

Package ‘BRISC’

July 22, 2018

Type Package

Title Fast Inference for Large Spatial Datasets using BRISC

Version 0.1.0

Maintainer Arkajyoti Saha <arkajyotisaha93@gmail.com>

Author Arkajyoti Saha [aut, cre],
Abhirup Datta [aut],
Jorge Nocedal [ctb],
Naoaki Okazaki [ctb]

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

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 6.0.1

Repository CRAN

Date/Publication 2018-07-22 14:00:02 UTC

R topics documented:

BRISC_bootstrap	2
BRISC_estimation	4
BRISC_prediction	6

Index	9
--------------	----------

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 source 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)
```

Arguments

<code>BRISC_Out</code>	an object of class "BRISC_Out", obtained as an output of <code>BRISC_estimation</code> .
<code>n_boot</code>	number of bootstrap samples. Default value is 100.
<code>h</code>	Number of core to be used in parallel computing setup for bootstrap samples. If <code>h = 1</code> , there is no parallelization. Default value is 1.
<code>n_omp</code>	number of threads to be used, value can be more than 1 if source code is compiled with OpenMP support. Default is 1.
<code>init</code>	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 <code>BRISC_estimate</code> and estimated values of parameters in <code>BRISC_estimate</code> respectively.
<code>verbose</code>	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 FALSE.

Value

A list comprising of the following:

<code>boot.Theta</code>	estimates of spatial covariance parameters corresponding to bootstrap samples.
<code>boot.Beta</code>	estimates of beta corresponding to bootstrap samples.
<code>confidence.interval</code>	confidence intervals corresponding to the parameters.
<code>boot.time</code>	time (in seconds) required to perform the bootstrapping after preprocessing data in R, reported using <code>proc.time()</code> .

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

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, x, y, sigma.sq = 1, tau.sq = 0.1, phi = 1,
                 nu = 0.5, n.neighbors = 15, n_omp = 1, order = "AMMD",
                 cov.model = "exponential", search.type = "tree",
                 verbose = TRUE, eps = 2e-05)
```

Arguments

<code>coords</code>	an $n \times 2$ matrix of the observation coordinates in R^2 (e.g., easting and northing).
<code>x</code>	an $n \times p$ matrix of the covariates in the observation coordinates.
<code>y</code>	a n length vector of response at the observed coordinates.
<code>sigma.sq</code>	starting value of sigma square. Default value is 1.
<code>tau.sq</code>	starting value of tau square. Default value is 0.1.
<code>phi</code>	starting value of phi. Default value is 1.
<code>nu</code>	starting value of nu, only required for matern covariance model. Default value is 0.5.
<code>n.neighbors</code>	number of neighbors used in the NNGP. Default value is 15.
<code>n_omp</code>	number of threads to be used, value can be more than 1 if source code is compiled with OpenMP support. Default is 1.
<code>order</code>	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 "AMMD".

<code>cov.model</code>	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".
<code>search.type</code>	Keyword that specifies type of nearest neighbor search algorithm to be used. Supported keywords are: "tree" and "brute". Both of them provide the same result, though "tree" should be faster. Default value is "tree".
<code>verbose</code>	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 FALSE.
<code>eps</code>	The tolerance to be used in centred finite difference approximation of derivatives. Default value is 2e-05.

Value

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

<code>ord</code>	the vector of indices used to order data necessary for fitting the NNGP model.
<code>coords</code>	the matrix <code>coords[ord,]</code> .
<code>y</code>	the vector <code>y[ord]</code> .
<code>X</code>	the matrix <code>x[ord, ,drop=FALSE]</code> .
<code>n.neighbors</code>	the used value of <code>n.neighbors</code> .
<code>cov.model</code>	the used covariance model.
<code>eps</code>	value of used <code>eps</code> .
<code>init</code>	Initial values of the parameters of the covariance model.
<code>Beta</code>	estimate of beta.
<code>Theta</code>	estimate of parameters of covarinace model.
<code>estimation.time</code>	time (in seconds) required to perform the model fitting after ordering and pre-processing data in R, reported using <code>proc.time()</code> .
<code>BRISC_Object</code>	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 (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))

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, x, y)
estimation_result$Theta ## Gives estimation of covariance model parameters.
estimation_result$Beta ## Gives estimation of Beta
```

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, X.0, coords.0, n_omp = 1, verbose = TRUE)
```

Arguments

<code>BRISC_Out</code>	an object of class "BRISC_Out", obtained as an output of <code>BRISC_estimation</code> .
<code>X.0</code>	the covariates for prediction locations. Its Structure should be identical (including intercept) with that of covariates provided for estimation purpose in <code>BRISC_estimation</code> .
<code>coords.0</code>	the spatial coordinates corresponding to prediction locations. Its caleshould be same as that of <code>coords</code> in <code>BRISC_estimation</code> .
<code>n_omp</code>	number of threads to be used, value can be more than 1 if source code is compiled with OpenMP support. Default is 1.
<code>verbose</code>	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 FALSE.

Value

A list comprising of the following:

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

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))

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[1:800,], x[1:800,], y[1:800])
prediction_result <- BRISC_prediction(estimation_result, x[801:1000,], coords[801:1000,])
```


Index

*Topic **model**

- BRISC_bootstrap, [2](#)
- BRISC_estimation, [4](#)
- BRISC_prediction, [6](#)

- BRISC_bootstrap, [2](#)
- BRISC_estimation, [4](#)
- BRISC_prediction, [6](#)