Package 'BSTZINB'

October 31, 2024

Type Package

Title Association Among Disease Counts and Socio-Environmental Factors

Version 1.0.1

Description Estimation of association between disease or death counts (e.g. COVID-19) and socio-environmental risk factors using a zero-inflated Bayesian spatiotemporal model. Non-spatiotemporal models and/or models without zero-inflation are also included for comparison. Functions to produce corresponding maps are also included. See Chakraborty et al. (2022) <doi:10.1007/s13253-022-00487-1> for more details on the method.

License GPL (>= 3)

Encoding UTF-8

LazyData true

Depends R (>= 2.10)

RoxygenNote 7.3.2

Imports BayesLogit, boot, coda, dplyr, ggplot2, gt, gtsummary, maps, matrixcalc, MCMCpack, msm, reshape, spam, viridis

Suggests knitr, rmarkdown, CorrMixed, ape, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

URL https://github.com/SumanM47/BSTZINB

BugReports https://github.com/SumanM47/BSTZINB/issues

NeedsCompilation no

Author Suman Majumder [cre, aut, cph],
Yoon-Bae Jun [aut, cph],
Sounak Chakraborty [ctb],
Chae-Young Lim [ctb],
Tanujit Dey [ctb]

Maintainer Suman Majumder <smajumd2@gmail.com>

Repository CRAN

Date/Publication 2024-10-31 22:50:02 UTC

Contents

`BNB` *Fit a Bayesian Negative Binomial Model*

Description

Generate posterior samples for the parameters in a Bayesian Negative Binomial Model

Usage

```
BNB(y, X, A,
           nchain=3, niter=100, nburn=20, nthin=1)
```

Arguments

`y`	vector of counts, must be non-negative
`X`	matrix of covariates, numeric
`A`	adjacency matrix, numeric
`nchain`	positive integer, number of MCMC chains to be run
`niter`	positive integer, number of iterations in each chain
`nburn`	non-negative integer, number of iterations to be discarded as burn-in samples
`nthin`	positive integer, thinning interval

Value

list of posterior samples of the parameters of the model

Examples

```
data(simdat)
y <- simdat$y
X <- cbind(simdat$V1,simdat$x)
data(county.adjacency)
data(USAcities)
IAcities <- subset(USAcities,state_id=="IA")
countyname <- unique(IAcities$county_name)
A <- get_adj_mat(county.adjacency,countyname,c("IA"))

res0 <- BNB(y, X, A, nchain=2, niter=100, nburn=20, nthin=1)
```

BSTNB *Fit a Bayesian Spatiotemporal Negative Binomial model*

Description

Generate posterior samples for the parameters in a Bayesian Spatiotemporal Negative Binomial Model

Usage

```
BSTNB(y,X,A,
            nchain=3,niter=100,nburn=20,nthin=1)
```

Arguments

y	vector of counts, must be non-negative
X	matrix of covariates, numeric
A	adjacency matrix, numeric
nchain	positive integer, number of MCMC chains to be run
niter	positive integer, number of iterations in each chain
nburn	non-negative integer, number of iterations to be discarded as burn-in samples
nthin	positive integer, thinning interval

Value

list of posterior samples of the parameters of the model

Examples

```
data(simdat)
y <- simdat$y
X <- cbind(simdat$V1,simdat$x)
data(county.adjacency)
data(USAcities)
IAcities <- subset(USAcities,state_id=="IA")
countyname <- unique(IAcities$county_name)
A <- get_adj_mat(county.adjacency,countyname,c("IA"))

res2 <- BSTNB(y, X, A, nchain=2, niter=100, nburn=20, nthin=1)
```

BSTZINB *Fit a Bayesian Spatiotemporal Zero Inflated Negative Binomial model*

Description

Generate posterior samples for the parameters in a Bayesian Spatiotemporal Zero Inflated Negative Binomial Model

Usage

```
BSTZINB(y,X,A,LinearT = TRUE,
            nchain=3,niter=100,nburn=20,nthin=1)
```

Arguments

y	vector of counts, must be non-negative
X	matrix of covariates, numeric
A	adjacency matrix, numeric
LinearT	logical, whether to fit a linear or non-linear temporal trend
nchain	positive integer, number of MCMC chains to be run
niter	positive integer, number of iterations in each chain
nburn	non-negative integer, number of iterations to be discarded as burn-in samples
nthin	positive integer, thinning interval

Value

list of posterior samples of the parameters of the model

Examples

```
data(simdat)
y <- simdat$y
X <- cbind(simdat$V1,simdat$x)
data(county.adjacency)
data(USAcities)
IAcities <- subset(USAcities,state_id=="IA")
countyname <- unique(IAcities$county_name)
A <- get_adj_mat(county.adjacency,countyname,c("IA"))

res3 <- BSTZINB(y, X, A, LinearT=TRUE, nchain=2, niter=100, nburn=20, nthin=1)
```

BZINB *Fit a Bayesian Zero Inflated Negative Binomial Model*

Description

Generate posterior samples for the parameters in a Bayesian Zero Inflated Negative Binomial Model

Usage

```
BZINB(y,X,A,
             nchain=3,niter=100,nburn=20,nthin=1)
```

Arguments

y	vector of counts, must be non-negative
X	matrix of covariates, numeric
A	adjacency matrix, numeric
nchain	positive integer, number of MCMC chains to be run
niter	positive integer, number of iterations in each chain
nburn	non-negative integer, number of iterations to be discarded as burn-in samples
nthin	positive integer, thinning interval

Value

list of posterior samples of the parameters of the model

Examples

```
data(simdat)
y <- simdat$y
X <- cbind(simdat$V1,simdat$x)
data(county.adjacency)
data(USAcities)
IAcities <- subset(USAcities,state_id=="IA")
countyname <- unique(IAcities$county_name)
A <- get_adj_mat(county.adjacency,countyname,c("IA"))

res1 <- BSTZINB(y, X, A, nchain=2, niter=100, nburn=20, nthin=1)
```

---

compute_NB_DIC *DIC for BSTNB or BNB fitted objects*

Description

Computes DIC for a BSTNB or BNB fitted object

Usage

```
compute_NB_DIC(y,bstfit,lastit,nchain)
```

Arguments

y	vector of counts, must be non-negative, the response used for fitting a BSTNB or BSTP model
bstfit	BSTNB or BNB fitted object
lastit	positive integer, size of the chain used to fit BSTZINB
nchain	positive integer, number of chains used to fit BSTZINB

Value

DIC value

Examples

```
data(simdat)
y <- simdat$y
X <- cbind(simdat$V1,simdat$x)
data(county.adjacency)
data(USAcities)
IAcities <- subset(USAcities,state_id=="IA")
countyname <- unique(IAcities$county_name)
A <- get_adj_mat(county.adjacency,countyname,c("IA"))
```

```
res2 <- BSTNB(y, X, A, nchain=3, niter=100, nburn=20, nthin=1)
compute_NB_DIC(y,res2,lastit=(100-20)/1,nchain=3)
```

`compute_ZINB_DIC` *DIC for BSTZINB fitted objects*

Description

Computes DIC for a BSTZINB fitted object

Usage

```
compute_ZINB_DIC(y,bstfit,lastit,nchain)
```

Arguments

`y`	vector of counts, must be non-negative, the response used for fitting a BSTZINB model
`bstfit`	BSTZINB fitted object
`lastit`	positive integer, size of the chain used to fit BSTZINB
`nchain`	positive integer, number of chains used to fit BSTZINB

Value

DIC value

Examples

```
data(simdat)
y <- simdat$y
X <- cbind(simdat$V1,simdat$x)
data(county.adjacency)
data(USAcities)
IAcities <- subset(USAcities,state_id=="IA")
countyname <- unique(IAcities$county_name)
A <- get_adj_mat(county.adjacency,countyname,c("IA"))

res3 <- BSTZINB(y, X, A, LinearT=TRUE, nchain=3, niter=100, nburn=20, nthin=1)
compute_ZINB_DIC(y,res3,lastit=(100-20)/1,nchain=3)
```

conv.test *convergence test for parameters in the fitted objects*

Description

Conducts a test of convergence for a given parameter in the fitted objects using the posterior samples for the said parameter

Usage

```
conv.test(params,nchain=3,thshold=1.96)
```

Arguments

params	numeric matrix of dimension 2 (iterations x number of parameters, single chain) or 3 (iterations x number of parameters x chain, multiple chains) of posterior samples
nchain	positive integer, number of chains used to fit BSTZINB, BSTNB or BSTP
thshold	positive scalar, the threshold for testing the convergence. Defaults to 1.96

Value

logical vector indicating whether convergence was achieved or not

Examples

```
data(simdat)
y <- simdat$y
X <- cbind(simdat$V1,simdat$x)
data(county.adjacency)
data(USAcities)
IAcities <- subset(USAcities,state_id=="IA")
countyname <- unique(IAcities$county_name)
A <- get_adj_mat(county.adjacency,countyname,c("IA"))

res3 <- BSTZINB(y, X, A, LinearT=TRUE, nchain=3, niter=100, nburn=20, nthin=1)
conv.test(res3$Alpha,nchain=3)
```

`county.adjacency` *county.adjacency: A dataframe containing neighborhood information for counties in the US*

Description

Data set containing neighborhood information for counties in the US, to be used to create adjacency matrices

Usage

```
county.adjacency
```

Format

`county.adjacency`**:**
A dataframe with 22200 rows and 4 columns

`get_adj_mat` *Adjacency matrix for counties of one or many states in the United States*

Description

Creates the adjacency matrix for the supplied counties within the United States using the available neighborhood information

Usage

```
get_adj_mat(county.adjacency,Countyvec,Statevec)
```

Arguments

`county.adjacency`	data frame containing the neighborhood information for the counties of the entire US
`Countyvec`	character vector containing the names of the counties for which the adjacency matrix is to be computed
`Statevec`	character vector containing the names of the states the supplied counties belong to

Value

the corresponding adjacency matrix

Examples

```
data(county.adjacency)
data(USAcities)
IAcities <- subset(USAcities,state_id=="IA")
countyname <- unique(IAcities$county_name)
A <- get_adj_mat(county.adjacency,countyname,c("IA"))
```

---

qRankPar *Bar plot for time-averaged log-q estimates over quantile-representative counties (descending order)*

Description

Produce a descending order of bar plot for time-averaged log-q estimates over quantile-representative counties

Usage

```
qRankPar(state.set,cname,bstfit,vn=12,
               cex.title=18, cex.lab=18, cex.legend=18)
```

Arguments

state.set	character vector of set of states on which the the graphics is to be made
cname	character vector of the names of the counties
bstfit	the fitted data for BSTP, BSTNB or BSTZINB
vn	positive integer, number of sample counties to display
cex.title	Positive number to control the size of the text of the main title. Defaults to 18.
cex.lab	Positive number to control the size of the text in the axes labels. Defaults to 18.
cex.legend	Positive number to control the size of the text in the legend. Defaults to 18.

Value

bar graph

Examples

```
data(simdat)
y <- simdat$y
X <- cbind(simdat$V1,simdat$x)
data(county.adjacency)
data(USAcities)
IAcities <- subset(USAcities,state_id=="IA")
countyname <- unique(IAcities$county_name)
A <- get_adj_mat(county.adjacency,countyname,c("IA"))
```

```
res3 <- BSTZINB(y, X, A, LinearT=TRUE, nchain=3, niter=100, nburn=20, nthin=1)
qRankPar(state.set=c("IA"),cname=countyname,bstfit=res3,vn=12,
         cex.title=18, cex.lab=12, cex.legend=12)
```

---

qRankParTop *Bar plot for time-averaged log-q estimates over top ranking counties (descending order)*

Description

Produce a descending order of bar plot for time-averaged log-q estimates over top ranking counties

Usage

```
qRankParTop(state.set,cname,bstfit,vn=12,
                   cex.title=18, cex.lab=18, cex.legend=18)
```

Arguments

state.set	character vector of set of states on which the the graphics is to be made
cname	character vector of the names of the counties
bstfit	the fitted data for BSTP, BSTNB or BSTZINB
vn	positive integer, number of sample counties to display
cex.title	Positive number to control the size of the text of the main title. Defaults to 18.
cex.lab	Positive number to control the size of the text in the axes labels. Defaults to 18.
cex.legend	Positive number to control the size of the text in the legend. Defaults to 18.

Value

bar graph

Examples

```
data(simdat)
y <- simdat$y
X <- cbind(simdat$V1,simdat$x)
data(county.adjacency)
data(USAcities)
IAcities <- subset(USAcities,state_id=="IA")
countyname <- unique(IAcities$county_name)
A <- get_adj_mat(county.adjacency,countyname,c("IA"))

res3 <- BSTZINB(y, X, A, LinearT=TRUE, nchain=3, niter=100, nburn=20, nthin=1)
qRankParTop(state.set=c("IA"),cname=countyname,bstfit=res3,vn=12,
```

```
                cex.title=18, cex.lab=12, cex.legend=12)
```

ResultTableSummary *Summary Table for a fitted object*

Description

Generates a short summary table for a fitted object using BSTP, BSTNB or BSTZINB function

Usage

```
ResultTableSummary(bstfit)
```

Arguments

bstfit	fitted object using the function BSTP, BSTNB or BSTZINB

Value

summary table

Examples

```
data(simdat)
y <- simdat$y
X <- cbind(simdat$V1,simdat$x)
data(county.adjacency)
data(USAcities)
IAcities <- subset(USAcities,state_id=="IA")
countyname <- unique(IAcities$county_name)
A <- get_adj_mat(county.adjacency,countyname,c("IA"))

res3 <- BSTZINB(y, X, A, LinearT=TRUE, nchain=3, niter=100, nburn=20, nthin=1)
ResultTableSummary(res3)
```

`ResultTableSummary2` *Generate a summary table of the outputs all different methods given the data*

Description

Fits BSTP, BSTNB and BSTZINB (with linear or non-linear temporal trend) to a given data and summarizes the results in a table

Usage

```
ResultTableSummary2(y,X,A,LinearT=FALSE,
                            nchain=3,niter=100,nburn=20,nthin=1)
```

Arguments

`y`	vector of counts, must be non-negative
`X`	matrix of covariates, numeric
`A`	adjacency matrix, numeric
`LinearT`	logical, whether to fit a linear or non-linear temporal trend
`nchain`	positive integer, number of MCMC chains to be run
`niter`	positive integer, number of iterations in each chain
`nburn`	non-negative integer, number of iterations to be discarded as burn-in samples
`nthin`	positive integer, thinning interval

Value

summary tables for the different methods

Examples

```
data(simdat)
y <- simdat$y
X <- cbind(simdat$V1,simdat$x)
data(county.adjacency)
data(USAcities)
IAcities <- subset(USAcities,state_id=="IA")
countyname <- unique(IAcities$county_name)
A <- get_adj_mat(county.adjacency,countyname,c("IA"))

ResultTableSummary2(y, X, A, LinearT=TRUE, nchain=3, niter=100, nburn=20, nthin=1)
```

simdat — *simdat: A simulated dataset containing response and covariates with region and time information*

Description

Synthetic dataframe to be used for examples and trial runs

Usage

```
simdat
```

Format

`simdat`:

A dataframe with 2376 rows and 5 columns: sid (region ID), tid (timepoint), y (count response), V1 (intercept), and x (covariate).

TimetrendCurve — *Time-trend curve over the study time domain for counties in the US*

Description

Produce a time-trend curve over the study time domain for counties in the US

Usage

```
TimetrendCurve(bstfit,cname,vn=5,smooth.mode=TRUE,
                     cex.title=18, cex.lab=18, cex.legend=18)
```

Arguments

`bstfit`	fitted object from BSTP, BSTNB or BSTZINB
`cname`	character vector of county names to use
`vn`	positive integer, number of sample counties to use
`smooth.mode`	logical, should splines be fitted to make it smooth
`cex.title`	Positive number to control the size of the text of the main title. Defaults to 18.
`cex.lab`	Positive number to control the size of the text in the axes labels. Defaults to 18.
`cex.legend`	Positive number to control the size of the text in the legend. Defaults to 18.

Value

time-trend curves

Examples

```
data(simdat)
y <- simdat$y
X <- cbind(simdat$V1,simdat$x)
data(county.adjacency)
data(USAcities)
IAcities <- subset(USAcities,state_id=="IA")
countyname <- unique(IAcities$county_name)
A <- get_adj_mat(county.adjacency,countyname,c("IA"))

res3 <- BSTZINB(y, X, A, LinearT=TRUE, nchain=3, niter=100, nburn=20, nthin=1)
TimetrendCurve(res3,cname=countyname,vn=5,smooth.mode=TRUE,cex.title=18, cex.lab=12, cex.legend=12)
```

USAcities *USAcities: A dataset containing state and county information for the cities in the United States*

Description

Dataframe to be used internally to make maps and get county information

Usage

```
USAcities
```

Format

`USAcities`**:**

A dataframe with 3232 rows and 4 columns: state_id (State abbreviation), county_name (County name), county_fips (FIPS codes for the counties) and population (County population).

USDmapCount *Draw spatial maps of various quantities over regions in the US*

Description

Creates a map of any given quantity (at a selected time or averaged over time) for regions in the US specified by state and county

Usage

```
USDmapCount(state.sel,dat,scol,tcol=NULL,tsel=NULL,cname,uplim=NULL)
```

Arguments

state.sel	character vector giving the selected states
dat	data frame having named components: y - the necessary quantity (numeric), sid - the region indices, tid - the time indices
scol	column index of the spatial regions
tcol	(optional) column index of the time points
tsel	(optional) selected time point
cname	character vector of county names, must match those in USAcities
uplim	(optional) numeric, upper limit for the given quantity

Value

spatial map of the required quantity over the specified region

Examples

```
data(simdat)
data(county.adjacency)
data(USAcities)
IAcities <- subset(USAcities,state_id=="IA")
countyname <- unique(IAcities$county_name)
USDmapCount(state.sel="IA",dat=simdat,scol=1,tcol=2,tsel=150,cname=countyname)
```

Index