Package 'SSNbayes'

January 20, 2025

Type Package

Title Bayesian Spatio-Temporal Analysis in Stream Networks

Version 0.0.3

Depends R (>= 4.0.0)

Imports plyr, dplyr, rstan, sf, methods, SSN2

Description Fits Bayesian spatio-temporal models and makes predictions on stream networks using the approach by Santos-Fernandez, Edgar, et al. (2022).``Bayesian spatio-temporal models for stream networks". <arXiv:2103.03538>. In these models, spatial dependence is captured using stream distance and flow connectivity, while temporal autocorrelation is modelled using vector autoregression methods.

License GPL-2

Encoding UTF-8

RoxygenNote 7.2.1

Suggests rmarkdown, knitr, testthat (>= 3.0.0)

VignetteBuilder knitr

URL https://github.com/EdgarSantos-Fernandez/SSNbayes

BugReports https://github.com/EdgarSantos-Fernandez/SSNbayes/issues

Config/testthat/edition 3

NeedsCompilation no

Author Edgar Santos-Fernandez [aut, cre, cph]

Maintainer Edgar Santos-Fernandez <santosfe@qut.edu.au>

Repository CRAN

Date/Publication 2023-12-03 17:20:02 UTC

Contents

`collapse` *Collapses a SpatialStreamNetwork object into a data frame*

Description

Collapses a SpatialStreamNetwork object into a data frame

Usage

```
collapse(ssn, par = "afvArea")
```

Arguments

`ssn`	An S4 SpatialStreamNetwork object created with SSN2 package.
`par`	A spatial parameter such as the computed_afv (additive function value).

Details

The parameters (par) has to be present in the observed data frame via ssn_get_data(n, name = "obs"). More details of the argument par can be found in the additive.function() from SSN .

Value

A data frame with the lat and long of the line segments in the network. The column line_id refers to the ID of the line.

Examples

```
#require("SSN2")
#path <- system.file("extdata/clearwater.ssn", package = "SSNbayes")
#ssn <- SSN2::ssn_import(path, predpts = "preds", overwrite  = TRUE)
#t.df <- collapse(ssn, par = 'afvArea')
```

dist_weight_mat *Creates a list containing the stream distances and weights*

Description

Creates a list containing the stream distances and weights

Usage

```
dist_weight_mat(path = path, net = 1, addfunccol = "addfunccol")
```

Arguments

path	Path to the files
net	(optional) A network from the SSN2 object
addfunccol	(optional) A parameter to compute the spatial weights

Value

A list of matrices

Examples

```
path <- system.file("extdata/clearwater.ssn", package = "SSNbayes")
mat_all <- dist_weight_mat(path, net = 2, addfunccol='afvArea')
```

dist_weight_mat_preds *Creates a list of distances and weights between observed and prediction sites*

Description

The output matrices are symmetric except the hydrologic distance matrix D.

Usage

```
dist_weight_mat_preds(path = path, net = 1, addfunccol = "addfunccol")
```

Arguments

path	Path with the name of the SpatialStreamNetwork object
net	(optional) A network from the SpatialStreamNetwork object
addfunccol	(optional) A parameter to compute the spatial weights

Value

A list of matrices

Examples

```
## Not run:
path <- system.file("extdata/clearwater.ssn", package = "SSNbayes")
mat_all_pred <- dist_weight_mat_preds(path, net = 2, addfunccol='afvArea')
## End(Not run)
```

---

krig

Internal function used to perform spatio-temporal prediction in R using a stanfit object from ssnbayes()

Description

Use predict.ssnbayes() instead. It will take an observed and a prediction data frame. It requires the same number of observation/locations per day. It requires location id (locID) and points id (pid). The locID are unique for each site. The pid is unique for each observation. Missing values are allowed in the response but not in the covariates.

Usage

```
krig(
  object = object,
  mat_all_preds = mat_all_preds,
  nsamples = 10,
  start = 1,
  chunk_size = 50,
  obs_data = obs_data,
  pred_data = pred_data,
  net = net,
  seed = seed
)
```

Arguments

object	A stanfit object returned from ssnbayes
mat_all_preds	A list with the distance/weights matrices
nsamples	The number of samples to draw from the posterior distributions. (nsamples <= iter)
start	(optional) The starting location id
chunk_size	(optional) the number of locID to make prediction from
obs_data	The observed data frame
pred_data	The predicted data frame
net	(optional) Network from the SSN object
seed	(optional) A seed for reproducibility

Value

A data frame

Author(s)

Edgar Santos-Fernandez

`mylm` *A simple modeling function using a formula and data*

Description

A simple modeling function using a formula and data

A simple modeling function using a formula and data

Usage

```
mylm(formula, data)
```


```
mylm(formula, data)
```


Arguments

`formula`	A formula as in lm()
`data`	A data.frame containing the elements specified in the formula

Value

A list of matrices

A list of matrices

Author(s)

Jay ver Hoef

Examples

```
options(na.action='na.pass')
data("iris")
out_list = mylm(formula = Petal.Length ~ Sepal.Length + Sepal.Width, data = iris)
options(na.action='na.pass')
data("iris")
out_list = mylm(formula = Petal.Length ~ Sepal.Length + Sepal.Width, data = iris)
```

`predict.ssnbayes` *Performs spatio-temporal prediction in R using an ssnbayes object from a fitted model.*

Description

It will take an observed and a prediction data frame. It requires the same number of observation/locations per day. It requires location id (locID) and points id (pid). The locID are unique for each site. The pid is unique for each observation. Missing values are allowed in the response but not in the covariates.

Usage

```
## S3 method for class 'ssnbayes'
predict(
  object = object,
  ...,
  path = path,
  obs_data = obs_data,
  pred_data = pred_data,
  net = net,
  nsamples = nsamples,
  addfunccol = addfunccol,
  locID_pred = locID_pred,
  chunk_size = chunk_size,
  seed = seed
)
```

Arguments

`object`	A stanfit object returned from ssnbayes
`...`	Other parameters
`path`	Path with the name of the SpatialStreamNetwork object
`obs_data`	The observed data frame
`pred_data`	The predicted data frame
`net`	(optional) Network from the SSN object
`nsamples`	The number of samples to draw from the posterior distributions. (nsamples <= iter)
`addfunccol`	The variable used for spatial weights
`locID_pred`	(optional) the location id for the predictions. Used when the number of pred locations is large.
`chunk_size`	(optional) the number of locID to make prediction from
`seed`	(optional) A seed for reproducibility

Details

The returned data frame is melted to produce a long dataset. See examples.

Value

A data frame with the location (locID), time point (date), plus the MCMC draws from the posterior from 1 to the number of iterations. The locID0 column is an internal consecutive location ID (locID) produced in the predictions, starting at max(locID(observed data)) + 1. It is used internally in the way predictions are made in chunks.

Author(s)

Edgar Santos-Fernandez

Examples

```
#require('SSNdata')
#clear_preds <- readRDS(system.file("extdata/clear_preds.RDS", package = "SSNdata"))
#clear_preds$y <- NA
#pred <- predict(object = fit_ar,
#                path = path,
#                obs_data = clear,
#                pred_data = clear_preds,
#                net = 2,
#                nsamples = 100, # numb of samples from the posterior
#                addfunccol = 'afvArea', # var for spatial weights
#                locID_pred = locID_pred,
#                chunk_size = 60)
```

---

pred_ssnbayes *Internal function used to perform spatio-temporal prediction in R using a stanfit object from ssnbayes()*

Description

Use predict.ssnbayes() instead. It will take an observed and a prediction data frame. It requires the same number of observation/locations per day. It requires location id (locID) and points id (pid). The locID are unique for each site. The pid is unique for each observation. Missing values are allowed in the response but not in the covariates.

Usage

```
pred_ssnbayes(
  object = object,
  path = path,
  obs_data = obs_data,
  pred_data = pred_data,
```

```
  net = 1,
  nsamples = 100,
  addfunccol = "afvArea",
  locID_pred = locID_pred,
  chunk_size = chunk_size,
  seed = seed
)
```

Arguments

object	A stanfit object returned from ssnbayes
path	Path with the name of the SpatialStreamNetwork object
obs_data	The observed data frame
pred_data	The predicted data frame
net	(optional) Network from the SSN object
nsamples	The number of samples to draw from the posterior distributions. (nsamples <= iter)
addfunccol	The variable used for spatial weights
locID_pred	(optional) the location id for the predictions. Used when the number of pred locations is large.
chunk_size	(optional) the number of locID to make prediction from
seed	(optional) A seed for reproducibility

Value

A data frame

Author(s)

Edgar Santos-Fernandez

Examples

```
#pred <- pred_ssnbayes(path = path,
#obs_data = clear,
#stanfit = fit_ar,
#pred_data = preds,
#net = 2,
#nsamples = 100, # number of samples to use from the posterior in the stanfit object
#addfunccol = 'afvArea') # variable used for spatial weights
```

ssnbayes *Fits a mixed linear regression model using Stan*

Description

It requires the same number of observation/locations per day. It requires location id (locID) and points id (pid). The locID are unique for each site. The pid is unique for each observation. Missing values are allowed in the response but not in the covariates.

Usage

```
ssnbayes(
  formula = formula,
  data = data,
  path = path,
  time_method = time_method,
  space_method = space_method,
  iter = 3000,
  warmup = 1500,
  chains = 3,
  refresh = max(iter/100, 1),
  net = 1,
  addfunccol = addfunccol,
  loglik = FALSE,
  seed = seed
)
```

Arguments

formula	A formula as in lm()
data	A long data frame containing the locations, dates, covariates and the response variable. It has to have the locID and date. No missing values are allowed in the covariates. The order in this data.fame MUST be: spatial locations (1 to S) at time t=1, then locations (1 to S) at t=2 and so on.
path	Path with the name of the SpatialStreamNetwork object
time_method	A list specifying the temporal structure (ar = Autorregressive; var = Vector autorregression) and coumn in the data with the time variable.
space_method	A list defining if use or not of an SSN object and the spatial correlation structure. The second element is the spatial covariance structure. A 3rd element is a list with the lon and lat for Euclidean distance models.
iter	Number of iterations
warmup	Warm up samples
chains	Number of chains
refresh	Sampler refreshing rate

net	The network id (optional). Used when the SSN object contains multiple networks.
addfunccol	Variable to compute the additive function. Used to compute the spatial weights.
loglik	Logic parameter denoting if the loglik will be computed by the model.
seed	(optional) A seed for reproducibility

Details

Missing values are not allowed in the covariates and they must be imputed before using ssnbayes(). Many options can be found in https://cran.r-project.org/web/views/MissingData.html The pid in the data has to be consecutive from 1 to the number of observations. Users can use the SpatialStreamNetwork created with the SSN package. This will provide the spatial stream information used to compute covariance matrices. If that is the case, the data has to have point ids (pid) matching the ones in SSN distance matrices, so that a mapping can occur.

Value

A list with the model fit

It returns a ssnbayes object (similar to stan returns). It includes the formula used to fit the model. The output can be transformed into the stanfit class using class(fits) <- c("stanfit").

Author(s)

Edgar Santos-Fernandez

Examples

```
## Not run:
#options(mc.cores = parallel::detectCores())
# Import SpatialStreamNetwork object
#path <- system.file("extdata/clearwater.ssn", package = "SSNbayes")
#n <- SSN2::ssn_import(path, predpts = "preds", overwrite  = TRUE)
## Imports a data.frame containing observations and covariates
#clear <- readRDS(system.file("extdata/clear_obs.RDS", package = "SSNbayes"))
#fit_ar <- ssnbayes(formula = y ~ SLOPE + elev + h2o_area + air_temp + sin + cos,
#                   data = clear,
#                   path = path,
#                   time_method = list("ar", "date"),
#                   space_method = list('use_ssn', c("Exponential.taildown")),
#                   iter = 2000,
#                   warmup = 1000,
#                   chains = 3,
#                   net = 2, # second network on the ssn object
#                   addfunccol='afvArea')
#space_method options examples
#use list('no_ssn', 'Exponential.Euclid', c('lon', 'lat')) if no ssn object is available

## End(Not run)
```

Index