm(n*p), ncol=p)%*%D + rep(mu,rep(n,p)))
}

set.seed(1)
n <- 300
coords <- cbind(runif(n,0,1), runif(n,0,1))

beta <- c(1,5)
x <- cbind(rnorm(n), rnorm(n))

sigma.sq = 1
phi = 5
tau.sq = 0.1
```

```
B <- as.matrix(beta)
D <- as.matrix(dist(coords))
R <- exp(-phi*D)
w <- rmvn(1, rep(0,n), sigma.sq*R)

y <- rnorm(n, x%*%B + w, sqrt(tau.sq))

estimation_result <- BRISC_estimation(coords, y, x)
bootstrap_result <- BRISC_bootstrap(estimation_result, n_boot = 10)
```

`BRISC_correlation` *Function for simulating correlated data with BRISC*

Description

The function `BRISC_correlation` creates correlated data (known structure) using Nearest Neighbor Gaussian Processes (NNGP). `BRISC_correlation` uses the sparse Cholesky representation of Vecchia's likelihood developed in Datta et al., 2016. Some code blocks are borrowed from the R package: spNNGP: Spatial Regression Models for Large Datasets using Nearest Neighbor Gaussian Processes https://CRAN.R-project.org/package=spNNGP .

Usage

```
BRISC_correlation(coords, sim, sigma.sq = 1, tau.sq = 0, phi = 1,
                  nu = 1.5, n.neighbors = NULL, n_omp = 1,
                  cov.model = "exponential",
                  search.type = "tree", stabilization = NULL,
                  verbose = TRUE, tol = 12)
```

Arguments

`coords`	an $n \times 2$ matrix of the observation coordinates in R^2 (e.g., easting and northing).
`sim`	an $n \times k$ matrix of the k many $n \times 1$ vectors from which the correlated data are calculated (see Details below).
`sigma.sq`	value of sigma square. Default value is 1.
`tau.sq`	value of tau square. Default value is 0.1.
`phi`	value of phi. Default value is 1.
`nu`	value of nu, only required for matern covariance model. Default value is 1.5.
`n.neighbors`	number of neighbors used in the NNGP. Default value is $max(100, n-1)$. We suggest a high value of `n.neighbors` for lower value of phi.
`n_omp`	number of threads to be used, value can be more than 1 if source code is compiled with OpenMP support. Default is 1.

cov.model	keyword that specifies the covariance function to be used in modelling the spatial dependence structure among the observations. Supported keywords are: "exponential", "matern", "spherical", and "gaussian" for exponential, Matern, spherical and Gaussian covariance function respectively. Default value is "exponential".
search.type	keyword that specifies type of nearest neighbor search algorithm to be used. Supported keywords are: "brute", "tree" and "cb". "brute" and "tree" provide the same result, though "tree" should be faster. "cb" implements fast code book search described in Ra and Kim (1993) modified for NNGP. If locations do not have identical coordinate values on the axis used for the nearest neighbor determination, then "cb" and "brute" should produce identical neighbor sets. However, if there are identical coordinate values on the axis used for nearest neighbor determination, then "cb" and "brute" might produce different, but equally valid neighbor sets, e.g., if data are on a grid. Default value is "tree".
stabilization	when we use a very smooth covarince model (lower values of phi for spherical and Gaussian covariance and low phi and high nu for Matern covarinace) in absence of a non-negligble nugget, the correlation process may fail due to computational instability. If stabilization = TRUE, performs stabilization by setting tau.sq = maxtau .sq, sigma.sq $* 1e-06$. Default value is TRUE for cov.model = "expoenential" and FALSE otherwise.
verbose	if TRUE, model specifications along with information regarding OpenMP support and progress of the algorithm is printed to the screen. Otherwise, nothing is printed to the screen. Default value is TRUE.
tol	the input observation coordinates are rounded to this many places after the decimal. The default value is 12.

Details

Denote g be the input sim. Let Σ be the precision matrix associated with the covariance model determined by the $cov.model$ and model parameters. Then BRISC_correlation calculates h, where h is given as follows:

$$S^{-0.5}h = g$$

where, $S^{-0.5}$ is a sparse approximation of the cholesky factor $\Sigma^{-0.5}$ of the precision matrix Σ^{-1}, obtained from NNGP.

Value

A list comprising of the following:

coords	the matrix coords.
n.neighbors	the used value of n.neighbors.
cov.model	the used covariance model.
Theta	parameters of covarinace model; accounts for stabilization.
input.data	the matrix sim.
output.data	the output matrix h in Details.

time	time (in seconds) required after preprocessing data in R, reported using, proc.time().

Author(s)

Arkajyoti Saha <arkajyotisaha93@gmail.com>,
Abhirup Datta <abhidatta@jhu.edu>

References

Datta, A., S. Banerjee, A.O. Finley, and A.E. Gelfand. (2016) Hierarchical Nearest-Neighbor Gaussian process models for large geostatistical datasets. Journal of the American Statistical Association, 111:800-812.

Andrew Finley, Abhirup Datta and Sudipto Banerjee (2017). spNNGP: Spatial Regression Models for Large Datasets using Nearest Neighbor Gaussian Processes. R package version 0.1.1. https://CRAN.R-project.org/package=spNNGP

Examples

```
set.seed(1)
n <- 1000
coords <- cbind(runif(n,0,1), runif(n,0,1))

sigma.sq = 1
phi = 1

set.seed(1)
sim <- matrix(rnorm(3*n),n, 3)
correlation_result <- BRISC_correlation(coords, sigma.sq = sigma.sq,
                                        phi = phi, sim = sim)
```

BRISC_decorrelation *Function for decorrelating data with BRISC*

Description

The function BRISC_decorrelation is used to decorrelate data (known structure) using Nearest Neighbor Gaussian Processes (NNGP). BRISC_decorrelation uses the sparse Cholesky representation of Vecchia's likelihood developed in Datta et al., 2016. Some code blocks are borrowed from the R package: spNNGP: Spatial Regression Models for Large Datasets using Nearest Neighbor Gaussian Processes https://CRAN.R-project.org/package=spNNGP .

Usage

```
BRISC_decorrelation(coords, sim, sigma.sq = 1, tau.sq = 0,
                    phi = 1, nu = 1.5, n.neighbors = NULL,
                    n_omp = 1, cov.model = "exponential",
                    search.type = "tree",
                    stabilization = NULL, verbose = TRUE,
                    tol = 12)
```

Arguments

coords	an $n \times 2$ matrix of the observation coordinates in R^2 (e.g., easting and northing).
sim	an $n \times k$ matrix of the k many $n \times 1$ vectors from which the decorrelated data are calculated (see Details below).
sigma.sq	value of sigma square. Default value is 1.
tau.sq	value of tau square. Default value is 0.1.
phi	value of phi. Default value is 1.
nu	value of nu, only required for Matern covariance model. Default value is 1.5.
n.neighbors	number of neighbors used in the NNGP. Default value is $max(100, n-1)$. We suggest a high value of n.neighbors for lower value of phi.
n_omp	number of threads to be used, value can be more than 1 if source code is compiled with OpenMP support. Default is 1.
cov.model	keyword that specifies the covariance function to be used in modelling the spatial dependence structure among the observations. Supported keywords are: "exponential", "matern", "spherical", and "gaussian" for exponential, Matern, spherical and Gaussian covariance function respectively. Default value is "exponential".
search.type	keyword that specifies type of nearest neighbor search algorithm to be used. Supported keywords are: "brute", "tree" and "cb". "brute" and "tree" provide the same result, though "tree" should be faster. "cb" implements fast code book search described in Ra and Kim (1993) modified for NNGP. If locations do not have identical coordinate values on the axis used for the nearest neighbor determination, then "cb" and "brute" should produce identical neighbor sets. However, if there are identical coordinate values on the axis used for nearest neighbor determination, then "cb" and "brute" might produce different, but equally valid neighbor sets, e.g., if data are on a grid. Default value is "tree".
stabilization	when the correlated data are generated from a very smooth covarince model (lower values of phi for spherical and Gaussian covariance and low phi and high nu for Matern covarinace), the decorrelation process may fail due to computational instability. If stabilization = TRUE, performs stabilization by adding a white noise to the data with nugget tau.sq = sigma.sq * 1e-06. Default value is TRUE for cov.model = "expoenential" and FALSE otherwise.
verbose	if TRUE, model specifications along with information regarding OpenMP support and progress of the algorithm is printed to the screen. Otherwise, nothing is printed to the screen. Default value is TRUE.
tol	the input observation coordinates are rounded to this many places after the decimal. The default value is 12.

Details

Denote h be the input sim. Let Σ be the covariance matrix associated with the covariance model determined by the $cov.model$ and model parameters. Then BRISC_decorrelation calculates g, where g is given as follows:

$$S^{-0.5}h = g$$

where, $S^{-0.5}$ is a sparse approximation of the cholesky factor $\Sigma^{-0.5}$ of the precision matrix Σ^{-1}, obtained from NNGP.

Value

A list comprising of the following:

`coords`	the matrix `coords`.
`n.neighbors`	the used value of `n.neighbors`.
`cov.model`	the used covariance model.
`Theta`	parameters of covarinace model; accounts for `stabilization`.
`input.data`	if `stabilization = FALSE`, return the matrix `sim`. If `stabilization = TRUE`, returns `sim` + used white noise in stabilization process.
`output.data`	the output matrix g in Details.
`time`	time (in seconds) required after preprocessing data in `R`, reported using, `proc.time()`.

Author(s)

Arkajyoti Saha <arkajyotisaha93@gmail.com>,
Abhirup Datta <abhidatta@jhu.edu>

References

Datta, A., S. Banerjee, A.O. Finley, and A.E. Gelfand. (2016) Hierarchical Nearest-Neighbor Gaussian process models for large geostatistical datasets. Journal of the American Statistical Association, 111:800-812.

Andrew Finley, Abhirup Datta and Sudipto Banerjee (2017). spNNGP: Spatial Regression Models for Large Datasets using Nearest Neighbor Gaussian Processes. R package version 0.1.1. https://CRAN.R-project.org/package=spNNGP

Examples

```
rmvn <- function(n, mu = 0, V = matrix(1)){
  p <- length(mu)
  if(any(is.na(match(dim(V),p))))
    stop("Dimension not right!")
  D <- chol(V)
  t(matrix(rnorm(n*p), ncol=p)%*%D + rep(mu,rep(n,p)))
}

set.seed(1)
n <- 1000
coords <- cbind(runif(n,0,1), runif(n,0,1))

sigma.sq = 1
phi = 1

set.seed(1)
```

```
D <- as.matrix(dist(coords))
R <- exp(-phi*D)
sim <- rmvn(3, rep(0,n), sigma.sq*R)
decorrelation_result <- BRISC_decorrelation(coords, sim = sim)
```

`BRISC_estimation` *Function for estimation with BRISC*

Description

The function `BRISC_estimation` fits univariate spatial regression models for large spatial data using Vecchia's approximate likelihood (Vecchia, 1988). `BRISC_estimation` uses the sparse Cholesky representation of Vecchia's likelihood developed in Datta et al., 2016. The Maximum Likelihood Estimates (MLE) of the parameters are used later for calculating the confidence interval via the `BRISC_bootstrap` (BRISC, Saha & Datta, 2018).
We recommend using `BRISC_estimation` followed by `BRISC_bootstrap` to obtain the confidence intervals for the model parameters.

The optimization is performed with C library of limited-memory BFGS libLBFGS: a library of Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS), http://www.chokkan.org/software/liblbfgs/ (Naoaki Okazaki). For user convenience the source codes of the package libLBFGS are provided in the package. The code for the coordinate ordering method, approximate Maximum Minimum Distance (Guinness, 2018) is available in https://github.com/joeguinness/gp_reorder/tree/master/R and is adopted with minor modification. Some code blocks are borrowed from the R package: spNNGP: Spatial Regression Models for Large Datasets using Nearest Neighbor Gaussian Processes https://CRAN.R-project.org/package=spNNGP .

Usage

```
BRISC_estimation(coords, y, x = NULL, sigma.sq = 1,
                 tau.sq = 0.1, phi = 1,
                 nu = 1.5, n.neighbors = 15,
                 n_omp = 1, order = "Sum_coords",
                 cov.model = "exponential",
                 search.type = "tree",
                 stabilization = NULL,
                 pred.stabilization = 1e-8,
                 verbose = TRUE, eps = 2e-05,
                 nugget_status = 1, ordering = NULL,
                 neighbor = NULL, tol = 12)
```

Arguments

`coords`	an $n \times 2$ matrix of the observation coordinates in R^2 (e.g., easting and northing).
`y`	an n length vector of response at the observed coordinates.
`x`	an $n \times p$ matrix of the covariates in the observation coordinates. Default value is $n \times 1$ matrix of 1 to adjust for the mean(intercept).

`sigma.sq` starting value of sigma square. Default value is 1.

`tau.sq` starting value of tau square. Default value is 0.1.

`phi` starting value of phi. Default value is 1.

`nu` starting value of nu, only required for matern covariance model. Default value is 1.5.

`n.neighbors` number of neighbors used in the NNGP. Default value is 15.

`n_omp` number of threads to be used, value can be more than 1 if source code is compiled with OpenMP support. Default is 1.

`order` keyword that specifies the ordering scheme to be used in ordering the observations. Supported keywords are: `"AMMD"` and `"Sum_coords"` for approximate Maximum Minimum Distance and sum of coordinate based ordering, respectively. Default value is `"Sum_coords"`. $n > 65$ is required for `"AMMD"`. Ignored, if `"ordering"` in not `NULL`.

`cov.model` keyword that specifies the covariance function to be used in modelling the spatial dependence structure among the observations. Supported keywords are: `"exponential"`, `"matern"`, `"spherical"`, and `"gaussian"` for exponential, Matern, spherical and Gaussian covariance function respectively. Default value is "exponential".

`search.type` keyword that specifies type of nearest neighbor search algorithm to be used. Supported keywords are: `"brute"`, `"tree"` and `"cb"`.
`"brute"` and `"tree"` provide the same result, though `"tree"` should be faster. `"cb"` implements fast code book search described in Ra and Kim (1993) modified for NNGP. If locations do not have identical coordinate values on the axis used for the nearest neighbor ordering (see `order` argument) then `"cb"` and `"brute"` should produce identical neighbor sets. However, if there are identical coordinate values on the axis used for nearest neighbor ordering, then `"cb"` and `"brute"` might produce different, but equally valid, neighbor sets, e.g., if data are on a grid. Default value is `"tree"`. Ignored, if `"neighbor"` in not `NULL`.

`stabilization` when the spatial errors are generated from a very smooth covarince model (lower values of phi for spherical and Gaussian covariance and low phi and high nu for Matern covarinace), the estimation process may fail due to computational instability. If `stabilization = TRUE`, performs stabilization by adding a white noise to the reordered data with nugget `tau.sq = sigma.sq` * 1e-06. Estimation is performed on this new data with `nugget_status = 1` (see `nugget_status` argument below). Default value is `TRUE` for `cov.model = "expoenential"` and `FALSE` otherwise.

`pred.stabilization` if not `NULL`, will truncate the estimated tau square to `pred.stabilization` * estimated sigma square. This provides additional stability in `BRISC_prediction`. Default value is $1e-8$.

`verbose` if `TRUE`, model specifications along with information regarding OpenMP support and progress of the algorithm is printed to the screen. Otherwise, nothing is printed to the screen. Default value is `TRUE`.

`eps` tolerance to be used in centred finite difference approximation of derivatives. Default value is 2e-05.

nugget_status	if nugget_status = 0, tau.sq is fixed to 0, if nugget_status = 1 tau.sq is estimated. Default value is 1.
ordering	if NULL, the observed locations will be ordered following the scheme in "order". If not NULL, the ordering step is bypassed and the input must denote the n length integer vector of ordering of the input coordinates that is to be used as the ordering of the coordinates for determination of the set of nearest neighbors. Output from BRISC_order can be used here.
neighbor	if NULL, neighbor set and corresponding information are created using the search type specified in "search.type". If not NULL, the step of searching the neighbors is bypassed and the input must be an output from BRISC_neighbor with identical input in "order", "ordering" and "search.type".
tol	the input observation coordinates, response and the covariates are rounded to this many places after the decimal. The default value is 12.

Value

An object of class BRISC_Out, which is a list comprising:

ord	the vector of indices used to order data necessary for fitting the NNGP model.
coords	the matrix coords[ord,].
y	If stabilization = FALSE, returns the vector y[ord]. If stabilization = TRUE, returns y[ord] + used white noise in stabilization process.
X	the matrix x[ord,,drop=FALSE].
n.neighbors	the used value of n.neighbors.
cov.model	the used covariance model.
eps	value of used eps for approximate derivation. Default value is 2e-05.
init	initial values of the parameters of the covariance model; accounts for stabilization.
Beta	estimate of beta.
Theta	estimate of parameters of covarinace model.
log_likelihood	value of Vecchia's approximate log likelihood with parameter estimates.
estimation.time	time (in seconds) required to perform the model fitting after ordering and preprocessing data in R, reported using proc.time().
BRISC_Object	object required for bootstrap and prediction.

Author(s)

Arkajyoti Saha <arkajyotisaha93@gmail.com>,
Abhirup Datta <abhidatta@jhu.edu>

References

Saha, A., & Datta, A. (2018). BRISC: bootstrap for rapid inference on spatial covariances. Stat, e184, DOI: 10.1002/sta4.184.

Datta, A., S. Banerjee, A.O. Finley, and A.E. Gelfand. (2016) Hierarchical Nearest-Neighbor Gaussian process models for large geostatistical datasets. Journal of the American Statistical Association, 111:800-812.

Guinness, J. (2018) Permutation and Grouping Methods for Sharpening Gaussian Process Approximations, Technometrics, DOI: 10.1080/00401706.2018.1437476, https://github.com/joeguinness/gp_reorder/tree/master/R .

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

Okazaki N. libLBFGS: a library of Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS), http://www.chokkan.org/software/liblbfgs/ .

Andrew Finley, Abhirup Datta and Sudipto Banerjee (2020). spNNGP: Spatial Regression Models for Large Datasets using Nearest Neighbor Gaussian Processes. R package version 0.1.4. https://CRAN.R-project.org/package=spNNGP

Ra, S. W., & Kim, J. K. (1993). A fast mean-distance-ordered partial codebook search algorithm for image vector quantization. IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, 40(9), 576-579.

Examples

```
rmvn <- function(n, mu = 0, V = matrix(1)){
  p <- length(mu)
  if(any(is.na(match(dim(V),p))))
    stop("Dimension not right!")
  D <- chol(V)
  t(matrix(rnorm(n*p), ncol=p)%*%D + rep(mu,rep(n,p)))
}

set.seed(1)
n <- 1000
coords <- cbind(runif(n,0,1), runif(n,0,1))

beta <- c(1,5)
x <- cbind(rnorm(n), rnorm(n))

sigma.sq = 1
phi = 1
tau.sq = 0.1

B <- as.matrix(beta)
D <- as.matrix(dist(coords))
R <- exp(-phi*D)
w <- rmvn(1, rep(0,n), sigma.sq*R)

y <- rnorm(n, x%*%B + w, sqrt(tau.sq))
```

```
estimation_result <- BRISC_estimation(coords, y, x)
estimation_result$Theta ##Estimates of covariance model parameters.
estimation_result$Beta ##Estimates of Beta
```

BRISC_neighbor *Function for finding set of nearest neighbors for BRISC*

Description

The function BRISC_neighbor creates the set of nearest neighbors for a given set of coordinates, which can be used as an input for "neighbor" argument in BRISC_estimation. This is especially useful for avoiding often computationally intensive nearest neighbor finding scheme in case of multiple application of BRISC_estimation on a fixed set of coordinates.

Usage

```
BRISC_neighbor(coords, n.neighbors = 15, n_omp = 1,
               order = "Sum_coords", search.type = "tree",
               verbose = TRUE, ordering = NULL, tol = 12
)
```

Arguments

coords	an $n \times 2$ matrix of the observation coordinates in R^2 (e.g., easting and northing).
n.neighbors	number of neighbors used in the NNGP. Default value is 15.
n_omp	number of threads to be used, value can be more than 1 if source code is compiled with OpenMP support. Default is 1.
order	keyword that specifies the ordering scheme to be used in ordering the observations. Supported keywords are: "AMMD" and "Sum_coords" for approximate Maximum Minimum Distance and sum of coordinate based ordering, respectively. Default value is "Sum_coords". $n > 65$ is required for "AMMD". Ignored, if ordering in not NULL.
search.type	keyword that specifies type of nearest neighbor search algorithm to be used. Supported keywords are: "brute", "tree" and "cb". "brute" and "tree" provide the same result, though "tree" should be faster. "cb" implements fast code book search described in Ra and Kim (1993) modified for NNGP. If locations do not have identical coordinate values on the axis used for the nearest neighbor ordering (see order argument) then "cb" and "brute" should produce identical neighbor sets. However, if there are identical coordinate values on the axis used for nearest neighbor ordering, then "cb" and "brute" might produce different, but equally valid, neighbor sets, e.g., if data are on a grid. Default value is "tree".
verbose	if TRUE, information regarding OpenMP support and progress of the algorithm is printed to the screen. Otherwise, nothing is printed to the screen. Default value is TRUE.

ordering	if not NULL, denotes the n length integer vector of ordering of the input coordinates and is used as the ordering of the coordinates for determination of the set of nearest neighbors.
tol	the input observation coordinates, response and the covariates are rounded to this many places after the decimal. The default value is 12.

Value

A list containing information regarding nearest neighbors which can be used as an input for "neighbor" argument in BRISC_estimation.

Author(s)

Arkajyoti Saha <arkajyotisaha93@gmail.com>,
Abhirup Datta <abhidatta@jhu.edu>

References

Saha, A., & Datta, A. (2018). BRISC: bootstrap for rapid inference on spatial covariances. Stat, e184, DOI: 10.1002/sta4.184.

Datta, A., S. Banerjee, A.O. Finley, and A.E. Gelfand. (2016) Hierarchical Nearest-Neighbor Gaussian process models for large geostatistical datasets. Journal of the American Statistical Association, 111:800-812.

Guinness, J. (2018) Permutation and Grouping Methods for Sharpening Gaussian Process Approximations, Technometrics, DOI: 10.1080/00401706.2018.1437476,
https://github.com/joeguinness/gp_reorder/tree/master/R .

Examples

```
set.seed(1)
n <- 1000
coords <- cbind(runif(n,0,1), runif(n,0,1))


neighbor_result <- BRISC_neighbor(coords)
```

BRISC_order *Function for ordering coordinates with BRISC*

Description

The function BRISC_order outputs the ordering for a set of coordinates, which can be used as an input for "ordering" argument in BRISC_estimation. This is especially useful for avoiding often computationally intensive location ordering scheme in case of multiple application of BRISC_estimation on a fixed set of coordinates.

Usage

```
BRISC_order(coords, order = "Sum_coords", verbose = TRUE)
```

Arguments

coords	an $n \times 2$ matrix of the observation coordinates in R^2 (e.g., easting and northing).
order	keyword that specifies the ordering scheme to be used in ordering the observations. Supported keywords are: "AMMD" and "Sum_coords" for approximate Maximum Minimum Distance and sum of coordinate based ordering, respectively. Default value is "Sum_coords". $n > 65$ is required for "AMMD".
verbose	if TRUE, progress of the algorithm is printed to the screen. Otherwise, nothing is printed to the screen. Default value is TRUE.

Value

An integer vector of ordering of the input coordinates which can be used as an input for "ordering" argument in BRISC_estimation.

Author(s)

Arkajyoti Saha <arkajyotisaha93@gmail.com>,
Abhirup Datta <abhidatta@jhu.edu>

References

Saha, A., & Datta, A. (2018). BRISC: bootstrap for rapid inference on spatial covariances. Stat, e184, DOI: 10.1002/sta4.184.

Datta, A., S. Banerjee, A.O. Finley, and A.E. Gelfand. (2016) Hierarchical Nearest-Neighbor Gaussian process models for large geostatistical datasets. Journal of the American Statistical Association, 111:800-812.

Guinness, J. (2018) Permutation and Grouping Methods for Sharpening Gaussian Process Approximations, Technometrics, DOI: 10.1080/00401706.2018.1437476,
https://github.com/joeguinness/gp_reorder/tree/master/R .

Examples

```
set.seed(1)
n <- 1000
coords <- cbind(runif(n,0,1), runif(n,0,1))


ordering_result <- BRISC_order(coords)
```

BRISC_prediction *Function for performing prediction with BRISC*

Description

The function BRISC_prediction performs fast prediction on a set of new locations with univariate spatial regression models using Nearest Neighbor Gaussian Processes (NNGP) (Datta et al., 2016). BRISC_prediction uses the parameter estimates from BRISC_estimation for the prediction. Some code blocks are borrowed from the R package: spNNGP: Spatial Regression Models for Large Datasets using Nearest Neighbor Gaussian Processes
https://CRAN.R-project.org/package=spNNGP .

Usage

```
BRISC_prediction(BRISC_Out, coords.0, X.0 = NULL, n_omp = 1,
                 verbose = TRUE, tol = 12)
```

Arguments

BRISC_Out	an object of class BRISC_Out, obtained as an output of BRISC_estimation.
coords.0	the spatial coordinates corresponding to prediction locations. Its structure should be same as that of coords in BRISC_estimation. Default value is a column of 1 to adjust for the mean (intercept).
X.0	the covariates for prediction locations. Its Structure should be identical (including intercept) with that of covariates provided for estimation purpose in BRISC_estimation.
n_omp	number of threads to be used, value can be more than 1 if source code is compiled with OpenMP support. Default is 1.
verbose	if TRUE, model specifications along with information regarding OpenMP support and progress of the algorithm is printed to the screen. Otherwise, nothing is printed to the screen. Default value is TRUE.
tol	the coordinates and the covariates corresponding to the prediction locations are rounded to this many places after the decimal. The default value is 12.

Value

A list comprising of the following:

prediction	predicted response corresponding to X.0 and coords.0.
prediction.ci	confidence intervals corresponding to the predictions.
prediction.time	time (in seconds) required to perform the prediction after preprocessing data in R, reported using proc.time().

Author(s)

Arkajyoti Saha <arkajyotisaha93@gmail.com>,
Abhirup Datta <abhidatta@jhu.edu>

References

Datta, A., S. Banerjee, A.O. Finley, and A.E. Gelfand. (2016) Hierarchical Nearest-Neighbor Gaussian process models for large geostatistical datasets. Journal of the American Statistical Association, 111:800-812.

Andrew Finley, Abhirup Datta and Sudipto Banerjee (2017). spNNGP: Spatial Regression Models for Large Datasets using Nearest Neighbor Gaussian Processes. R package version 0.1.1. https://CRAN.R-project.org/package=spNNGP

Examples

```
rmvn <- function(n, mu = 0, V = matrix(1)){
  p <- length(mu)
  if(any(is.na(match(dim(V),p))))
    stop("Dimension not right!")
  D <- chol(V)
  t(matrix(rnorm(n*p), ncol=p)%*%D + rep(mu,rep(n,p)))
}

set.seed(1)
n <- 500
coords <- cbind(runif(n,0,1), runif(n,0,1))

beta <- c(1,5)
x <- cbind(rnorm(n), rnorm(n))

sigma.sq = 1
phi = 1
tau.sq = 0.1

B <- as.matrix(beta)
D <- as.matrix(dist(coords))
R <- exp(-phi*D)
w <- rmvn(1, rep(0,n), sigma.sq*R)

y <- rnorm(n, x%*%B + w, sqrt(tau.sq))

estimation_result <- BRISC_estimation(coords[1:400,], y[1:400], x[1:400,])
prediction_result <- BRISC_prediction(estimation_result,
                                      coords[401:500,], x[401:500,])
```

BRISC_simulation *Function to simulate data with BRISC*

Description

The function BRISC_simulation simulates correlated data (known structure) using Nearest Neighbor Gaussian Processes (NNGP). BRISC_simulation uses the sparse Cholesky representation of Vecchia's likelihood developed in Datta et al., 2016. BRISC_simulation uses BRISC_correlation for this purpose.

Usage

```
BRISC_simulation(coords, sim_number = 1,
                 seeds =  NULL, sigma.sq = 1,
                 tau.sq = 0, phi = 1, nu = 1.5,
                 n.neighbors = NULL, n_omp = 1,
                 cov.model = "exponential",
                 search.type = "tree",
                 stabilization = NULL,
                 verbose = TRUE, tol = 12)
```

Arguments

coords	an $n \times 2$ matrix of the observation coordinates in R^2 (e.g., easting and northing).
sim_number	number of simulations. Default value is 1.
seeds	seeds which are used in generation of the initial independent data. Default value is NULL. If non-null, the number of seeds must be equal to sim_number.
sigma.sq	value of sigma square. Default value is 1.
tau.sq	value of tau square. Default value is 0.1.
phi	value of phi. Default value is 1.
nu	starting value of nu, only required for matern covariance model. Default value is 1.5.
n.neighbors	number of neighbors used in the NNGP. Default value is 15.
n_omp	number of threads to be used, value can be more than 1 if source code is compiled with OpenMP support. Default is 1.
cov.model	keyword that specifies the covariance function to be used in modelling the spatial dependence structure among the observations. Supported keywords are: "exponential", "matern", "spherical", and "gaussian" for exponential, Matern, spherical and Gaussian covariance function respectively. Default value is "exponential".
search.type	keyword that specifies type of nearest neighbor search algorithm to be used. Supported keywords are: "brute", "tree" and "cb". "brute" and "tree" provide the same result, though "tree" should be faster.

	"cb" implements fast code book search described in Ra and Kim (1993) modified for NNGP. If locations do not have identical coordinate values on the axis used for the nearest neighbor ordering (see order argument) then "cb" and "brute" should produce identical neighbor sets. However, if there are identical coordinate values on the axis used for nearest neighbor ordering, then "cb" and "brute" might produce different, but equally valid, neighbor sets, e.g., if data are on a grid. Default value is "tree".
stabilization	when we use a very smooth covarince model (lower values of phi for spherical and Gaussian covariance and low phi and high nu for Matern covarinace) in absence of a non-negligble nugget, the correlation process may fail due to computational instability. If stabilization = TRUE, performs stabilization by setting tau.sq = maxtau .sq, sigma.sq $* 1e-06$. Default value is TRUE for cov.model = "expoenential" and FALSE otherwise.
verbose	if TRUE, model specifications along with information regarding OpenMP support and progress of the algorithm is printed to the screen. Otherwise, nothing is printed to the screen. Default value is TRUE.
tol	the input observation coordinates are rounded to this many places after the decimal. The default value is 12.

Value

A list comprising of the following:

coords	the matrix coords.
n.neighbors	the used value of n.neighbors.
cov.model	the used covariance model.
Theta	parameters of covarinace model; accounts for stabilization.
input.data	the $n \times sim_number$ matrix of generated independent data. Here i^{th} column denotes the data corresponding to the i^{th} simulation.
output.data	the $n \times sim_number$ matrix of generated correlated data. Here i^{th} column denotes the data corresponding to the i^{th} simulation.
time	time (in seconds) required after preprocessing data in R, reported using, proc.time().

Author(s)

Arkajyoti Saha <arkajyotisaha93@gmail.com>,
Abhirup Datta <abhidatta@jhu.edu>

Examples

```
set.seed(1)
n <- 1000
coords <- cbind(runif(n,0,1), runif(n,0,1))

sigma.sq = 1
phi = 1
```

```
simulation_result <- BRISC_simulation(coords, sim_number = 3)
```

BRISC_variogram.ci *Function for plotting estimated Variogram and confidence region*

Description

The function BRISC_variogram.ci plots estimated Variogram and associated confience region. BRISC_variogram.ci uses the parameter estimates from BRISC_estimation and associated confidence interval from BRISC_bootstrap.

Usage

```
BRISC_variogram.ci(BRISC_Out, confidence_est,
                   plot.variogram = FALSE)
```

Arguments

BRISC_Out	an object of class BRISC_Out, obtained as an output of BRISC_estimation.
confidence_est	bootstrp sample of the Theta parameters, obtained from BRISC_bootstrap.
plot.variogram	if TRUE, plots the variogram and the associated confidence region. Default is FALSE.

Value

A list comprising of the following:

variogram	variogram and associated confidence region corresponding to lag ranging from 0 to 20, evaluated at 0.01 frequency.
Plot	plots the Variogram and associated confidence region with legends.

Author(s)

Arkajyoti Saha <arkajyotisaha93@gmail.com>,
Abhirup Datta <abhidatta@jhu.edu>

Examples

```
rmvn <- function(n, mu = 0, V = matrix(1)){
  p <- length(mu)
  if(any(is.na(match(dim(V),p))))
    stop("Dimension not right!")
  D <- chol(V)
  t(matrix(rnorm(n*p), ncol=p)%*%D + rep(mu,rep(n,p)))
}
```

```
set.seed(1)
n <- 300
coords <- cbind(runif(n,0,1), runif(n,0,1))

beta <- c(1,5)
x <- cbind(rnorm(n), rnorm(n))

sigma.sq = 1
phi = 5
tau.sq = 0.1

B <- as.matrix(beta)
D <- as.matrix(dist(coords))
R <- exp(-phi*D)
w <- rmvn(1, rep(0,n), sigma.sq*R)

y <- rnorm(n, x%*%B + w, sqrt(tau.sq))

estimation_result <- BRISC_estimation(coords, y, x)
bootstrap_result <- BRISC_bootstrap(estimation_result, n_boot = 10)
varg <- BRISC_variogram.ci(estimation_result,
                           bootstrap_result$boot.Theta,
                           plot.variogram = TRUE)
```

Index

∗ **model**
BRISC_bootstrap, 2
BRISC_correlation, 4
BRISC_decorrelation, 6
BRISC_estimation, 9
BRISC_neighbor, 13
BRISC_order, 14
BRISC_prediction, 16
BRISC_simulation, 18
BRISC_variogram.ci, 20

BRISC_bootstrap, 2
BRISC_correlation, 4, *18*
BRISC_decorrelation, 6
BRISC_estimation, 9
BRISC_neighbor, 13
BRISC_order, 14
BRISC_prediction, 16
BRISC_simulation, 18
BRISC_variogram.ci, 20