Package: geostan (via r-universe)

August 20, 2024

Title Bayesian Spatial Analysis

Version 0.6.2

Date 2024-05-31

URL https://connordonegan.github.io/geostan/

BugReports https://github.com/ConnorDonegan/geostan/issues

Description For spatial data analysis; provides exploratory spatial analysis tools, spatial regression models, disease mapping models, model diagnostics, and special methods for inference with small area survey data (e.g., the America Community Survey (ACS)) and censored population health surveillance data. Models are pre-specified using the Stan programming language, a platform for Bayesian inference using Markov chain Monte Carlo (MCMC). References: Carpenter et al. (2017) <doi:10.18637/jss.v076.i01>; Donegan (2021) <doi:10.31219/osf.io/3ey65>; Donegan (2022) <doi:10.21105/joss.04716>; Donegan, Chun and Hughes (2020) <doi:10.1016/j.spasta.2020.100450>; Donegan, Chun and Griffith (2021) <doi:10.3390/ijerph18136856>; Morris et al. (2019) <doi:10.1016/j.sste.2019.100301>.

License GPL (>= 3)

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Biarch true

Depends R (>= 3.4)

Imports spdep (>= 1.0), sf (>= 1.0-10), ggplot2 (>= 3.0.0), methods, graphics, stats, MASS, truncnorm, signs, gridExtra, utils, Matrix (>= 1.3), Rcpp (>= 0.12.0), RcppParallel (>= 5.0.1), rstan (>= 2.26.0), rstantools (>= 2.1.1)

LinkingTo BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), RcppParallel (>= 5.0.1), rstan (>= 2.26.0), StanHeaders (>= 2.26.0)

Suggests testthat, knitr, rmarkdown, bayesplot

SystemRequirements GNU make

VignetteBuilder knitr

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

RemoteUrl https://github.com/connordonegan/geostan

RemoteRef HEAD

RemoteSha c330dfcf72a1b5bd0d2371aa75c81d72cbddaec8

Contents

`geostan-package` *The geostan R package.*

Description

Bayesian spatial modeling powered by Stan. **geostan** provides access to a variety of hierarchical spatial models using the R formula interface, supporting a complete spatial analysis workflow with a suite of spatial analysis tools. It is designed primarily for public health and social science research but is generally applicable to modeling areal data. Unique features of the package include its spatial measurement error model (for inference with small area estimates such as those from the American Community Survey), its fast proper conditional autoregressive (CAR) and simultaneous autoregressive (SAR) models, and its eigenvector spatial filtering (ESF) models. The package also supports spatial regression with raster layers.

Author(s)

Maintainer: Connor Donegan <connor.donegan@gmail.com> (ORCID)

Other contributors:

- Mitzi Morris [contributor]
- Amy Tims [contributor]

References

Carpenter, B., Gelman, A., Hoffman, M.D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., Riddell, A., 2017. Stan: A probabilistic programming language. Journal of statistical software 76. doi:10.18637/jss.v076.i01.

Donegan, C., Y. Chun and A. E. Hughes (2020). Bayesian estimation of spatial filters with Moran's Eigenvectors and hierarchical shrinkage priors. *Spatial Statistics*. doi:10.1016/j.spasta.2020.100450 (open access: doi:10.31219/osf.io/fah3z).

Donegan, Connor and Chun, Yongwan and Griffith, Daniel A. (2021). Modeling community health with areal data: Bayesian inference with survey standard errors and spatial structure. *Int. J. Env. Res. and Public Health* 18 (13): 6856. doi:10.3390/ijerph18136856. Supplementary material: `https://github.com/ConnorDonegan/survey-HBM`.

Donegan, Connor (2021). Building spatial conditional autoregressive models in the Stan programming language. *OSF Preprints*. doi:10.31219/osf.io/3ey65.

Donegan, Connor (2022) geostan: An R package for Bayesian spatial analysis. *The Journal of Open Source Software*. 7, no. 79: 4716. doi:10.21105/joss.04716.

Gabry, J., Goodrich, B. and Lysy, M. (2020). rstantools: Tools for developers of R packages interfacing with Stan. R package version 2.1.1 `https://mc-stan.org/rstantools/`.

Morris, M., Wheeler-Martin, K., Simpson, D., Mooney, S. J., Gelman, A., & DiMaggio, C. (2019). Bayesian hierarchical spatial models: Implementing the Besag York Mollié model in stan. Spatial and spatio-temporal epidemiology, 31, 100301. doi:10.1016/j.sste.2019.100301.

Stan Development Team (2019). RStan: the R interface to Stan. R package version 2.19.2. `https://mc-stan.org`

See Also

Useful links:

- `https://connordonegan.github.io/geostan/`
- Report bugs at `https://github.com/ConnorDonegan/geostan/issues`

edges *Edge list*

Description

Creates a list of connected nodes following the graph representation of a spatial connectivity matrix.

Usage

```
edges(C, unique_pairs_only = TRUE, shape)
```

Arguments

`C`	A connectivity matrix where connection between two nodes is indicated by non-zero entries.
`unique_pairs_only`	By default, only unique pairs of nodes (i, j) will be included in the output.
`shape`	Optional spatial object (geometry) to which `C` refers. If given, the function returns an `sf` object.

Details

This is used internally for `stan_icar`, can be helpful for creating the scaling factor for BYM2 models fit with `stan_icar`, and can be used for visualizing a spatial connectivity matrix.

Value

If `shape` is missing, this returns a `data.frame` with three columns. The first two columns (`node1` and `node2`) contain the indices of connected pairs of nodes; only unique pairs of nodes are included (unless `unique_pairs_only = FALSE`). The third column (`weight`) contains the corresponding matrix element, `C[node1, node2]`.

If `shape` is provided, the results are joined to an `sf` object so the connections can be visualized.

See Also

`shape2mat`, `prep_icar_data`, `stan_icar`

Examples

```
data(sentencing)
C <- shape2mat(sentencing)
nbs <- edges(C)
head(nbs)

## similar to:
head(Matrix::summary(C))
head(Matrix::summary(shape2mat(georgia, "W")))
```

```
## add geometry for plotting
library(sf)
E <- edges(C, shape = sentencing)
g1 = st_geometry(E)
g2 = st_geometry(sentencing)
plot(g1, lwd = .2)
plot(g2, add = TRUE)
```

`expected_mc` *Expected value of the residual Moran coefficient*

Description

Expected value for the Moran coefficient of model residuals under the null hypothesis of no spatial autocorrelation.

Usage

```
expected_mc(X, C)
```

Arguments

`X`	model matrix, including column of ones.
`C`	Connectivity matrix.

Value

Returns a numeric value.

Source

Chun, Yongwan and Griffith, Daniel A. (2013). Spatial statistics and geostatistics. Sage, p. 18.

Examples

```
data(georgia)
C <- shape2mat(georgia)
X <- model.matrix(~ ICE + college, georgia)
expected_mc(X, C)
```

make_EV *Extract eigenfunctions of a connectivity matrix for spatial filtering*

Description

Extract eigenfunctions of a connectivity matrix for spatial filtering

Usage

```
make_EV(C, nsa = FALSE, threshold = 0.2, values = FALSE)
```

Arguments

`C`	A binary spatial weights matrix. See `shape2mat`.
`nsa`	Logical. Default of `nsa = FALSE` excludes eigenvectors capturing negative spatial autocorrelation. Setting `nsa = TRUE` will result in a candidate set of EVs that contains eigenvectors representing positive and negative SA.
`threshold`	Defaults to `threshold=0.2` to exclude eigenvectors representing spatial autocorrelation levels that are less than `threshold` times the maximum possible Moran coefficient achievable for the given spatial connectivity matrix. If `theshold = 0`, all eigenvectors will be returned (however, the eigenvector of constants (with eigenvalue of zero) will be dropped automatically).
`values`	Should eigenvalues be returned also? Defaults to `FALSE`.

Details

Returns a set of eigenvectors related to the Moran coefficient (MC), limited to those eigenvectors with |MC| > `threshold` if `nsa = TRUE` or MC > `threshold` if `nsa = FALSE`, optionally with corresponding eigenvalues.

Value

A `data.frame` of eigenvectors for spatial filtering. If `values=TRUE` then a named list is returned with elements `eigenvectors` and `eigenvalues`.

Source

Daniel Griffith and Yongwan Chun. 2014. "Spatial Autocorrelation and Spatial Filtering." in M. M. Fischer and P. Nijkamp (eds.), *Handbook of Regional Science.* Springer.

See Also

stan_esf, mc

Examples

```
library(ggplot2)
data(georgia)
C <- shape2mat(georgia, style = "B")
EV <- make_EV(C)
head(EV)

ggplot(georgia) +
  geom_sf(aes(fill = EV[,1])) +
  scale_fill_gradient2()
```

---

mc — *The Moran coefficient (Moran's I)*

Description

The Moran coefficient, a measure of spatial autocorrelation (also known as Global Moran's I)

Usage

```
mc(x, w, digits = 3, warn = TRUE, na.rm = FALSE)
```

Arguments

`x`	Numeric vector of input values, length n.
`w`	An n x n spatial connectivity matrix. See `shape2mat`.
`digits`	Number of digits to round results to.
`warn`	If `FALSE`, no warning will be printed to inform you when observations with zero neighbors or `NA` values have been dropped.
`na.rm`	If `na.rm = TRUE`, observations with `NA` values will be dropped from both `x` and `w`.

Details

The formula for the Moran coefficient (MC) is

$$MC = \frac{n}{K} \frac{\sum_i \sum_j w_{ij}(y_i - \overline{y})(y_j - \overline{y})}{\sum_i (y_i - \overline{y})^2}$$

where n is the number of observations and K is the sum of all values in the spatial connectivity matrix W, i.e., the sum of all row-sums: $K = \sum_i \sum_j w_{ij}$.

If any observations with no neighbors are found (i.e. `any(Matrix::rowSums(w) == 0)`) they will be dropped automatically and a message will print stating how many were dropped. The alternative is for those observations to have a spatial lage of zero—but zero is not a neutral value, see the Moran scatter plot.

Value

The Moran coefficient, a numeric value.

Source

Chun, Yongwan, and Daniel A. Griffith. Spatial Statistics and Geostatistics: Theory and Applications for Geographic Information Science and Technology. Sage, 2013.

Cliff, Andrew David, and J. Keith Ord. Spatial processes: models & applications. Taylor & Francis, 1981.

See Also

moran_plot, lisa, aple, gr, lg

Examples

```
library(sf)
data(georgia)
w <- shape2mat(georgia, style = "W")
x <- georgia$ICE
mc(x, w)
```

---

moran_plot   *Moran scatter plot*

Description

Plots a set of values against their spatially lagged values and gives the Moran coefficient as a measure of spatial autocorrelation.

Usage

```
moran_plot(
  x,
  w,
  xlab = "x (centered)",
  ylab = "Spatial Lag",
  pch = 20,
  col = "darkred",
  size = 2,
  alpha = 1,
  lwd = 0.5,
  na.rm = FALSE
)
```

Arguments

`x`	A numeric vector of length n.
`w`	An n x n spatial connectivity matrix.
`xlab`	Label for the x-axis.
`ylab`	Label for the y-axis.
`pch`	Symbol type.
`col`	Symbol color.
`size`	Symbol size.
`alpha`	Symbol transparency.
`lwd`	Width of the regression line.
`na.rm`	If `na.rm = TRUE`, any observations of `x` with `NA` values will be dropped from `x` and from `w`.

Details

For details on the symbol parameters see the documentation for geom_point.

If any observations with no neighbors are found (i.e. `any(Matrix::rowSums(w) == 0)`) they will be dropped automatically and a message will print stating how many were dropped.

Value

Returns a gg plot, a scatter plot with `x` on the horizontal and its spatially lagged values on the vertical axis (i.e. a Moran scatter plot).

Source

Anselin, Luc. "Local indicators of spatial association—LISA." Geographical analysis 27, no. 2 (1995): 93-115.

See Also

mc, lisa, aple

Examples

```
data(georgia)
x <- georgia$income
w <- shape2mat(georgia, "W")
moran_plot(x, w)
```

posterior_predict *Draw samples from the posterior predictive distribution*

Description

Draw samples from the posterior predictive distribution of a fitted `geostan` model.

Usage

```
posterior_predict(object, S, summary = FALSE, width = 0.95, car_parts, seed)
```

Arguments

`object`	A `geostan_fit` object.
`S`	Optional; number of samples to take from the posterior distribution. The default, and maximum, is the total number of samples stored in the model.
`summary`	Should the predictive distribution be summarized by its means and central quantile intervals? If `summary = FALSE`, an `S` x `N` matrix of samples will be returned. If `summary = TRUE`, then a `data.frame` with the means and `100*width` credible intervals is returned.
`width`	Only used if `summary = TRUE`, to set the quantiles for the credible intervals. Defaults to `width = 0.95`.
`car_parts`	Data for CAR model specification; only required for `stan_car` with `family = auto_gaussian()`.
`seed`	A single integer value to be used in a call to `set.seed` before taking samples from the posterior distribution.

Value

A matrix of size S x N containing samples from the posterior predictive distribution, where S is the number of samples drawn and N is the number of observations. If `summary = TRUE`, a `data.frame` with N rows and 3 columns is returned (with column names `mu`, `lwr`, and `upr`).

Examples

```
fit <- stan_glm(sents ~ offset(log(expected_sents)),
                 re = ~ name,
                 data = sentencing,
                 family = poisson(),
                 chains = 2, iter = 600) # for speed only

yrep <- posterior_predict(fit, S = 65)
plot(density(yrep[1,]))
for (i in 2:nrow(yrep)) lines(density(yrep[i,]), col = 'gray30')
lines(density(sentencing$sents), col = 'darkred', lwd = 2)
```

sentencing *Florida state prison sentencing counts by county, 1905-1910*

Description

Simple features (`sf`) with historic (1910) county boundaries of Florida with aggregated state prison sentencing counts and census data. Sentencing and population counts are aggregates over the period 1905-1910, where populations were interpolated linearly between decennial censuses of 1900 and 1910.

Usage

```
sentencing
```

Format

Simple features (`sf`)/`data.frame` with the following attributes:

name County name

wpop White population total for years 1905-1910

bpop Black population total for years 1905-1910

sents Number of state prison sentences, 1905-1910

plantation_belt Binary indicator for inclusion in the plantation belt

pct_ag_1910 Percent of land area in agriculture, 1910

expected_sents Expected sentences given demographic information and state level sentencing rates by race

sir_raw Standardized incident ratio (observed/expected sentences)

Source

Donegan, Connor. "The Making of Florida's 'Criminal Class': Race, Modernity and the Convict Leasing Program." Florida Historical Quarterly 97.4 (2019): 408-434. `https://osf.io/2wj7s/`.

Mullen, Lincoln A. and Bratt, Jordon. "USABoundaries: Historical and Contemporary Boundaries of the United States of America," Journal of Open Source Software 3, no. 23 (2018): 314, doi:10.21105/joss.00314.

Examples

```
data(sentencing)
print(sentencing)
```

shape2mat *Create spatial and space-time connectivity matrices*

Description

Creates sparse matrix representations of spatial connectivity structures

Usage

```
shape2mat(
  shape,
  style = c("B", "W"),
  queen,
  method = c("queen", "rook", "knn"),
  k = 1,
  longlat = NULL,
  snap = sqrt(.Machine$double.eps),
  t = 1,
  st.style = c("contemp", "lag"),
  quiet = FALSE
)
```

Arguments

`shape`	An object of class `sf`, `SpatialPolygons` or `SpatialPolygonsDataFrame`.
`style`	What kind of coding scheme should be used to create the spatial connectivity matrix? Defaults to "B" for binary; use "W" for row-standardized weights.
`queen`	Deprecated: use the 'method' argument instead. This option is passed to `poly2nb` to set the contiguity condition. Defaults to `TRUE` so that a single shared boundary point (rather than a shared border/line) between polygons is sufficient for them to be considered neighbors.
`method`	Method for determining neighbors: queen, rook, or k-nearest neighbors. See Details for more information.
`k`	Number of neighbors to select for k-nearest neighbor method. Passed to `spdep::knearneigh`.
`longlat`	If longlat = TRUE, Great Circle (rather than Euclidean) distances are used; great circle circle distances account for curvature of the Earth.
`snap`	Passed to `spdep::poly2nb`; "boundary points less than 'snap' distance apart are considered to indicate contiguity."
`t`	Number of time periods. Only the binary coding scheme is available for space-time connectivity matrices.
`st.style`	For space-time data, what type of space-time connectivity structure should be used? Options are "lag" for the lagged specification and "contemp" (the default) for contemporaneous specification (see Details).
`quiet`	If `TRUE`, messages will be silenced.

Details

The method argument currently has three options. The queen contiguity condition defines neighbors as polygons that share at least one point with one another. The rook condition requires that they share a line or border with one another. K-nearest neighbors is based on distance between centroids. All methods are implemented using the spdep package and then converted to sparse matrix format.

Haining and Li (Ch. 4) provide a helpful discussion of spatial connectivity matrices (Ch. 4).

The space-time connectivity matrix can be used for eigenvector space-time filtering (`stan_esf`. The `lagged' space-time structure connects each observation to its own past (one period lagged) value and the` temporaneous' specification links each observation to its neighbors and to its own in situ past (one period lagged) value (Griffith 2012, p. 23).

Value

A spatial connectivity matrix in sparse matrix format. Binary matrices are of class `ngCMatrix`, row-standardized are of class `dgCMatrix`, created by `sparseMatrix`.

Source

Bivand, Roger S. and Pebesma, Edzer and Gomez-Rubio, Virgilio (2013). Applied spatial data analysis with R, Second edition. Springer, NY. https://asdar-book.org/

Griffith, Daniel A. (2012). Space, time, and space-time eigenvector filter specifications that account for autocorrelation. Estadística Espanola, 54(177), 7-34.

Haining, Robert P. and Li, Guangquan (2020). Modelling Spatial and Spatial-Temporal Data: A Bayesian Approach. CRC Press.

See Also

`edges` `row_standardize` `n_nbs`

Examples

```
data(georgia)

## binary adjacency matrix
C <- shape2mat(georgia, "B", method = 'rook')

## number of neighbors per observation
summary( n_nbs(C) )
head(Matrix::summary(C))

## row-standardized matrix
W <- shape2mat(georgia, "W", method = 'rook')

## summary of weights
E <- edges(W, unique_pairs_only = FALSE)
summary(E$weight)

## space-time matricies
## for eigenvector space-time filtering
```

```
## if you have multiple years with same geometry/geography,
## provide the geometry (for a single year!) and number of years \code{t}
Cst <- shape2mat(georgia, t = 5)
dim(Cst)
EVst <- make_EV(Cst)
dim(EVst)
```

---

stan_esf *Spatial filtering*

Description

Fit a spatial regression model using eigenvector spatial filtering (ESF).

Usage

```
stan_esf(
  formula,
  slx,
  re,
  data,
  C,
  EV = make_EV(C, nsa = nsa, threshold = threshold),
  nsa = FALSE,
  threshold = 0.25,
  family = gaussian(),
  prior = NULL,
  ME = NULL,
  centerx = FALSE,
  censor_point,
  prior_only = FALSE,
  chains = 4,
  iter = 2000,
  refresh = 500,
  keep_all = FALSE,
  slim = FALSE,
  drop = NULL,
  pars = NULL,
  control = NULL,
  quiet = FALSE,
  ...
)
```

Arguments

`formula` A model formula, following the R `formula` syntax. Binomial models are specified by setting the left hand side of the equation to a data frame of successes and failures, as in `cbind(successes, failures) ~ x`.

slx
: Formula to specify any spatially-lagged covariates. As in, ~ x1 + x2 (the intercept term will be removed internally). When setting priors for beta, remember to include priors for any SLX terms.

re
: To include a varying intercept (or "random effects") term, alpha_re, specify the grouping variable here using formula syntax, as in ~ ID. Then, alpha_re is a vector of parameters added to the linear predictor of the model, and:

```
alpha_re ~ N(0, alpha_tau)
alpha_tau ~ Student_t(d.f., location, scale).
```

data
: A data.frame or an object coercible to a data frame by as.data.frame containing the model data.

C
: Spatial connectivity matrix which will be used to calculate eigenvectors, if EV is not provided by the user. Typically, the binary connectivity matrix is best for calculating eigenvectors (i.e., using C = shape2mat(shape, style = "B")). This matrix will also be used to calculate residual spatial autocorrelation and any user specified slx terms; it will be row-standardized before calculating slx terms. See shape2mat.

EV
: A matrix of eigenvectors from any (transformed) connectivity matrix, presumably spatial (see make_EV). If EV is provided, still also provide a spatial weights matrix C for other purposes; threshold and nsa are ignored for user provided EV.

nsa
: Include eigenvectors representing negative spatial autocorrelation? Defaults to nsa = FALSE. This is ignored if EV is provided.

threshold
: Eigenvectors with standardized Moran coefficient values below this threshold value will be excluded from the candidate set of eigenvectors, EV. This defaults to threshold = 0.25, and is ignored if EV is provided.

family
: The likelihood function for the outcome variable. Current options are family = gaussian(), student_t() and poisson(link = "log"), and binomial(link = "logit").

prior
: A named list of parameters for prior distributions (see priors):

intercept The intercept is assigned a Gaussian prior distribution (see normal.

beta Regression coefficients are assigned Gaussian prior distributions. Variables must follow their order of appearance in the model formula. Note that if you also use slx terms (spatially lagged covariates), and you use custom priors for beta, then you have to provide priors for the slx terms. Since slx terms are *prepended* to the design matrix, the prior for the slx term will be listed first.

sigma For family = gaussian() and family = student_t() models, the scale parameter, sigma, is assigned a (half-) Student's t prior distribution. The half-Student's t prior for sigma is constrained to be positive.

nu nu is the degrees of freedom parameter in the Student's t likelihood (only used when family = student_t()). nu is assigned a gamma prior distribution. The default prior is prior = list(nu = gamma2(alpha = 3, beta = 0.2)).

tau The scale parameter for random effects, or varying intercepts, terms. This scale parameter, `tau`, is assigned a half-Student's t prior. To set this, use, e.g., `prior = list(tau = student_t(df = 20, location = 0, scale = 20))`.

beta_ev The eigenvector coefficients are assigned the horseshoe prior (Piironen and Vehtari, 2017), parameterized by `global_scale` (to control overall prior sparsity), plus the degrees of freedom and scale of a Student's t model for any large coefficients (see `priors`). To allow the spatial filter to account for a greater amount of spatial autocorrelation (i.e., if you find the residuals contain spatial autocorrelation), increase the global scale parameter (to a maximum of `global_scale = 1`).

`ME` To model observational uncertainty (i.e. measurement or sampling error) in any or all of the covariates, provide a list of data as constructed by the `prep_me_data` function.

`centerx` To center predictors on their mean values, use `centerx = TRUE`. If the ME argument is used, the modeled covariate (i.e., latent variable), rather than the raw observations, will be centered. When using the ME argument, this is the recommended method for centering the covariates.

`censor_point` Integer value indicating the maximum censored value; this argument is for modeling censored (suppressed) outcome data, typically disease case counts or deaths. For example, the US Centers for Disease Control and Prevention censors (does not report) death counts that are nine or fewer, so if you're using CDC WONDER mortality data you could provide `censor_point = 9`.

`prior_only` Draw samples from the prior distributions of parameters only.

`chains` Number of MCMC chains to estimate. Default `chains = 4`.

`iter` Number of samples per chain. Default `iter = 2000`.

`refresh` Stan will print the progress of the sampler every `refresh` number of samples. Defaults to `500`; set `refresh=0` to silence this.

`keep_all` If `keep_all = TRUE` then samples for all parameters in the Stan model will be kept; this is necessary if you want to do model comparison with Bayes factors and the `bridgesampling` package.

`slim` If `slim = TRUE`, then the Stan model will not collect the most memory-intensive parameters (including n-length vectors of fitted values, log-likelihoods, and ME-modeled covariate values). This will disable many convenience functions that are otherwise available for fitted `geostan` models, such as the extraction of residuals, fitted values, and spatial trends, WAIC, and spatial diagnostics, and ME diagnostics; many quantities of interest, such as fitted values and spatial trends, can still be calculated manually using given parameter estimates. The "slim" option is useful for data-intensive routines, such as regression with raster data, Monte Carlo studies, and measurement error models. For more control over which parameters are kept or dropped, use the `drop` argument instead of `slim`.

`drop` Provide a vector of character strings to specify the names of any parameters that you do not want MCMC samples for. Dropping parameters in this way can improve sampling speed and reduce memory usage. The following parameter vectors can potentially be dropped from ESF models:

fitted The N-length vector of fitted values

log_lik The N-length vector of pointwise log-likelihoods, which is used to calculate WAIC.

alpha_re Vector of 'random effects'/varying intercepts.

x_true N-length vector of 'latent'/modeled covariate values created for measurement error (ME) models.

esf The N-length eigenvector spatial filter.

beta_ev The vector of coefficients for the eigenvectors.

If `slim = TRUE`, then `drop` will be ignored.

`pars` Optional; specify any additional parameters you'd like stored from the Stan model.

`control` A named list of parameters to control the sampler's behavior. See stan for details.

`quiet` By default, any prior distributions that have not been assigned by the user are printed to the console. If `quiet = TRUE`, these will not be printed.

`...` Other arguments passed to sampling.

Details

Eigenvector spatial filtering (ESF) is a method for spatial regression analysis. ESF is extensively covered in Griffith et al. (2019). This function implements the methodology introduced in Donegan et al. (2020), which uses Piironen and Vehtari's (2017) regularized horseshoe prior.

ESF decomposes spatial autocorrelation into a linear combination of various patterns, typically at different scales (such as local, regional, and global trends). By adding a spatial filter to a regression model, these spatial autocorrelation patterns are shifted from the residuals to the spatial filter. ESF models take the spectral decomposition of a transformed spatial connectivity matrix, C. The resulting eigenvectors, E, are mutually orthogonal and uncorrelated map patterns. The spatial filter equals $E\beta_E$ where β_E is a vector of coefficients.

ESF decomposes the data into a global mean, α, global patterns contributed by covariates $X\beta$, spatial trends $E\beta_E$, and residual variation. Thus, for `family=gaussian()`,

$$y \sim Gauss(\alpha + X * \beta + E\beta_E, \sigma).$$

An ESF component can be incorporated into the linear predictor of any generalized linear model. For example, a spatial Poisson model for rare disease incidence may be specified as follows:

$$y \sim Poisson(e^{O+\mu})$$
$$\mu = \alpha + E\beta_E + A$$
$$A \sim Guass(0, \tau)$$
$$\tau \sim student(20, 0, 2)$$
$$\beta_E \sim horseshoe(.)$$

The form of this model is similar to the BYM model (see stan_icar), in the sense that it contains a spatially structured trend term ($E\beta_E$) and an unstructured ('random effects') term (A).

The spatial.geostan_fit method will return $E\beta_E$.

The model can also be extended to the space-time domain; see shape2mat to specify a space-time connectivity matrix.

The coefficients β_E are assigned the regularized horseshoe prior (Piironen and Vehtari, 2017), resulting in a relatively sparse model specification. In addition, numerous eigenvectors are automatically dropped because they represent trace amounts of spatial autocorrelation (this is controlled by the `threshold` argument). By default, `stan_esf` will drop all eigenvectors representing negative spatial autocorrelation patterns. You can change this behavior using the `nsa` argument.

Additional functionality:

The CAR models can also incorporate spatially-lagged covariates, measurement/sampling error in covariates (particularly when using small area survey estimates as covariates), missing outcome data, and censored outcomes (such as arise when a disease surveillance system suppresses data for privacy reasons). For details on these options, please see the Details section in the documentation for stan_glm.

Value

An object of class class `geostan_fit` (a list) containing:

summary Summaries of the main parameters of interest; a data frame

diagnostic Widely Applicable Information Criteria (WAIC) with a measure of effective number of parameters (`eff_pars`) and mean log pointwise predictive density (`lpd`), and mean residual spatial autocorrelation as measured by the Moran coefficient.

data a data frame containing the model data

EV A matrix of eigenvectors created with `w` and `geostan::make_EV`

C The spatial weights matrix used to construct EV

family the user-provided or default `family` argument used to fit the model

formula The model formula provided by the user (not including ESF component)

slx The `slx` formula

re A list containing `re`, the random effects (varying intercepts) formula if provided, and `data` a data frame with columns `id`, the grouping variable, and `idx`, the index values assigned to each group.

priors Prior specifications.

x_center If covariates are centered internally (`centerx = TRUE`), then `x_center` is a numeric vector of the values on which covariates were centered.

ME The `ME` data list, if one was provided by the user for measurement error models.

spatial A data frame with the name of the spatial component parameter ("esf") and method ("ESF")

stanfit an object of class `stanfit` returned by `rstan::stan`

Author(s)

Connor Donegan, <connor.donegan@gmail.com>

Source

Chun, Y., D. A. Griffith, M. Lee and P. Sinha (2016). Eigenvector selection with stepwise regression techniques to construct eigenvector spatial filters. *Journal of Geographical Systems*, 18(1), 67-85. doi:10.1007/s1010901502253.

Dray, S., P. Legendre & P. R. Peres-Neto (2006). Spatial modelling: a comprehensive framework for principal coordinate analysis of neighbour matrices (PCNM). *Ecological Modeling*, 196(3-4), 483-493.

Donegan, C., Y. Chun and A. E. Hughes (2020). Bayesian estimation of spatial filters with Moran's Eigenvectors and hierarchical shrinkage priors. *Spatial Statistics*. doi:10.1016/j.spasta.2020.100450 (open access: doi:10.31219/osf.io/fah3z).

Donegan, Connor (2021). Building spatial conditional autoregressive (CAR) models in the Stan programming language. *OSF Preprints*. doi:10.31219/osf.io/3ey65.

Griffith, Daniel A., and P. R. Peres-Neto (2006). Spatial modeling in ecology: the flexibility of eigenfunction spatial analyses. *Ecology* 87(10), 2603-2613.

Griffith, D., and Y. Chun (2014). Spatial autocorrelation and spatial filtering, Handbook of Regional Science. Fischer, MM and Nijkamp, P. eds.

Griffith, D., Chun, Y. and Li, B. (2019). *Spatial Regression Analysis Using Eigenvector Spatial Filtering*. Elsevier.

Piironen, J and A. Vehtari (2017). Sparsity information and regularization in the horseshoe and other shrinkage priors. In *Electronic Journal of Statistics*, 11(2):5018-5051.

Examples

```
data(sentencing)
# spatial weights matrix with binary coding scheme
C <- shape2mat(sentencing, style = "B")

# log-expected number of sentences
## expected counts are based on county racial composition and mean sentencing rates
log_e <- log(sentencing$expected_sents)

# fit spatial Poisson model with ESF + unstructured 'random effects'
fit.esf <- stan_esf(sents ~ offset(log_e),
                   re = ~ name,
                   family = poisson(),
                   data = sentencing,
                   C = C,
                   chains = 2, iter = 800) # for speed only

# spatial diagnostics
sp_diag(fit.esf, sentencing)
plot(fit.esf)

# plot marginal posterior distributions of beta_ev (eigenvector coefficients)
plot(fit.esf, pars = "beta_ev")

# plot the marginal posterior distributions of the spatial filter
plot(fit.esf, pars = "esf")
```

```
# calculate log-standardized incidence ratios
library(ggplot2)
library(sf)
f <- fitted(fit.esf, rates = FALSE)$mean
SSR <-  f / sentencing$expected_sents
log.SSR <- log( SSR, base = 2 )

# map the log-SSRs
 ggplot(sentencing) +
   geom_sf(aes(fill = log.SSR)) +
   scale_fill_gradient2(
    midpoint = 0,
    name = NULL,
    breaks = seq(-3, 3, by = 0.5)
  ) +
   labs(title = "Log-Standardized Sentencing Ratios",
    subtitle = "log( Fitted/Expected ), base 2"
 ) +
   theme_void()
```

`stan_icar` *Intrinsic autoregressive models*

Description

The intrinsic conditional auto-regressive (ICAR) model for spatial count data. Options include the BYM model, the BYM2 model, and a solo ICAR term.

Usage

```
stan_icar(
  formula,
  slx,
  re,
  data,
  C,
  family = poisson(),
  type = c("icar", "bym", "bym2"),
  scale_factor = NULL,
  prior = NULL,
  ME = NULL,
  centerx = FALSE,
  censor_point,
  prior_only = FALSE,
  chains = 4,
  iter = 2000,
  refresh = 500,
```

```
  keep_all = FALSE,
  slim = FALSE,
  drop = NULL,
  pars = NULL,
  control = NULL,
  quiet = FALSE,
  ...
)
```

Arguments

`formula` A model formula, following the R formula syntax. Binomial models can be specified by setting the left hand side of the equation to a data frame of successes and failures, as in `cbind(successes, failures) ~ x`.

`slx` Formula to specify any spatially-lagged covariates. As in, `~ x1 + x2` (the intercept term will be removed internally). When setting priors for `beta`, remember to include priors for any SLX terms.

`re` To include a varying intercept (or "random effects") term, `alpha_re`, specify the grouping variable here using formula syntax, as in `~ ID`. Then, `alpha_re` is a vector of parameters added to the linear predictor of the model, and:

```
alpha_re ~ N(0, alpha_tau)
alpha_tau ~ Student_t(d.f., location, scale).
```

Before using this term, read the `Details` section and the `type` argument. Specifically, if you use `type = bym`, then an observational-level `re` term is already included in the model. (Similar for `type = bym2`.)

`data` A `data.frame` or an object coercible to a data frame by `as.data.frame` containing the model data.

`C` Spatial connectivity matrix which will be used to construct an edge list for the ICAR model, and to calculate residual spatial autocorrelation as well as any user specified `slx` terms. It will automatically be row-standardized before calculating `slx` terms. `C` must be a binary symmetric `n x n` matrix.

`family` The likelihood function for the outcome variable. Current options are `binomial(link = "logit")` and `poisson(link = "log")`.

`type` Defaults to "icar" (partial pooling of neighboring observations through parameter `phi`); specify "bym" to add a second parameter vector `theta` to perform partial pooling across all observations; specify "bym2" for the innovation introduced by Riebler et al. (2016). See `Details` for more information.

`scale_factor` For the BYM2 model, optional. If missing, this will be set to a vector of ones. See `Details`.

`prior` A named list of parameters for prior distributions (see `priors`):

intercept The intercept is assigned a Gaussian prior distribution (see `normal`.

beta Regression coefficients are assigned Gaussian prior distributions. Variables must follow their order of appearance in the model `formula`. Note that if you also use `slx` terms (spatially lagged covariates), and you use custom priors for `beta`, then you have to provide priors for the slx terms.

Since slx terms are *prepended* to the design matrix, the prior for the slx term will be listed first.

sigma For `family = gaussian()` and `family = student_t()` models, the scale parameter, `sigma`, is assigned a (half-) Student's t prior distribution. The half-Student's t prior for `sigma` is constrained to be positive.

nu `nu` is the degrees of freedom parameter in the Student's t likelihood (only used when `family = student_t()`). `nu` is assigned a gamma prior distribution. The default prior is `prior = list(nu = gamma2(alpha = 3, beta = 0.2))`.

tau The scale parameter for random effects, or varying intercepts, terms. This scale parameter, `tau`, is assigned a half-Student's t prior. To set this, use, e.g., `prior = list(tau = student_t(df = 20, location = 0, scale = 20))`.

`ME` To model observational uncertainty (i.e. measurement or sampling error) in any or all of the covariates, provide a list of data as constructed by the `prep_me_data` function.

`centerx` To center predictors on their mean values, use `centerx = TRUE`. If the ME argument is used, the modeled covariate (i.e., latent variable), rather than the raw observations, will be centered. When using the ME argument, this is the recommended method for centering the covariates.

`censor_point` Integer value indicating the maximum censored value; this argument is for modeling censored (suppressed) outcome data, typically disease case counts or deaths. For example, the US Centers for Disease Control and Prevention censors (does not report) death counts that are nine or fewer, so if you're using CDC WONDER mortality data you could provide `censor_point = 9`.

`prior_only` Draw samples from the prior distributions of parameters only.

`chains` Number of MCMC chains to estimate.

`iter` Number of samples per chain. .

`refresh` Stan will print the progress of the sampler every `refresh` number of samples; set `refresh=0` to silence this.

`keep_all` If `keep_all = TRUE` then samples for all parameters in the Stan model will be kept; this is necessary if you want to do model comparison with Bayes factors and the `bridgesampling` package.

`slim` If `slim = TRUE`, then the Stan model will not collect the most memory-intensive parameters (including n-length vectors of fitted values, log-likelihoods, and ME-modeled covariate values). This will disable many convenience functions that are otherwise available for fitted `geostan` models, such as the extraction of residuals, fitted values, and spatial trends, WAIC, and spatial diagnostics, and ME diagnostics; many quantities of interest, such as fitted values and spatial trends, can still be calculated manually using given parameter estimates. The "slim" option is designed for data-intensive routines, such as regression with raster data, Monte Carlo studies, and measurement error models. For more control over which parameters are kept or dropped, use the `drop` argument instead of `slim`.

`drop` Provide a vector of character strings to specify the names of any parameters that you do not want MCMC samples for. Dropping parameters in this way can improve sampling speed and reduce memory usage. The following parameter vectors can potentially be dropped from ICAR models:

fitted The N-length vector of fitted values

log_lik The N-length vector of pointwise log-likelihoods, which is used to calculate WAIC.

alpha_re Vector of 'random effects'/varying intercepts.

x_true N-length vector of 'latent'/modeled covariate values created for measurement error (ME) models.

phi The N-length vector of spatially-autocorrelated parameters (with the ICAR prior).

theta The N-length vector of spatially unstructured parameters ('random effects'), for the BYM and BYM2 models.

If `slim = TRUE`, then `drop` will be ignored.

`pars` Optional; specify any additional parameters you'd like stored from the Stan model.

`control` A named list of parameters to control the sampler's behavior. See `stan` for details.

`quiet` Controls (most) automatic printing to the console. By default, any prior distributions that have not been assigned by the user are printed to the console. If `quiet = TRUE`, these will not be printed. Using `quiet = TRUE` will also force `refresh = 0`.

`...` Other arguments passed to sampling.

Details

The intrinsic conditional autoregressive (ICAR) model for spatial data was introduced by Besag et al. (1991). The Stan code for the ICAR component of the model and the BYM2 option is from Morris et al. (2019) with adjustments to enable non-binary weights and disconnected graph structures (see Freni-Sterrantino (2018) and Donegan (2021)).

The exact specification depends on the `type` argument.

ICAR:

For Poisson models for count data, y, the basic model specification (`type = "icar"`) is:

$$y\ Poisson(e^{O+\mu+\phi})$$
$$\phi \sim ICAR(\tau_s)$$
$$\tau_s \sim Gauss(0, 1)$$

where μ contains an intercept and potentially covariates. The spatial trend phi has a mean of zero and a single scale parameter τ_s (which user's will see printed as the parameter named `spatial_scale`).

The ICAR prior model is a CAR model that has a spatial autocorrelation parameter ρ equal to 1 (see stan_car). Thus the ICAR prior places high probability on a very smooth spatially (or temporally) varying mean. This is rarely sufficient to model the amount of variation present in social and health data. For this reason, the BYM model is typically employed.

BYM:

Often, an observational-level random effect term, `theta`, is added to capture (heterogeneous or unstructured) deviations from $\mu + \phi$. The combined term is referred to as a convolution term:
$convolution = \phi + \theta$.

This is known as the BYM model (Besag et al. 1991), and can be specified using `type = "bym"`:
$y \sim Poisson(e^{O+\mu+\phi+\theta})$

$$\phi \sim ICAR(\tau_s)$$
$$\theta \sim Gaussian(0, \tau_{ns})$$
$$\tau_s \sim Gaussian(0, 1)$$
$$\tau_{ns} \sim Gaussian(0, 1)$$

The model is named after Besag, York, and Mollié (1991).

BYM2:

Riebler et al. (2016) introduce a variation on the BYM model (`type = "bym2"`). This specification combines ϕ and θ using a mixing parameter ρ that controls the proportion of the variation that is attributable to the spatially autocorrelated term ϕ rather than the spatially unstructured term θ. The terms share a single scale parameter τ:

$$convolution = [sqrt(\rho * S) * \tilde{\phi} + sqrt(1 - \rho)\tilde{\theta}] * \tau$$
$$\tilde{\phi} \sim Gaussian(0, 1)$$
$$\tilde{\theta} \sim Gaussian(0, 1)$$
$$\tau \sim Gaussian(0, 1)$$

The terms $\tilde{\phi}$, $\tilde{\theta}$ are standard normal deviates, ρ is restricted to values between zero and one, and S is the 'scale_factor' (a constant term provided by the user). By default, the 'scale_factor' is equal to one, so that it does nothing. Riebler et al. (2016) argue that the interpretation or meaning of the scale of the ICAR model depends on the graph structure of the connectivity matrix C. This implies that the same prior distribution assigned to τ_s will differ in its implications if C is changed; in other words, the priors are not transportable across models, and models that use the same nominal prior actually have different priors assigned to τ_s.

Borrowing `R` code from Morris (2017) and following Freni-Sterrantino et al. (2018), the following `R` code can be used to create the 'scale_factor' S for the BYM2 model (note, this requires the INLA R package), given a spatial adjacency matrix, C:

```
## create a list of data for stan_icar
icar.data <- geostan::prep_icar_data(C)
## calculate scale_factor for each of k connected group of nodes
k <- icar.data$k
scale_factor <- vector(mode = "numeric", length = k)
for (j in 1:k) {
  g.idx <- which(icar.data$comp_id == j)
  if (length(g.idx) == 1) {
       scale_factor[j] <- 1
       next
    }
```

```
  Cg <- C[g.idx, g.idx]
  scale_factor[j] <- scale_c(Cg)
}
```

This code adjusts for 'islands' or areas with zero neighbors, and it also handles disconnected graph structures (see Donegan and Morris 2021). Following Freni-Sterrantino (2018), disconnected components of the graph structure are given their own intercept term; however, this value is added to ϕ automatically inside the Stan model. Therefore, the user never needs to make any adjustments for this term. (To avoid complications from using a disconnected graph structure, you can apply a proper CAR model instead of the ICAR: `stan_car`).

Note, the code above requires the `scale_c` function; it has package dependencies that are not included in `geostan`. To use `scale_c`, you have to load the following `R` function:

```
#' compute scaling factor for adjacency matrix, accounting for differences in spatial connectivity
#'
#' @param C connectivity matrix
#'
#' @details
#'
#' Requires the following packages:
#'
#' library(Matrix)
#' library(INLA);
#' library(spdep)
#' library(igraph)
#'
#' @source  Morris (2017)
#'
scale_c <- function(C) {
 geometric_mean <- function(x) exp(mean(log(x)))
 N = dim(C)[1]
 Q =  Diagonal(N, rowSums(C)) - C
 Q_pert = Q + Diagonal(N) * max(diag(Q)) * sqrt(.Machine$double.eps)
 Q_inv = inla.qinv(Q_pert, constr=list(A = matrix(1,1,N),e=0))
 scaling_factor <- geometric_mean(Matrix::diag(Q_inv))
 return(scaling_factor)
}
```

Additional functionality:

The CAR models can also incorporate spatially-lagged covariates, measurement/sampling error in covariates (particularly when using small area survey estimates as covariates), missing outcome data, and censored outcomes (such as arise when a disease surveillance system suppresses data for privacy reasons). For details on these options, please see the Details section in the documentation for stan_glm.

Value

An object of class class `geostan_fit` (a list) containing:

summary Summaries of the main parameters of interest; a data frame

diagnostic Widely Applicable Information Criteria (WAIC) with a measure of effective number of parameters (`eff_pars`) and mean log pointwise predictive density (`lpd`), and mean residual spatial autocorrelation as measured by the Moran coefficient.

stanfit an object of class `stanfit` returned by `rstan::stan`

data a data frame containing the model data

edges The edge list representing all unique sets of neighbors and the weight attached to each pair (i.e., their corresponding element in the connectivity matrix C

C Spatial connectivity matrix

family the user-provided or default `family` argument used to fit the model

formula The model formula provided by the user (not including ICAR component)

slx The `slx` formula

re A list with two name elements, `formula` and `Data`, containing the formula `re` and a data frame with columns `id` (the grouping variable) and `idx` (the index values assigned to each group).

priors Prior specifications.

x_center If covariates are centered internally (`centerx = TRUE`), then `x_center` is a numeric vector of the values on which covariates were centered.

spatial A data frame with the name of the spatial parameter (`"phi"` if `type = "icar"` else `"convolution"`) and method (`toupper(type)`).

Author(s)

Connor Donegan, <connor.donegan@gmail.com>

Source

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2), 192-225.

Besag, J., York, J., and Mollié, A. (1991). Bayesian image restoration, with two applications in spatial statistics. *Annals of the Institute of Statistical Mathematics*, 43(1), 1-20.

Donegan, Connor and Morris, Mitzi (2021). Flexible functions for ICAR, BYM, and BYM2 models in Stan. Code repository. https://github.com/ConnorDonegan/Stan-IAR

Donegan, Connor (2021b). Building spatial conditional autoregressive (CAR) models in the Stan programming language. *OSF Preprints*. doi:10.31219/osf.io/3ey65.

Donegan, Connor and Chun, Yongwan and Griffith, Daniel A. (2021). Modeling community health with areal data: Bayesian inference with survey standard errors and spatial structure. *Int. J. Env. Res. and Public Health* 18 (13): 6856. DOI: 10.3390/ijerph18136856 Data and code: https://github.com/ConnorDonegan/survey-HBM.

Freni-Sterrantino, Anna, Massimo Ventrucci, and Håvard Rue (2018). A Note on Intrinsic Conditional Autoregressive Models for Disconnected Graphs. *Spatial and Spatio-Temporal Epidemiology*, 26: 25–34.

Morris, Mitzi (2017). Spatial Models in Stan: Intrinsic Auto-Regressive Models for Areal Data. https://mc-stan.org/users/documentation/case-studies/icar_stan.html

Morris, M., Wheeler-Martin, K., Simpson, D., Mooney, S. J., Gelman, A., & DiMaggio, C. (2019). Bayesian hierarchical spatial models: Implementing the Besag York Mollié model in stan. *Spatial and spatio-temporal epidemiology*, 31, 100301.

Riebler, A., Sorbye, S. H., Simpson, D., & Rue, H. (2016). An intuitive Bayesian spatial model for disease mapping that accounts for scaling. *Statistical Methods in Medical Research*, 25(4), 1145-1165.

See Also

shape2mat, stan_car, stan_esf, stan_glm, prep_icar_data

Examples

```
data(sentencing)
C <- shape2mat(sentencing, "B")
log_e <- log(sentencing$expected_sents)
fit.bym <- stan_icar(sents ~ offset(log_e),
                     family = poisson(),
                     data = sentencing,
                     type = "bym",
                     C = C,
                     chains = 2, iter = 800) # for speed only

# spatial diagnostics
sp_diag(fit.bym, sentencing)

# check effective sample size and convergence
library(rstan)
rstan::stan_ess(fit.bym$stanfit)
rstan::stan_rhat(fit.bym$stanfit)

# calculate log-standardized incidence ratios
# (observed/exected case counts)
library(ggplot2)
library(sf)

f <- fitted(fit.bym, rates = FALSE)$mean
SSR <- f / sentencing$expected_sents
log.SSR <- log( SSR, base = 2)

ggplot(sentencing) +
  geom_sf(aes(fill = log.SSR)) +
  scale_fill_gradient2(
   low = "navy",
   high = "darkred"
  ) +
  labs(title = "Log-standardized sentencing ratios",
       subtitle = "log( Fitted/Expected), base 2") +
  theme_void() +
  theme(
   legend.position = "bottom",
   legend.key.height = unit(0.35, "cm"),
```

```
  legend.key.width = unit(1.5, "cm")
 )
```

waic *Widely Applicable Information Criteria (WAIC)*

Description

Widely Application Information Criteria (WAIC) for model comparison

Usage

```
waic(fit, pointwise = FALSE, digits = 2)
```

Arguments

fit	An `geostan_fit` object or any Stan model with a parameter named "log_lik", the pointwise log likelihood of the observations.
pointwise	Logical (defaults to `FALSE`), should a vector of values for each observation be returned?
digits	Round results to this many digits.

Value

A vector of length 3 with `WAIC`, a rough measure of the effective number of parameters estimated by the model `Eff_pars`, and log predictive density `Lpd`. If `pointwise = TRUE`, results are returned in a `data.frame`.

Source

Watanabe, S. (2010). Asymptotic equivalence of Bayes cross validation and widely application information criterion in singular learning theory. Journal of Machine Learning Research 11, 3571-3594.

Examples

```
data(georgia)
fit <- stan_glm(log(rate.male) ~ 1, data = georgia,
                chains = 2, iter = 800) # for speed only
waic(fit)
```

Index