Package ‘geostan’

May 11, 2024

Title Bayesian Spatial Analysis

Version 0.6.1

Date 2024-05-08

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

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

Description For Bayesian inference with spatial data, provides exploratory spatial analysis tools, multiple spatial model specifications, spatial 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

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

NeedsCompilation yes

Author Connor Donegan [aut, cre] (<https://orcid.org/0000-0002-9698-5443>),
Mitzi Morris [ctb],
Amy Tims [ctb]

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

Repository CRAN

Date/Publication 2024-05-10 22:23:01 UTC

R topics documented:

`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

`aple` *Spatial autocorrelation estimator*

Description

The approximate-profile likelihood estimator for the spatial autocorrelation parameter from a simultaneous autoregressive (SAR) model (Li et al. 2007). Note, the `APLE` approximation is not reliable when the number of observations is large.

Usage

```
aple(x, w, digits = 3)
```

Arguments

`x` Numeric vector of values, length `n`. This will be standardized internally with `scale(x)`.

`w` An `n x n` row-standardized spatial connectivity matrix. See shape2mat.

`digits` Number of digits to round results to.

Details

The `APLE` is an estimate of the spatial autocorrelation parameter one would obtain from fitting an intercept-only SAR model.

Value

the APLE estimate, a numeric value.

Source

Li, Honfei and Calder, Catherine A. and Cressie, Noel (2007). Beyond Moran's I: testing for spatial dependence based on the spatial autoregressive model. Geographical Analysis: 39(4): 357-375.

See Also

mc, moran_plot, lisa, sim_sar

Examples

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

`as.matrix.geostan_fit` *Extract samples from a fitted model*

Description

Extract samples from the joint posterior distribution of parameters.

Usage

```
## S3 method for class 'geostan_fit'
as.matrix(x, ...)

## S3 method for class 'geostan_fit'
as.data.frame(x, ...)

## S3 method for class 'geostan_fit'
as.array(x, ...)
```

Arguments

`x` A fitted model object of class `geostan_fit`.

`...` Further arguments passed to `rstan` methods for for `as.data.frame`, `as.matrix`, or `as.array`

Value

A matrix, data frame, or array of MCMC samples is returned.

Examples

```
data(georgia)
A <- shape2mat(georgia, "B")

fit <- stan_glm(deaths.male ~ offset(log(pop.at.risk.male)),
                C = A,
                data = georgia,
                family = poisson(),
                chains = 1, iter = 600) # for speed only


s <- as.matrix(fit)
dim(s)

a <- as.matrix(fit, pars = "intercept")
dim(a)

# Or extract the stanfit object
S <- fit$stanfit
print(S, pars = "intercept")
samples <- as.matrix(S)
dim(samples)
```

auto_gaussian *Auto-Gaussian family for CAR models*

Description

create a family object for the auto-Gaussian CAR or SAR specification

Usage

```
auto_gaussian(type)
```

Arguments

`type`	Optional; either "CAR" for conditionally specified auto-model or "SAR" for the simultaneously specified auto-model. The type is added internally by `stan_car` or `stan_sar` when needed.

Value

An object of class `family`

See Also

`stan_car`

Examples

```
cp = prep_car_data(shape2mat(georgia))
fit <- stan_car(log(rate.male) ~ 1,
                data = georgia,
                car_parts = cp,
                family = auto_gaussian(),
                chains = 2, iter = 700) # for speed only
print(fit)
```

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)
```

eigen_grid — *Eigenvalues of a spatial weights matrix: for spatial regression with raster data*

Description

Approximate eigenvalues for the row-standardized spatial connectivity matrix W of a regular tessellation, e.g., remotely sensed imagery.

Usage

```
eigen_grid(row = 5, col = 5)
```

Arguments

`row`	Number of rows in the raster dataset
`col`	Number of columns in the raster dataset

Details

Uses Equation 5 from Griffith (2000) to calculate the eigenvalues for a row-standardized spatial weights matrix; this is valid for a regular tessellation (rectangular grid or raster). The rook criteria is used to define adjacency.

The purpose is to calculate eigenvalues of the spatial weights matrix for the CAR and SAR models, enabling spatial regression with large raster data sets. This function is used internally by `prep_sar_data2` and `prep_car_data2`. For more details, see: `vignette("raster-regression", package = "geostan")`.

Source

Griffith, Daniel A. (2000). Eigenfunction properties and approximations of selected incidence matrices employed in spatial analyses. *Linear Algebra and its Applications* 321 (1-3): 95-112. doi:10.1016/S00243795(00)000318.

See Also

`prep_sar_data2`, `prep_car_data2`

Examples

```
e <- eigen_grid(row = 50, col = 95)
print(head(e, 25))
```

`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)
```

`georgia` *Georgia all-cause, sex-specific mortality, ages 55-64, years 2014-2018*

Description

A simple features (sf) object for Georgia counties with sex- and age-specific deaths and populations at risk (2014-2018), plus select estimates (with standard errors) of county characteristics. Standard errors of the ICE were calculated using the Census Bureau's variance replicate tables.

Usage

```
georgia
```

Format

A simple features object with county geometries and the following columns:

GEOID Six digit combined state and county FIPS code

NAME County name

ALAND Land area

AWATER Water area

population Census Bureau 2018 county population estimate

white Percent White, ACS 2018 five-year estimate

black Percent Black, ACS 2018 five-year estimate

hisp Percent Hispanic/Latino, ACS 2018 five-year estimate

ai Percent American Indian, ACS 2018 five-year estimate

deaths.male Male deaths, 55-64 yo, 2014-2018

pop.at.risk.male Population estimate, males, 55-64 yo, years 2014-2018 (total), ACS 2018 five-year estimate

pop.at.risk.male.se Standard error of the pop.at.risk.male estimate

deaths.female Female deaths, 55-64 yo, 2014-2018

pop.at.risk.female Population estimate, females, 55-64 yo, years 2014-2018 (total), ACS 2018 five-year estimate

pop.at.risk.female.se Standard error of the pop.at.risk.female estimate

ICE Index of Concentration at the Extremes

ICE.se Standard error of the ICE estimate, calculated using variance replicate tables

income Median household income, ACS 2018 five-year estimate

income.se Standard error of the income estimate

college Percent of the population age 25 or higher than has a bachelors degree of higher, ACS 2018 five-year estimate

college.se Standard error of the college estimate

insurance Percent of the population with health insurance coverage, ACS 2018 five-year estimate

insurance.se Standard error of the insurance estimate

rate.male Raw (crude) age-specific male mortality rate, 2014-2018

rate.female Raw (crude) age-specific female mortality rate, 2014-2018

geometry simple features geometry for county boundaries

Source

Centers for Disease Control and Prevention, National Center for Health Statistics. Underlying Cause of Death 1999-2018 on CDC Wonder Online Database. 2020. Available online: `http://wonder.cdc.gov` (accessed on 19 October 2020).

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`.

Kyle Walker and Matt Herman (2020). tidycensus: Load US Census Boundary and Attribute Data as 'tidyverse' and 'sf'-Ready Data Frames. R package version 0.11. `https://CRAN.R-project.org/package=tidycensus`

US Census Bureau. Variance Replicate Tables, 2018. Available online: `https://www.census.gov/programs-surveys/acs/data/variance-tables.2018.html` (accessed on 19 October 2020).

Examples

```
data(georgia)
head(georgia)

library(sf)
plot(georgia[,'rate.female'])
```

`get_shp` *Download shapefiles*

Description

Given a url to a shapefile in a compressed .zip file, download the file and unzip it into a folder in your working directory.

Usage

```
get_shp(url, folder = "shape")
```

Arguments

`url`	url to download a shapefile.
`folder`	what to name the new folder in your working directory containing the shapefile

Value

A folder in your working directory with the shapefile; filepaths are printed to the console.

Examples

```
## Not run:
library(sf)
url <- "https://www2.census.gov/geo/tiger/GENZ2019/shp/cb_2019_us_state_20m.zip"
folder <- tempdir()
print(folder)
get_shp(url, folder)
states <- sf::st_read(folder)
head(states)

## End(Not run)
```

gr *The Geary Ratio*

Description

An index for spatial autocorrelation. Complete spatial randomness (lack of spatial pattern) is indicated by a Geary Ratio (GR) of 1; positive autocorrelation moves the index towards zero, while negative autocorrelation will push the index towards 2.

Usage

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

Arguments

`x`	Numeric vector of length `n`. By default, this will be standardized using the `scale` function.
`w`	An `n x n` spatial connectivity matrix. See shape2mat.
`digits`	Number of digits to round results to.
`na.rm`	If `na.rm = TRUE`, observations with `NA` values will be dropped from both `x` and `w`.
`warn`	If `FALSE`, no warning will be printed to inform you when observations with `NA` values have been dropped, or if any observations without neighbors have been found.

Details

The Geary Ratio is an index of spatial autocorrelation. The numerator contains a series of sums of squared deviations, which will be smaller when each observation is similar to its neighbors. This term makes the index sensitive to local outliers, which is advantageous for detecting such outliers and for measuring negative autocorrelation. The denominator contains the total sum of squared deviations from the mean value. Hence, under strong positive autocorrelation, the GR approaches zero. Zero spatial autocorrelation is represented by a GR of 1. Negative autocorrelation pushes the GR above 1, towards 2.

$$GR = \frac{n-1}{2K} \frac{M}{D}$$

$$M = \sum_i \sum_j w_{i,j} (x_i - x_j)^2$$

$$D = \sum_i (x_i - \overline{x})^2$$

Observations with no neighbors are removed before calculating the GR. The alternative is for those observations to contribute zero to the numerator—but zero is not a neutral value, it represents strong positive autocorrelation.

Source

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

Geary, R. C. "The contiguity ratio and statistical mapping." The Incorporated Statistician 5, no. 3 (1954): 115-127_129-146.

Unwin, Antony. "Geary's Contiguity Ratio." The Economic and Social Review 27, no. 2 (1996): 145-159.

Examples

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

`lg` *Local Geary*

Description

A local indicator of spatial association based on the Geary Ratio (Geary's C) for exploratory spatial data analysis. Large values of this statistic highlight local outliers, that is, values that are not like their neighbors.

Usage

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

Arguments

`x` Numeric vector of length `n`. By default, this will be standardized using the `scale` function.

`w` An `n x n` spatial connectivity matrix. See shape2mat.

`digits` Number of digits to round results to.

`scale` If `TRUE`, then `x` will automatically be standardized using `scale(x, center = TRUE, scale = TRUE)`.

`na.rm` If `na.rm = TRUE`, observations with `NA` values will be dropped from both `x` and `w`.

`warn` If `FALSE`, no warning will be printed to inform you when observations with `NA` values have been dropped, or if any observations without neighbors have been found.

Details

Local Geary's C is found in the numerator of the Geary Ratio (GR). For the i^{th} observation, the local Geary statistic is

$$C_i = \sum_j w_{i,j} * (x_i - x_j)^2$$

Hence, local Geary values will be largest for those observations that are most unlike their neighboring values. If a binary connectivity matrix is used (rather than row-standardized), then having many neighbors will also increase the value of the local Geary statistic. For most purposes, the row-standardized spatial weights matrix may be the more appropriate choice.

Value

The function returns a vector of numeric values, each value being a local indicator of spatial association (or dissimilarity), ordered as `x`.

Source

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

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

Examples

```
library(ggplot2)
data(georgia)
x <- log(georgia$income)
w <- shape2mat(georgia, "W")
lisd <- lg(x, w)
```

```
hist(lisd)
ggplot(georgia) +
  geom_sf(aes(fill = lisd)) +
scale_fill_gradient(high = "navy",
                    low = "white")
 ## or try: scale_fill_viridis()
```

`lisa` *Local Moran's I*

Description

A local indicator of spatial association (LISA) based on Moran's I (the Moran coefficient) for exploratory data analysis.

Usage

```
lisa(x, w, type = TRUE, scale = TRUE, digits = 3)
```

Arguments

`x` Numeric vector of length `n`.

`w` An `n x n` spatial connectivity matrix. See shape2mat. If `w` is not row standardized (`all(Matrix::rowSums(w) == 1)`), it will automatically be row-standardized.

`type` Return the type of association also (High-High, Low-Low, High-Low, and Low-High)? Defaults to `FALSE`.

`scale` If `TRUE`, then `x` will automatically be standardized using `scale(x, center = TRUE, scale = TRUE)`. If `FALSE`, then the variate will be centered but not scaled, using `scale(x, center = TRUE, scale = FALSE)`.

`digits` Number of digits to round results to.

Details

The values of `x` will automatically be centered first with `z = scale(x, center = TRUE, scale = scale)` (with user control over the `scale` argument). The LISA values are the product of each `z` value with the weighted sum of their respective surrounding value:

$$I_i = z_i \sum_j w_{ij} z_j$$

(or in R code: `lisa = z * (w %*% z)`). These are for exploratory analysis and model diagnostics.

An above-average value (i.e. positive z-value) with positive mean spatial lag indicates local positive spatial autocorrelation and is designated type "High-High"; a low value surrounded by high values indicates negative spatial autocorrelation and is designated type "Low-High", and so on.

This function uses Equation 7 from Anselin (1995). Note that the spdep package uses Formula 12, which divides the same value by a constant term $\sum_i z_i^2/n$. So the `geostan` version can be made equal to the spdep version by dividing by that value.

Value

If type = FALSE a numeric vector of lisa values for exploratory analysis of local spatial autocorrelation. If type = TRUE, a data.frame with columns Li (the lisa value) and type.

Source

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

See Also

moran_plot, mc, aple, lg, gr

Examples

```
library(ggplot2)
library(sf)
data(georgia)
w <- shape2mat(georgia, "W")
x <- georgia$ICE
li = lisa(x, w)
head(li)
ggplot(georgia, aes(fill = li$Li)) +
  geom_sf() +
  scale_fill_gradient2()
```

---

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)
```

me_diag *Measurement error model diagnostics*

Description

Visual diagnostics for spatial measurement error models.

Usage

```
me_diag(
  fit,
  varname,
  shape,
  probs = c(0.025, 0.975),
  plot = TRUE,
  mc_style = c("scatter", "hist"),
  size = 0.25,
  index = 0,
  style = c("W", "B"),
  w = shape2mat(shape, match.arg(style)),
  binwidth = function(x) 0.5 * sd(x)
)
```

Arguments

`fit` A `geostan_fit` model object as returned from a call to one of the `geostan::stan_*` functions.

`varname` Name of the modeled variable (a character string, as it appears in the model formula).

`shape` An object of class `sf` or another spatial object coercible to `sf` with `sf::st_as_sf`.

`probs` Lower and upper quantiles of the credible interval to plot.

`plot` If `FALSE`, return a list of `ggplot`s and a `data.frame` with the raw data values alongside a posterior summary of the modeled variable.

`mc_style` Character string indicating how to plot the Moran coefficient for the delta values: if `mc = "scatter"`, then `moran_plot` will be used with the marginal residuals; if `mc = "hist"`, then a histogram of Moran coefficient values will be returned, where each plotted value represents the degree of residual autocorrelation in a draw from the join posterior distribution of delta values.

`size` Size of points and lines, passed to `geom_pointrange`.

`index` Integer value; use this if you wish to identify observations with the largest `n=index` absolute Delta values; data on the top `n=index` observations ordered by absolute Delta value will be printed to the console and the plots will be labeled with the indices of the identified observations.

`style` Style of connectivity matrix; if `w` is not provided, `style` is passed to `shape2mat` and defaults to "W" for row-standardized.

w — An optional spatial connectivity matrix; if not provided, one will be created using shape2mat.

binwidth — A function with a single argument that will be passed to the binwidth argument in geom_histogram. The default is to set the width of bins to 0.5 * sd(x).

Value

A grid of spatial diagnostic plots for measurement error models comparing the raw observations to the posterior distribution of the true values. Includes a point-interval plot of raw values and modeled values; a Moran scatter plot for delta = z - x where z are the survey estimates and x are the modeled values; and a map of the delta values (take at their posterior means).

Source

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.

See Also

sp_diag, moran_plot, mc, aple

Examples

```
library(sf)
data(georgia)
## binary adjacency matrix
A <- shape2mat(georgia, "B")
## prepare data for the CAR model, using WCAR specification
cars <- prep_car_data(A, style = "WCAR")
## provide list of data for the measurement error model
ME <- prep_me_data(se = data.frame(ICE = georgia$ICE.se),
                   car_parts = cars)
## sample from the prior probability model only, including the ME model
fit <- stan_glm(log(rate.male) ~ ICE,
                ME = ME,
                data = georgia,
                prior_only = TRUE,
                iter = 800, # for speed only
                chains = 2, # for speed only
                refresh = 0 # silence some printing
                )

## see ME diagnostics
me_diag(fit, "ICE", georgia)
## see index values for the largest (absolute) delta values
 ## (differences between raw estimate and the posterior mean)
me_diag(fit, "ICE", georgia, index = 3)
```

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)
```

---

n_eff *Effective sample size*

Description

An approximate calculation for the effective sample size for spatially autocorrelated data. Only valid for approximately normally distributed data.

Usage

```
n_eff(n, rho)
```

Arguments

n	Number of observations.
rho	Spatial autocorrelation parameter from a simultaneous autoregressive model.

Details

Implements Equation 3 from Griffith (2005).

Value

Returns effective sample size n*, a numeric value.

Source

Griffith, Daniel A. (2005). Effective geographic sample size in the presence of spatial autocorrelation. Annals of the Association of American Geographers. Vol. 95(4): 740-760.

See Also

sim_sar, aple

Examples

```
n_eff(100, 0)
n_eff(100, 0.5)
n_eff(100, 0.9)
n_eff(100, 1)

rho <- seq(0, 1, by = 0.01)
plot(rho, n_eff(100, rho),
     type = 'l',
     ylab = "Effective Sample Size")
```

n_nbs *Count neighbors in a connectivity matrix*

Description

Count neighbors in a connectivity matrix

Usage

```
n_nbs(C)
```

Arguments

C	A connectivity matrix

Value

A vector with the number of non-zero values in each row of C

Examples

```
data(sentencing)
C <- shape2mat(sentencing)
sentencing$Ni <- n_nbs(C)
```

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)
```

predict.geostan_fit *Predict method for* geostan_fit *models*

Description

Obtain predicted values from a fitted model by providing new covariate values.

Usage

```
## S3 method for class 'geostan_fit'
predict(
  object,
  newdata,
  alpha = as.matrix(object, pars = "intercept"),
  center = object$x_center,
  summary = TRUE,
  type = c("link", "response"),
  ...
)
```

Arguments

object	A fitted model object of class geostan_fit.

newdata	A data frame in which to look for variables with which to predict, presumably for the purpose of viewing marginal effects. Note that if the model formula includes an offset term, newdata must contain the offset. Note also that any spatially-lagged covariate terms will be ignored if they were provided using the slx argument. If covariates in the model were centered using the centerx argument, the predict.geostan_fit method will automatically center the predictors in newdata using the values stored in object$x_center. If newdata is missing, the fitted values of the model will be returned.

alpha	An N-by-1 matrix of MCMC samples for the intercept; this is provided by default. However, this argument might be used if there is a need to incorporate the spatial trend term, in which case it may be thought of as a spatially-varying intercept. If used, note that the intercept needs to be provided on the scale of the linear predictor.

center	Optional vector of numeric values or a logical scalar to pass to scale. Defaults to using object$x_center. If the model was fit using centerx = TRUE, then covariates were centered and their mean values are stored in object$x_center and the predict method will use them to automatically center newdata; if the model was fit with centerx = FALSE, then object$x_center = FALSE and newdata will not be centered.

summary	If FALSE, a matrix containing samples from the posterior distribution at each observation is returned. The default, TRUE, will summarize results by providing an estimate (mean) and credible interval (formed by taking quantiles of the MCMC samples).

`type`	By default, results from `predict` are on the scale of the linear predictor (`type = "link"`)). The alternative (`type = "response"`) is on the scale of the response variable. For example, the default return values for a Poisson model on the log scale, and using `type = "response"` will return the original scale of the outcome variable (by exponentiating the log values).
`...`	Not used

Details

The purpose of the predict method is to explore marginal effects of (combinations of) covariates.

The model formula will be taken from `object$formula`, and then a model matrix will be created by passing `newdata` to the model.frame function (as in: `model.frame(newdata, object$formula)`. Parameters are taken from `as.matrix(object, pars = c("intercept", "beta"))`.

Be aware that in generalized linear models (such as Poisson and Binomial models) marginal effects plots on the response scale may be sensitive to the level of other covariates in the model. If the model includes any spatially-lagged covariates (introduced using the `slx` argument) or a spatial autocorrelation term (for example, you used a spatial CAR, SAR, or ESF model), these terms will essentially be fixed at zero for the purposes of calculating marginal effects. If you want to change this, you can introduce spatial trend values by specifying a varying intercept using the `alpha` argument.

Value

If `summary = FALSE`, a matrix of samples is returned. If `summary = TRUE` (the default), a data frame is returned.

Examples

```
data(georgia)
georgia$income <- georgia$income / 1e3

fit <- stan_glm(deaths.male ~ offset(log(pop.at.risk.male)) + log(income),
               data = georgia,
               centerx = TRUE,
               family = poisson(),
               chains = 2, iter = 600) # for speed only

# note: pop.at.risk.male=1 leads to log(pop.at.risk.male)=0
# so that the predicted values are rates
newdata <- data.frame(
             income = seq(min(georgia$income),
                          max(georgia$income),
                           length.out = 100),
             pop.at.risk.male = 1)

preds <- predict(fit, newdata, type = "response")
head(preds)
plot(preds$income,
     preds$mean * 10e3,
     type = "l",
     ylab = "Deaths per 10,000",
     xlab = "Income ($1,000s)")

# here the predictions are rates per 10,000
newdata$pop.at.risk.male <- 10e3
preds <- predict(fit, newdata, type = "response")
head(preds)
plot(preds$income,
    preds$mean,
    type = "l",
    ylab = "Deaths per 10,000",
    xlab = "Income ($1,000s)")
```

`prep_car_data` *Prepare data for the CAR model*

Description

Prepare data for the CAR model

Usage

```
prep_car_data(
  A,
  style = c("WCAR", "ACAR", "DCAR"),
  k = 1,
  gamma = 0,
  lambda = TRUE,
  cmat = TRUE,
  stan_fn = ifelse(style == "WCAR", "wcar_normal_lpdf", "car_normal_lpdf")
)
```

Arguments

`A` Binary adjacency matrix; for `style` = `DCAR`, provide a symmetric matrix of distances instead. The distance matrix should be sparse, meaning that most distances should be zero (usually obtained by setting some threshold distance beyond which all are zero).

`style` Specification for the connectivity matrix (C) and conditional variances (M); one of "WCAR", "ACAR", or "DCAR".

`k` For `style` = `DCAR`, distances will be raised to the -k power (d^-k).

`gamma` For `style` = `DCAR`, distances will be offset by `gamma` before raising to the `-k`th power.

`lambda` If TRUE, return eigenvalues required for calculating the log determinant of the precision matrix and for determining the range of permissible values of rho. These will also be printed with a message if lambda = TRUE.

`cmat` If `cmat` = `TRUE`, return the full matrix C (in sparse matrix format).

`stan_fn` Two computational methods are available for CAR models using `stan_car`: `car\_normal\_lpdf` and `wcar\_normal\_lpdf`. For WCAR models, either method will work but `wcar\_normal\_lpdf` is faster. To force use `car\_normal\_lpdf` when `style = 'WCAR'`, provide `stan_fn = "car_normal_lpdf"`.

Details

The CAR model is:

```
Normal(Mu, Sigma), Sigma = (I - rho * C)^-1 * M * tau^2,
```

where `I` is the identity matrix, `rho` is a spatial autocorrelation parameter, `C` is a connectivity matrix, and `M * tau^2` is a diagonal matrix with conditional variances on the diagonal. `tau^2` is a (scalar) scale parameter.

In the WCAR specification, `C` is the row-standardized version of `A`. This means that the non-zero elements of `A` will be converted to `1/N_i` where `N_i` is the number of neighbors for the `i`th site (obtained using `Matrix::rowSums(A)`. The conditional variances (on the diagonal of `M * tau^2`), are also proportional to `1/N_i`.

The ACAR specification is from Cressie, Perrin and Thomas-Agnon (2005); also see Cressie and Wikle (2011, p. 188) and Donegan (2021).

The DCAR specification is inverse distance-based, and requires the user provide a (sparse) distance matrix instead of a binary adjacency matrix. (For `A`, provide a symmetric matrix of distances, not inverse distances!) Internally, non-zero elements of `A` will be converted to: `d_{ij} = (a_{ij} + gamma)^(-k)` (Cliff and Ord 1981, p. 144; Donegan 2021). Default values are `k=1` and `gamma=0`. Following Cressie (2015), these values will be scaled (divided) by their maximum value. For further details, see the DCAR_A specification in Donegan (2021).

For inverse-distance weighting schemes, see Cliff and Ord (1981); for distance-based CAR specifications, see Cressie (2015 [1993]), Haining and Li (2020), and Donegan (2021).

When using `stan_car`, always use `cmat = TRUE` (the default).

Details on CAR model specifications can be found in Table 1 of Donegan (2021).

Value

A list containing all of the data elements required by the CAR model in `stan_car`.

Source

Cliff A, Ord J (1981). Spatial Processes: Models and Applications. Pion.

Cressie N (2015 [1993]). Statistics for Spatial Data. Revised edition. John Wiley & Sons.

Cressie N, Perrin O, Thomas-Agnan C (2005). “Likelihood-based estimation for Gaussian MRFs.” Statistical Methodology, 2(1), 1–16.

Cressie N, Wikle CK (2011). Statistics for Spatio-Temporal Data. John Wiley & Sons.

Donegan, Connor (2021). Spatial conditional autoregressive models in Stan. *OSF Preprints*. doi:10.31219/osf.io/3ey65.

Haining RP, Li G (2020). Modelling Spatial and Spatio-Temporal Data: A Bayesian Approach. CRC Press.

Examples

```
data(georgia)

## use a binary adjacency matrix
A <- shape2mat(georgia, style = "B")

## WCAR specification
cp <- prep_car_data(A, "WCAR")
1 / range(cp$lambda)

## ACAR specification
cp <- prep_car_data(A, "ACAR")

## DCAR specification (inverse-distance based)
A <- shape2mat(georgia, "B")
D <- sf::st_distance(sf::st_centroid(georgia))
A <- D * A
cp <- prep_car_data(A, "DCAR", k = 1)
```

`prep_car_data2` *Prepare data for the CAR model: raster analysis*

Description

Prepare a list of data required for the CAR model; this is for working with (large) raster data files only. For non-raster analysis, see prep_car_data.

Usage

```
prep_car_data2(row = 100, col = 100)
```

Arguments

`row`	Number of rows in the raster
`col`	Number of columns in the raster

Details

Prepare input data for the CAR model when your dataset consists of observations on a regular (rectangular) tessellation, such as a raster layer or remotely sensed imagery. The rook criteria is used to determine adjacency. This function uses Equation 5 from Griffith (2000) to generate approximate eigenvalues for a row-standardized spatial weights matrix from a P-by-Q dimension regular tessellation.

This function can accommodate very large numbers of observations for use with `stan_car`; for large N data, it is also recommended to use `slim = TRUE` or the `drop` argument. For more details, see: `vignette("raster-regression", package = "geostan")`.

Source

Griffith, Daniel A. (2000). Eigenfunction properties and approximations of selected incidence matrices employed in spatial analyses. *Linear Algebra and its Applications* 321 (1-3): 95-112. doi:10.1016/S00243795(00)000318.

See Also

prep_sar_data2, prep_car_data, stan_car.

Examples

```
row = 100
col = 120
car_dl <- prep_car_data2(row = row, col = col)
```

---

prep_icar_data *Prepare data for ICAR models*

Description

Given a symmetric n x n connectivity matrix, prepare data for intrinsic conditional autoregressive models in Stan. This function may be used for building custom ICAR models in Stan. This is used internally by stan_icar.

Usage

```
prep_icar_data(C, scale_factor = NULL)
```

Arguments

C	Connectivity matrix
scale_factor	Optional vector of scale factors for each connected portion of the graph structure. If not provided by the user it will be fixed to a vector of ones.

Details

This is used internally to prepare data for stan_icar models. It can also be helpful for fitting custom ICAR models outside of geostan.

Value

list of data to add to Stan data list:

k number of groups

group_size number of nodes per group

n_edges number of connections between nodes (unique pairs only)

node1 first node

node2 second node. (`node1[i]` and `node2[i]` form a connected pair)

weight The element `C[node1, node2]`.

group_idx indices for each observation belonging each group, ordered by group.

m number of disconnected regions requiring their own intercept.

A n-by-m matrix of dummy variables for the component-specific intercepts.

inv_sqrt_scale_factor By default, this will be a k-length vector of ones. Placeholder for user-specified information. If user provided `scale_factor`, then this will be `1/sqrt(scale_factor)`.

comp_id n-length vector indicating the group membership of each observation.

Source

Besag, Julian, Jeremy York, and Annie Mollié. 1991. "Bayesian Image Restoration, with Two Applications in Spatial Statistics." Annals of the Institute of Statistical Mathematics 43 (1): 1–20.

Donegan, Connor. Flexible Functions for ICAR, BYM, and BYM2 Models in Stan. Code Repository. 2021. Available online: https://github.com/ConnorDonegan/Stan-IAR/ (accessed Sept. 10, 2021).

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, Katherine Wheeler-Martin, Dan Simpson, Stephen J Mooney, Andrew Gelman, and Charles DiMaggio. 2019. "Bayesian Hierarchical Spatial Models: Implementing the Besag York Mollié Model in Stan." Spatial and Spatio-Temporal Epidemiology 31: 100301.

Riebler, Andrea, Sigrunn H Sørbye, Daniel Simpson, and Håvard Rue. 2016. "An Intuitive Bayesian Spatial Model for Disease Mapping That Accounts for Scaling." Statistical Methods in Medical Research 25 (4): 1145–65.

See Also

`edges`, `shape2mat`, `stan_icar`, `prep_car_data`

Examples

```
data(sentencing)
C <- shape2mat(sentencing)
icar.data.list <- prep_icar_data(C)
```

`prep_me_data` *Prepare data for spatial measurement error models*

Description

Prepares the list of data required for geostan's (spatial) measurement error models. Given a data frame of standard errors and any optional arguments, the function returns a list with all required data for the models, filling in missing elements with default values.

Usage

```
prep_me_data(
  se,
  car_parts,
  prior,
  logit = rep(FALSE, times = ncol(se)),
  bounds = c(-Inf, Inf)
)
```

Arguments

`se` Data frame of standard errors; column names must match (exactly) the variable names used in the model formula.

`car_parts` A list of data required for spatial CAR models, as created by `prep_car_data`; optional. If omitted, the measurement error model will be a non-spatial Student's t model.

`prior` A named list of prior distributions (see `priors`). If none are provided, default priors will be assigned. The list of priors may include the following parameters:

df If using a non-spatial ME model, the degrees of freedom (df) for the Student's t model is assigned a gamma prior with default parameters of `gamma2(alpha = 3, beta = 0.2)`. Provide values for each covariate in `se`, listing the values in the same order as the columns of `se`.

location The prior for the location parameter (mu) is a normal (Gaussian) distribution (the default being `normal(location = 0, scale = 100)`). To adjust the prior distributions, provide values for each covariate in `se`, listing the values in the same order as the columns of se.

scale The prior for the scale parameters is Student's t, and the default parameters are `student_t(df = 10, location = 0, scale = 40)`. To adjust, provide values for each covariate in `se`, listing the values in the same order as the columns of se.

car_rho The CAR model, if used, has a spatial autocorrelation parameter, `rho`, which is assigned a uniform prior distribution. You must specify values that are within the permissible range of values for `rho`; these are automatically printed to the console by the `prep_car_data` function.

`logit` Optional vector of logical values (`TRUE`, `FALSE`) indicating if the latent variable should be logit-transformed. Only use for rates. This keeps rates between zero and one and may improve modeling of skewed variables (e.g., the poverty rate).

`bounds` Rarely needed; an optional numeric vector of length two providing the upper and lower bounds, respectively, of the variables (e.g., a magnitudes must be greater than 0). If not provided, they will be set to c(-Inf, Inf) (i.e., unbounded).

Value

A list of data as required for (spatial) ME models. Missing arguments will be filled in with default values, including prior distributions.

See Also

se_log

Examples

```
data(georgia)

## for a non-spatial prior model for two covariates
se <- data.frame(ICE = georgia$ICE.se,
                 college = georgia$college.se)
ME <- prep_me_data(se)

## see default priors
print(ME$prior)

## set prior for the scale parameters
ME <- prep_me_data(se,
                   prior = list(scale = student_t(df = c(10, 10),
                                                  location = c(0, 0),
                                                  scale = c(20, 20))))

## for a spatial prior model (often recommended)
A <- shape2mat(georgia, "B")
cars <- prep_car_data(A)
ME <- prep_me_data(se,
                   car_parts = cars)
```

`prep_sar_data` *Prepare data for a simultaneous autoregressive (SAR) model*

Description

Given a spatial weights matrix W, this function prepares data for the simultaneous autoregressive (SAR) model (a.k.a spatial error model (SEM)) in Stan. This is used internally by `stan_sar`, and may also be used for building custom SAR models in Stan.

Usage

```
prep_sar_data(W)
```

Arguments

W	Spatial weights matrix, typically row-standardized.

Details

This is used internally to prepare data for `stan_sar` models. It can also be helpful for fitting custom SAR models in Stan (outside of `geostan`).

Value

list of data to add to a Stan data list:

ImW_w Numeric vector containing the non-zero elements of matrix $(I - W)$.

ImW_v An integer vector containing the column indices of the non-zero elements of $(I - W)$.

ImW_u An integer vector indicating where in `ImW_w` a given row's non-zero values start.

nImW_w Number of entries in `ImW_w`.

Widx Integer vector containing the indices corresponding to values of `-W` in `ImW_w` (i.e. non-diagonal entries of $(I - W)$).

nW Integer length of `Widx`.

eigenvalues_w Eigenvalues of W matrix.

n Number of rows in W.

W Sparse matrix representation of W

rho_min Minimum permissible value of ρ (`1/min(eigenvalues_w)`).

rho_max Maximum permissible value of ρ (`1/max(eigenvalues_w)`.

The function will also print the range of permissible ρ values to the console.

See Also

shape2mat, stan_sar, prep_car_data, prep_icar_data

Examples

```
data(georgia)
W <- shape2mat(georgia, "W")
sar_dl <- prep_sar_data(W)
```

prep_sar_data2 *Prepare data for SAR model: raster analysis*

Description

Prepares a list of data required for using the SAR model; this is for working with (large) raster data files. For non-raster analysis, see prep_sar_data.

Usage

```
prep_sar_data2(row, col)
```

Arguments

`row`	Number of rows in the raster
`col`	Number of columns in the raster

Details

Prepare data for the SAR model when your raw dataset consists of observations on a regular tessellation, such as a raster layer or remotely sensed imagery. The rook criteria is used to determine adjacency. This function uses Equation 5 from Griffith (2000) to calculate the eigenvalues for a row-standardized spatial weights matrix of a P-by-Q dimension regular tessellation.

This function can accommodate very large numbers of observations for use with `stan_sar`; for large N data, it is also recommended to use `slim = TRUE` or the `drop` argument. For details, see: `vignette("raster-regression", package = "geostan")`.

Source

Griffith, Daniel A. (2000). Eigenfunction properties and approximations of selected incidence matrices employed in spatial analyses. *Linear Algebra and its Applications* 321 (1-3): 95-112. doi:10.1016/S00243795(00)000318.

See Also

`prep_car_data2`, `prep_sar_data`, `stan_sar`.

Examples

```
row = 100
col = 120
sar_dl <- prep_sar_data2(row = row, col = col)
```

print.geostan_fit *print or plot a fitted geostan model*

Description

Print a summary of model results to the R console, or plot posterior distributions of model parameters.

Usage

```
## S3 method for class 'geostan_fit'
print(x, probs = c(0.025, 0.2, 0.5, 0.8, 0.975), digits = 3, pars = NULL, ...)

## S3 method for class 'geostan_fit'
plot(x, pars, plotfun = "hist", fill = "steelblue4", ...)
```

Arguments

`x` A fitted model object of class `geostan_fit`.

`probs` Argument passed to `quantile`; which quantiles to calculate and print.

`digits` number of digits to print

`pars` parameters to include; a character string (or vector) of parameter names.

`...` additional arguments to `rstan::plot` or `rstan::print.stanfit`.

`plotfun` Argument passed to `rstan::plot`. Options include histograms ("hist"), MCMC traceplots ("trace"), and density plots ("dens"). Diagnostic plots are also available such as Rhat statistics ("rhat"), effective sample size ("ess"), and MCMC autocorrelation ("ac").

`fill` fill color for histograms and density plots.

Examples

```
data(georgia)
georgia$income <- georgia$income/1e3

fit <- stan_glm(deaths.male ~ offset(log(pop.at.risk.male)) + log(income),
                centerx = TRUE,
                data = georgia,
                family = poisson(),
                chains = 2, iter = 600) # for speed only


# print and plot results
print(fit)
plot(fit)
```

priors *Prior distributions*

Description

Prior distributions

Usage

```
uniform(lower, upper, variable = NULL)

normal(location = 0, scale, variable = NULL)

student_t(df = 10, location = 0, scale, variable = NULL)

gamma2(alpha, beta, variable = NULL)

hs(global_scale = 1, slab_df = 10, slab_scale, variable = "beta_ev")
```

Arguments

`lower, upper`	lower and upper bounds of the distribution
`variable`	A reserved slot for the variable name; if provided by the user, this may be ignored by **geostan**.
`location`	Location parameter(s), numeric value(s)
`scale`	Scale parameter(s), positive numeric value(s)
`df`	Degrees of freedom, positive numeric value(s)
`alpha`	shape parameter, positive numeric value(s)
`beta`	inverse scale parameter, positive numeric value(s)
`global_scale`	Control the (prior) degree of sparsity in the horseshoe model (0 < global_scale < 1).
`slab_df`	Degrees of freedom for the Student's t model for large coefficients in the horseshoe model (slab_df > 0).
`slab_scale`	Scale parameter for the Student's t model for large coefficients in the horseshoe model (slab_scale > 0).

Details

The prior distribution functions are used to set the values of prior parameters.

Users can control the values of the parameters, but the distribution (model) itself is fixed. The intercept and regression coefficients are given Gaussian prior distributions and scale parameters are assigned Student's t prior distributions. Degrees of freedom parameters are assigned gamma priors, and the spatial autocorrelation parameter in the CAR model, rho, is assigned a uniform prior. The horseshoe (`hs`) model is used by `stan_esf`.

Note that the variable argument is used internally by geostan, and any user provided values will be ignored.

Parameterizations:

For details on how any distribution is parameterized, see the Stan Language Functions Reference document: https://mc-stan.org/users/documentation/.

The horseshoe prior:

The horseshoe prior is used by stan_esf as a prior for the eigenvector coefficients. The horseshoe model encodes a prior state of knowledge that effectively states, 'I believe a small number of these variables may be important, but I don't know which of them is important.' The horseshoe is a normal distribution with unknown scale (Polson and Scott 2010):

```
beta_j ~ Normal(0, tau^2 * lambda_j^2)
```

The scale parameter for this prior is the product of two terms: lambda_j^2 is specific to the variable beta_j, and tau^2 is known as the global shrinkage parameter.

The global shrinkage parameter is assigned a half-Cauchy prior:

```
tau ~ Cauchy(0, global_scale * sigma)
```

where global_scale is provided by the user and sigma is the scale parameter for the outcome variable; for Poisson and binomial models, sigma is fixed at one. Use global_scale to control the overall sparsity of the model.

The second part of the model is a Student's t prior for lambda_j. Most lambda_j will be small, since the model is half-Cauchy:

```
lambda_j ~ Cauchy(0, 1)
```

This model results in most lambda_j being small, but due to the long tails of the Cauchy distribution, strong evidence in the data can force any particular lambda_j to be large. Piironen and Vehtari (2017) adjust the model so that those large lambda_j are effectively assigned a Student's t model:

```
Big_lambda_j ~ Student_t(slab_df, 0, slab_scale)
```

This is a schematic representation of the model; see Piironen and Vehtari (2017) or Donegan et al. (2020) for details.

Value

An object of class prior which will be used internally by **geostan** to set parameters of prior distributions.

Student's t:

Return value for student_t depends on the input; if no arguments are provided (specifically, if the scale parameter is missing), this will return an object of class 'family'; if at least the scale parameter is provided, student_t will return an object of class prior containing parameter values for the Student's t distribution.

Source

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).

Polson, N.G. and J.G. Scott (2010). Shrink globally, act locally: Sparse Bayesian regularization and prediction. *Bayesian Statistics* 9, 501-538.

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

```
# std. normal priors to k=2 covariates
data(georgia)
prior <- list()
k <- 2
prior$beta <- normal(location = rep(0, times = k),
                     scale = rep(1, times = k))
prior$intercept <- normal(-5, 3)
print(prior)
fit <- stan_glm(deaths.male ~ offset(log(pop.at.risk.male)) + ICE + college,
                re = ~ GEOID,
                data = georgia,
                family = poisson(),
                prior = prior,
                prior_only = TRUE,
                chains = 2, iter = 600) # for speed only
plot(fit)

se <- data.frame(insurance = georgia$insurance.se)
prior <- list()
prior$df <- gamma2(3, 0.2)
prior$location <- normal(50, 50)
prior$scale <- student_t(12, 10, 20)
print(prior)
ME <- prep_me_data(se = se, prior = prior)
fit <- stan_glm(log(rate.male) ~ insurance,
                data = georgia,
                ME = ME,
                prior_only = TRUE,
               chains = 2, iter = 600) # for speed only
```

`residuals.geostan_fit` *Extract residuals, fitted values, or the spatial trend*

Description

Extract model residuals, fitted values, or spatial trend from a fitted `geostan_fit` model.

Usage

```
## S3 method for class 'geostan_fit'
residuals(object, summary = TRUE, rates = TRUE, detrend = TRUE, ...)

## S3 method for class 'geostan_fit'
fitted(object, summary = TRUE, rates = TRUE, trend = TRUE, ...)

spatial(object, summary = TRUE, ...)

## S3 method for class 'geostan_fit'
spatial(object, summary = TRUE, ...)
```

Arguments

`object` A fitted model object of class `geostan_fit`.

`summary` Logical; should the values be summarized by their mean, standard deviation, and quantiles (`probs = c(.025, .2, .5, .8, .975)`) for each observation? Otherwise, a matrix containing samples from the posterior distributions is returned.

`rates` For Poisson and Binomial models, should the fitted values be returned as rates, as opposed to raw counts? Defaults to `TRUE`; see the `Details` section for more information.

`detrend` For auto-normal models (CAR and SAR models with Gaussian likelihood only); if `detrend = TRUE`, the implicit spatial trend will be removed from the residuals. The implicit spatial trend is `Trend = rho * C %*% (Y - Mu)` (see stan_car or stan_sar). I.e., `resid = Y - (Mu + Trend)`.

`...` Not used

`trend` For auto-normal models (CAR and SAR models with Gaussian likelihood only); if `trend = TRUE`, the fitted values will include the implicit spatial trend term. The implicit spatial trend is `Trend = rho * C %*% (Y - Mu)` (see stan_car or stan_sar). I.e., if `trend = TRUE`, `fitted = Mu + Trend`.

Details

When `rates = FALSE` and the model is Poisson or Binomial, the fitted values returned by the `fitted` method are the expected value of the response variable. The `rates` argument is used to translate count outcomes to rates by dividing by the appropriate denominator. The behavior of the `rates` argument depends on the model specification. Consider a Poisson model of disease incidence, such as the following intercept-only case:

```
fit <- stan_glm(y ~ offset(log(E)),
                data = data,
                family = poisson())
```

If the fitted values are extracted using `rates = FALSE`, then `fitted(fit)` will return the expectation of y. If `rates = TRUE` (the default), then `fitted(fit)` will return the expected value of the rate $\frac{y}{E}$.

If a binomial model is used instead of the Poisson, then using `rates = TRUE` will return the expectation of $\frac{y}{N}$ where N is the sum of the number of 'successes' and 'failures', as in:

```
fit <- stan_glm(cbind(successes, failures) ~ 1,
                data = data,
                family = binomial())
```

Examples

```
data(georgia)
A <- shape2mat(georgia, "B")

fit <- stan_esf(deaths.male ~ offset(log(pop.at.risk.male)),
                C = A,
                data = georgia,
                family = poisson(),
                chains = 1, iter = 600) # for speed only


# Residuals
r <- resid(fit)
moran_plot(r$mean, A)
head(r)

# Fitted values
f <- fitted(fit)

# Fitted values, unstandardized
f <- fitted(fit, rates = FALSE)
head(f)

# Spatial trend
esf <- spatial(fit)
head(esf)
```

row_standardize *Row-standardize a matrix; safe for zero row-sums.*

Description

Row-standardize a matrix; safe for zero row-sums.

Usage

```
row_standardize(C, warn = TRUE, msg = "Row standardizing connectivity matrix")
```

Arguments

C	A matrix
warn	Print msg if warn = TRUE.
msg	A warning message to print.

Value

A row-standardized matrix, W (i.e., all row sums equal 1, or zero).

Examples

```
A <- shape2mat(georgia)
head(Matrix::summary(A))
Matrix::rowSums(A)

W <- row_standardize(A)
head(Matrix::summary(W))
Matrix::rowSums(W)
```

`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)
```

se_log *Standard error of log(x)*

Description

Transform the standard error of x to standard error of log(x).

Usage

```
se_log(x, se, method = c("mc", "delta"), nsim = 5000, bounds = c(0, Inf))
```

Arguments

x	An estimate
se	Standard error of x
method	The "delta" method uses a Taylor series approximation; the default method, "mc", uses a simple monte carlo method.
nsim	Number of draws to take if method = "mc".
bounds	Lower and upper bounds for the variable, used in the monte carlo method. Must be a length-two numeric vector with lower bound greater than or equal to zero (i.e. c(lower, upper) as in default bounds = c(0, Inf).

Details

The delta method returns x^(-1) * se. The monte carlo method is detailed in the examples section.

Value

Numeric vector of standard errors

Examples

```
data(georgia)
x = georgia$college
se = georgia$college.se

lse1 = se_log(x, se)
lse2 = se_log(x, se, method = "delta")
plot(lse1, lse2); abline(0, 1)

# the monte carlo method
x = 10
se = 2
```

```
z = rnorm(n = 20e3, mean = x,  sd = se)
l.z = log(z)
sd(l.z)
se_log(x, se, method = "mc")
se_log(x, se, method = "delta")
```

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)
```

`sim_sar` *Simulate spatially autocorrelated data*

Description

Given a spatial weights matrix and degree of autocorrelation, returns autocorrelated data.

Usage

```
sim_sar(m = 1, mu = rep(0, nrow(w)), w, rho, sigma = 1, ...)
```

Arguments

`m` The number of samples required. Defaults to `m=1` to return an `n`-length vector; if `m>1`, an `m x n` matrix is returned (i.e. each row will contain a sample of correlated values).

`mu` An `n`-length vector of mean values. Defaults to a vector of zeros with length equal to `nrow(w)`.

`w` Row-standardized `n x n` spatial weights matrix.

`rho` Spatial autocorrelation parameter in the range (-1, 1). Typically a scalar value; otherwise an n-length numeric vector.

`sigma` Scale parameter (standard deviation). Defaults to `sigma = 1`. Typically a scalar value; otherwise an n-length numeric vector.

`...` further arguments passed to `MASS::mvrnorm`.

Details

Calls `MASS::mvrnorm` internally to draw from the multivariate normal distribution. The covariance matrix is specified following the simultaneous autoregressive (SAR) model.

Value

If `m = 1` a vector of the same length as `mu`, otherwise an `m x length(mu)` matrix with one sample in each row.

See Also

aple, mc, moran_plot, lisa, shape2mat

Examples

```
data(georgia)
w <- shape2mat(georgia, "W")
x <- sim_sar(w = w, rho = 0.5)
aple(x, w)

x <- sim_sar(w = w, rho = 0.7, m = 10)
dim(x)
apply(x, 1, aple, w = w)
```

sp_diag *Visual displays of spatial data and spatial models*

Description

Visual diagnostics for areal data and model residuals

Usage

```
sp_diag(y, shape, ...)

## S3 method for class 'geostan_fit'
sp_diag(
  y,
  shape,
  name = "Residual",
  plot = TRUE,
  mc_style = c("scatter", "hist"),
  style = c("W", "B"),
  w,
  rates = TRUE,
  binwidth = function(x) 0.5 * stats::sd(x, na.rm = TRUE),
  size = 0.1,
  ...
)

## S3 method for class 'numeric'
sp_diag(
  y,
  shape,
  name = "y",
  plot = TRUE,
  mc_style = c("scatter", "hist"),
  style = c("W", "B"),
  w = shape2mat(shape, match.arg(style)),
  binwidth = function(x) 0.5 * stats::sd(x, na.rm = TRUE),
  ...
)
```

Arguments

y
A numeric vector, or a fitted geostan model (class geostan_fit).

shape
An object of class sf or another spatial object coercible to sf with sf::st_as_sf such as SpatialPolygonsDataFrame.

...
Additional arguments passed to residuals.geostan_fit. For binomial and Poisson models, this includes the option to view the outcome variable as a rate (the default) rather than a count; for stan_car models with auto-Gaussian likelihood (fit$family$family = "auto_gaussian"), the residuals will be detrended by default, trend = FALSE‘.

name
The name to use on the plot labels; default to "y" or, if y is a geostan_fit object, to "Residuals".

plot
If FALSE, return a list of gg plots.

mc_style
Character string indicating how to plot the residual Moran coefficient (only used if y is a fitted model): if mc = "scatter", then moran_plot will be used with the marginal residuals; if mc = "hist", then a histogram of Moran coefficient values will be returned, where each plotted value represents the degree of residual autocorrelation in a draw from the join posterior distribution of model parameters.

style
Style of connectivity matrix; if w is not provided, style is passed to shape2mat and defaults to "W" for row-standardized.

w
An optional spatial connectivity matrix; if not provided and y is a numeric vector, one will be created using shape2mat. If w is not provided and y is a fitted geostan model, then the spatial connectivity matrix that is stored with the fitted model (y$C) will be used.

rates
For Poisson and binomial models, convert the outcome variable to a rate before plotting fitted values and residuals. Defaults to rates = TRUE.

binwidth
A function with a single argument that will be passed to the binwidth argument in geom_histogram. The default is to set the width of bins to 0.5 * sd(x).

size
Point size and linewidth for point-interval plot of observed vs. fitted values (passed to geom_pointrange).

Details

When provided with a numeric vector, this function plots a histogram, Moran scatter plot, and map.

When provided with a fitted geostan model, the function returns a point-interval plot of observed values against fitted values (mean and 95 percent credible interval), either a Moran scatter plot of residuals or a histogram of Moran coefficient values calculated from the joint posterior distribution of the residuals, and a map of the mean posterior residuals (means of the marginal distributions).

When y is a fitted CAR or SAR model with family = auto_gaussian(), the fitted values will include implicit spatial trend term, i.e. the call to fitted.geostan_fit will use the default trend =

TRUE and the call to residuals.geostan_fit will use the default detrend = TRUE. (See stan_car or stan_sar for additional details on their implicit spatial trend components.)

Value

A grid of spatial diagnostic plots. If plot = TRUE, the ggplots are drawn using grid.arrange; otherwise, they are returned in a list. For the geostan_fit method, the underlying data for the Moran coefficient (as required for mc_style = "hist") will also be returned if plot = FALSE.

See Also

me_diag, mc, moran_plot, aple

Examples

```
data(georgia)
sp_diag(georgia$college, georgia)

bin_fn <- function(y) mad(y, na.rm = TRUE)
sp_diag(georgia$college, georgia, binwidth = bin_fn)

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

stan_car *Conditional autoregressive (CAR) models*

Description

Use the CAR model as a prior on parameters, or fit data to a spatial Gaussian CAR model.

Usage

```
stan_car(
  formula,
  slx,
  re,
  data,
  car_parts,
  C,
  family = gaussian(),
  prior = NULL,
  ME = NULL,
  centerx = FALSE,
  prior_only = FALSE,
  censor_point,
  chains = 4,
  iter = 2000,
  refresh = 500,
  keep_all = FALSE,
  slim = FALSE,
  drop = NULL,
  pars = NULL,
  control = NULL,
  ...
)
```

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).
```

With the CAR model, any `alpha_re` term should be at a *different* level or scale than the observations; that is, at a different scale than the autocorrelation structure of the CAR model itself.

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

`car_parts` A list of data for the CAR model, as returned by `prep_car_data`.

`C` Optional spatial connectivity matrix which will be used to calculate residual spatial autocorrelation as well as any user specified `slx` terms; it will automatically be row-standardized before calculating `slx` terms. See `shape2mat`.

`family` The likelihood function for the outcome variable. Current options are `auto_gaussian()`, `binomial(link = "logit")`, and `poisson(link = "log")`; if `family = gaussian()` is provided, it will automatically be converted to `auto_gaussian()`.

`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.

car_scale Scale parameter for the CAR model, `car_scale`. The scale is assigned a Student's t prior model (constrained to be positive).

car_rho The spatial autocorrelation parameter in the CAR model, `rho`, is assigned a uniform prior distribution. By default, the prior will be uniform over all permissible values as determined by the eigenvalues of the connectivity matrix, `C`. The range of permissible values for `rho` is automatically printed to the console by `prep_car_data`.

tau The scale parameter for any varying intercepts (a.k.a exchangeable random effects, or partial pooling) terms. This scale parameter, `tau`, is assigned a Student's t prior (constrained to be positive).

`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.

`prior_only` Logical value; if `TRUE`, draw samples only from the prior distributions of parameters.

`censor_point` Integer value indicating the maximum censored value; this argument is for modeling censored (suppressed) outcome data, typically disease case counts or deaths.

`chains` Number of MCMC chains to use.

`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 CAR models:

fitted The N-length vector of fitted values

log_lambda_mu Linear predictor inside the CAR model (for Poisson and binomial models)

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.

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.

`...` Other arguments passed to `sampling`.

Details

CAR models are discussed in Cressie and Wikle (2011, p. 184-88), Cressie (2015, Ch. 6-7), and Haining and Li (2020, p. 249-51). It is often used for areal or lattice data.

Details for the Stan code for this implementation of the CAR model can be found in Donegan (2021).

The general scheme for the CAR model is as follows:

$$y \sim Gauss(\mu, (I - \rho C)^{-1} M),$$

where I is the identity matrix, ρ is a spatial dependence parameter, C is a spatial connectivity matrix, and M is a diagonal matrix of variance terms. The diagonal of M contains a scale parameter τ multiplied by a vector of weights (often set to be proportional to the inverse of the number of neighbors assigned to each site). The CAR model owes its name to the fact that this joint distribution corresponds to a set of conditional distributions that relate the expected value of each observation to a function of neighboring values, i.e., the Markov condition holds:

$$E(y_i | y_1, y_2, \dots, y_{i-1}, y_{i+1}, \dots, y_n) = \mu_i + \rho \sum_{j=1}^{n} c_{i,j}(y_j - \mu_j),$$

where entries of $c_{i,j}$ are non-zero only if $j \in N(i)$ and $N(i)$ indexes the sites that are neighbors of the i^{th} site.

With the Gaussian probability distribution,

$$y_i | y_j : j \neq i \sim Gauss(\mu_i + \rho \sum_{j=1}^{n} c_{i,j}(y_j - \mu_j), \tau_i^2)$$

where τ_i is a scale parameter and μ_i may contain covariates or simply the intercept.

The covariance matrix of the CAR model contains two parameters: ρ (`car_rho`) which controls the kind (positive or negative) and degree of spatial autocorrelation, and the scale parameter τ (`car_scale`). The range of permissible values for ρ depends on the specification of $\boldsymbol{C}$ and $\boldsymbol{M}$; for specification options, see prep_car_data and Cressie and Wikle (2011, pp. 184-188) or Donegan (2021).

Further details of the models and results depend on the `family` argument, as well as on the particular CAR specification chosen (from prep_car_data).

Auto-Gaussian:

When `family = auto_gaussian()` (the default), the CAR model is applied directly to the data as follows:

$$y \sim Gauss(\mu, (I - \rho C)^{-1} M),$$

where μ is the mean vector (with intercept, covariates, etc.), C is a spatial connectivity matrix, and M is a known diagonal matrix containing the conditional variances τ_i^2. C and M are provided by prep_car_data.

The auto-Gaussian model contains an implicit spatial trend (i.e. autocorrelation) component ϕ which can be calculated as follows (Cressie 2015, p. 564):

$$\phi = \rho C(y - \mu).$$

This term can be extracted from a fitted auto-Gaussian model using the spatial method.

When applied to a fitted auto-Gaussian model, the residuals.geostan_fit method returns 'de-trended' residuals R by default. That is,

$$R = y - \mu - \rho C(y - \mu).$$

To obtain "raw" residuals ($y - \mu$), use `residuals(fit, detrend = FALSE)`. Similarly, the fitted values obtained from the fitted.geostan_fit will include the spatial trend term by default.

Poisson:

For `family = poisson()`, the model is specified as:

$$y \sim Poisson(e^{O+\lambda})$$

$$\lambda \sim Gauss(\mu, (I - \rho C)^{-1} \boldsymbol{M}).$$

If the raw outcome consists of a rate $\frac{y}{p}$ with observed counts y and denominator p (often this will be the size of the population at risk), then the offset term $O = log(p)$ is the log of the denominator.

This is often written (equivalently) as:

$$y \sim Poisson(e^{O+\mu+\phi})$$

$$\phi \sim Gauss(0, (I - \rho C)^{-1} \boldsymbol{M}).$$

For Poisson models, the spatial method returns the parameter vector ϕ.

In the Poisson CAR model, ϕ contains a latent spatial trend as well as additional variation around it: $\phi_i = \rho \sum_{i=1}^n c_{ij} \phi_j + \epsilon_i$, where $\epsilon_i \sim Gauss(0, \tau_i^2)$. If you would like to extract the latent/implicit spatial trend from ϕ, you can do so by calculating (following Cressie 2015, p. 564):

$$\rho C \phi.$$

Binomial:

For `family = binomial()`, the model is specified as:

$$y \sim Binomial(N, \lambda)$$

$$logit(\lambda) \sim Gauss(\mu, (I - \rho C)^{-1} \boldsymbol{M}).$$

where outcome data y are counts, N is the number of trials, λ is the 'success' rate, and μ contains the intercept and possibly covariates. Note that the model formula should be structured as: `cbind(sucesses, failures) ~ x`, such that `trials = successes + failures`.

This is often written (equivalently) as:

$$y \sim Binomial(N, \lambda)$$

$$logit(\lambda) = \mu + \phi$$

$$\phi \sim Gauss(0, (I - \rho C)^{-1} \boldsymbol{M}).$$

For fitted Binomial models, the spatial method will return the parameter vector `phi`.

As is also the case for the Poisson model, ϕ contains a latent spatial trend as well as additional variation around it. If you would like to extract the latent/implicit spatial trend from ϕ, you can do so by calculating:

$$\rho C \phi.$$

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

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

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

slx The `slx` formula

re A list containing `re`, the varying intercepts (`re`) 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.

spatial A data frame with the name of the spatial component parameter (either "phi" or, for auto Gaussian models, "trend") and method ("CAR")

ME A list indicating if the object contains an ME model; if so, the user-provided ME list is also stored here.

C Spatial connectivity matrix (in sparse matrix format).

Author(s)

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

Source

Besag, Julian (1974). Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society* B36.2: 192–225.

Cressie, Noel (2015 (1993)). *Statistics for Spatial Data*. Wiley Classics, Revised Edition.

Cressie, Noel and Wikle, Christopher (2011). *Statistics for Spatio-Temporal Data*. Wiley.

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.

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

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

Examples

```
# model mortality risk
data(georgia)
C <- shape2mat(georgia, style = "B")
cp <- prep_car_data(C)

fit <- stan_car(deaths.male ~ offset(log(pop.at.risk.male)),
                car_parts = cp,
                data = georgia,
                family = poisson(),
                iter = 800, chains = 1 # for example speed only
                 )
rstan::stan_rhat(fit$stanfit)
rstan::stan_mcse(fit$stanfit)
print(fit)
sp_diag(fit, georgia)


## DCAR specification (inverse-distance based)
library(sf)
A <- shape2mat(georgia, "B")
D <- sf::st_distance(sf::st_centroid(georgia))
A <- D * A
cp <- prep_car_data(A, "DCAR", k = 1)

fit <- stan_car(deaths.male ~ offset(log(pop.at.risk.male)),
               data = georgia,
               car = cp,
               family = poisson(),
                 iter = 800, chains = 1 # for example speed only
)
print(fit)
```

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,
  ...
)
```

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.

`...` 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_glm *Generalized linear models*

Description

Fit a generalized linear model.

Usage

```
stan_glm(
  formula,
  slx,
  re,
  data,
  C,
  family = gaussian(),
  prior = NULL,
  ME = NULL,
  centerx = FALSE,
  prior_only = FALSE,
  censor_point,
  chains = 4,
  iter = 2000,
  refresh = 1000,
  keep_all = FALSE,
  slim = FALSE,
  drop = NULL,
  pars = NULL,
  control = NULL,
  ...
)
```

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` Optional spatial connectivity matrix which will be used to calculate residual spatial autocorrelation as well as any user specified `slx` terms; it will automatically be row-standardized before calculating `slx` terms. See `shape2mat`.

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

`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.

prior_only Draw samples from the prior distributions of parameters only.

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.

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 required 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 GLM 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.

Using drop = c('fitted', 'log_lik', 'alpha_re', 'x_true') is equivalent to slim = TRUE. If slim = TRUE, then drop will be ignored.

pars 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.

`...` Other arguments passed to sampling.

Details

Fit a generalized linear model using the R formula interface. Default prior distributions are designed to be weakly informative relative to the data. Much of the functionality intended for spatial models, such as the ability to add spatially lagged covariates and observational error models, are also available in `stan_glm`. All of `geostan`'s spatial models build on top of the same Stan code used in `stan_glm`.

Poisson models and disease mapping:

In spatial statistics, Poisson models are often used to calculate incidence rates (mortality rates, or disease incidence rates) for administrative areas like counties or census tracts. If y are counts of cases, and P are populations at risk, then the crude rates are y/P. The purpose is to model risk η for which crude rates are a (noisy) indicator. Our analysis should also respect the fact that the amount of information contained in the observations y/P increases with P. Hierarchical Poisson models are often used to incorporate all of this information.

For the Poisson model, y is specified as the outcome and the log of the population at risk `log(P)` needs to be provided as an offset term. For such a case, disease incidence across the collection of areas could be modeled as:

$$y \sim Poisson(e^{log(P)+\eta})$$

$$\eta = \alpha + A$$

$$A \sim Gauss(0, \tau)$$

$$\tau \sim Student(20, 0, 2)$$

where α is the mean log-risk (incidence rate) and A is a vector of (so-called) random effects, which enable partial pooling of information across observations. Covariates can be added to the model for the log-rates, such that $\eta = \alpha + X\beta + A$.

Note that the denominator for the rates is specified as a log-offset to provide a consistent, formula-line interface to the model. Using the log-offest (as above) is equivalent to the following:

$$y \sim Poisson(P * e^{\eta})$$

where P is still the population at risk and it is multiplied by e^{η}, the incidence rate (risk).

Spatially lagged covariates (SLX):

The `slx` argument is a convenience function for including SLX terms. For example,

$$y = WX\gamma + X\beta + \epsilon$$

where W is a row-standardized spatial weights matrix (see `shape2mat`), WX is the mean neighboring value of X, and γ is a coefficient vector. This specifies a regression with spatially lagged covariates. SLX terms can specified by providing a formula to the `slx` argument:

```
stan_glm(y ~ x1 + x2, slx = ~ x1 + x2, \...),
```

which is a shortcut for

```
stan_glm(y ~ I(W \%*\% x1) + I(W \%*\% x2) + x1 + x2, \...)
```

SLX terms will always be *prepended* to the design matrix, as above, which is important to know when setting prior distributions for regression coefficients.

For measurement error (ME) models, the SLX argument is the only way to include spatially lagged covariates since the SLX term needs to be re-calculated on each iteration of the MCMC algorithm.

Measurement error (ME) models:

The ME models are designed for surveys with spatial sampling designs, such as the American Community Survey (ACS) estimates. For a tutorial, see `vignette("spatial-me-models", package = "geostan")`.

Given estimates x, their standard errors s, and the target quantity of interest (i.e., the unknown true value) z, the ME models have one of the the following two specifications, depending on the user input. If a spatial CAR model is specified, then:

$$x \sim Gauss(z, s^2)$$
$$z \sim Gauss(\mu_z, \Sigma_z)$$
$$\Sigma_z = (I - \rho C)^{-1} M$$
$$\mu_z \sim Gauss(0, 100)$$
$$\tau_z \sim Student(10, 0, 40), \tau > 0$$
$$\rho_z \sim uniform(l, u)$$

where Σ specifies the covariance matrix of a spatial conditional autoregressive (CAR) model with scale parameter τ (on the diagonal of M), autocorrelation parameter ρ, and l, u are the lower and upper bounds that ρ is permitted to take (which is determined by the extreme eigenvalues of the spatial connectivity matrix C). M contains the inverse of the row sums of C on its diagonal multiplied by τ (following the "WCAR" specification).

For non-spatial ME models, the following is used instead:

$$x \sim Gauss(z, s^2)$$
$$z \sim student_t(\nu_z, \mu_z, \sigma_z)$$
$$\nu_z \sim gamma(3, 0.2)$$
$$\mu_z \sim Gauss(0, 100)$$
$$\sigma_z \sim student(10, 0, 40)$$

For strongly skewed variables, such as census tract poverty rates, it can be advantageous to apply a logit transformation to z before applying the CAR or Student-t prior model. When the `logit` argument is used, the first two lines of the model specification become:

$$x \sim Gauss(z, s^2)$$
$$logit(z) \sim Gauss(\mu_z, \Sigma_z)$$

and similarly for the Student t model:

$$x \sim Gauss(z, s^2)$$
$$logit(z) \sim student(\nu_z, \mu_z, \sigma_z)$$

Missing data:

For most geostan models, missing (NA) observations are allowed in the outcome variable. However, there cannot be any missing covariate data. Models that can handle missing data are: any Poisson or binomial model (GLM, SAR, CAR, ESF, ICAR), all GLMs and ESF models. The only models that cannot handle missing outcome data are the CAR and SAR models when the outcome is a continuous variable (auto-normal/Gaussian models).

When observations are missing, they will simply be ignored when calculating the likelihood in the MCMC sampling process (reflecting the absence of information). The estimated model parameters (including any covariates and spatial trend) will then be used to produce estimates or fitted values for the missing observations. The `fitted` and `posterior_predict` functions will work as normal in this case, and return values for all rows in your data.

Censored counts:

Vital statistics systems and disease surveillance programs typically suppress case counts when they are smaller than a specific threshold value. In such cases, the observation of a censored count is not the same as a missing value; instead, you are informed that the value is an integer somewhere between zero and the threshold value. For Poisson models (`family = poisson()`)), you can use the `censor_point` argument to encode this information into your model.

Internally, `geostan` will keep the index values of each censored observation, and the index value of each of the fully observed outcome values. For all observed counts, the likelihood statement will be:

$$p(y_i|data, model) = poisson(y_i|\mu_i),$$

as usual, where μ_i may include whatever spatial terms are present in the model.

For each censored count, the likelihood statement will equal the cumulative Poisson distribution function for values zero through the censor point:

$$p(y_i|data, model) = \sum_{m=0}^{M} Poisson(m|\mu_i),$$

where M is the censor point and μ_i again is the fitted value for the i^{th} observation.

For example, the US Centers for Disease Control and Prevention's CDC WONDER database censors all death counts between 0 and 9. To model CDC WONDER mortality data, you could provide `censor_point = 9` and then the likelihood statement for censored counts would equal the summation of the Poisson probability mass function over each integer ranging from zero through 9 (inclusive), conditional on the fitted values (i.e., all model parameters). See Donegan (2021) for additional discussion, references, and Stan code.

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

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

C The spatial weights matrix, if one was provided by the user.

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 NA, slot is maintained for use in `geostan_fit` methods.

Author(s)

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

Source

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`.

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

Examples

```
data(sentencing)

sentencing$log_e <- log(sentencing$expected_sents)
fit.pois <- stan_glm(sents ~ offset(log_e),
                     re = ~ name,
                     family = poisson(),
                     data = sentencing,
                    chains = 2, iter = 800) # for speed only

# MCMC diagnostics plot: Rhat values should all by very near 1
rstan::stan_rhat(fit.pois$stanfit)

# effective sample size for all parameters and generated quantities
# (including residuals, predicted values, etc.)
rstan::stan_ess(fit.pois$stanfit)

# or for a particular parameter
rstan::stan_ess(fit.pois$stanfit, "alpha_re")

# Spatial autocorrelation/residual diagnostics
```

```
sp_diag(fit.pois, sentencing)

## Posterior predictive distribution
yrep <- posterior_predict(fit.pois, S = 65)
y <- sentencing$sents
plot(density(yrep[1,]))
for (i in 2:nrow(yrep)) lines(density(yrep[i,]), col = "gray30")
lines(density(sentencing$sents), col = "darkred", lwd = 2)
```

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,
  ...
)
```

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.

`...` 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, Mitzi (2017). Spatial Models in Stan: Intrinsic Auto-Regressive Models for Areal Data. <ht
#'
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, 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")
  )
```

stan_sar *Simultaneous autoregressive (SAR) models*

Description

Fit data to an spatial Gaussian SAR (spatial error) model, or model a vector of spatially-autocorrelated parameters using a SAR prior model.

Usage

```
stan_sar(
  formula,
  slx,
  re,
  data,
  C,
  sar_parts = prep_sar_data(C),
  family = auto_gaussian(),
  prior = NULL,
  ME = NULL,
  centerx = FALSE,
  prior_only = FALSE,
  censor_point,
  chains = 4,
  iter = 2000,
  refresh = 500,
  keep_all = FALSE,
  pars = NULL,
  slim = FALSE,
  drop = NULL,
  control = NULL,
  ...
)
```

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).
```

With the SAR model, any `alpha_re` term should be at a *different* level or scale than the observations; that is, at a different scale than the autocorrelation structure of the SAR model itself.

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

`C` Spatial weights matrix (conventionally referred to as W in the SAR model). Typically, this will be created using `geostan::shape2mat(shape, style = "W")`.

This will be passed internally to prep_sar_data, and will also be used to calculate residual spatial autocorrelation as well as any user specified slx terms; it will automatically be row-standardized before calculating slx terms. See shape2mat.

sar_parts
Optional. If not provided, then prep_sar_data will be used automatically to create sar_parts using the user-provided spatial weights matrix.

family
The likelihood function for the outcome variable. Current options are auto_gaussian(), binomial(link = "logit"), and poisson(link = "log"); if family = gaussian() is provided, it will automatically be converted to auto_gaussian().

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.

sar_scale Scale parameter for the SAR model, sar_scale. The scale is assigned a Student's t prior model (constrained to be positive).

sar_rho The spatial autocorrelation parameter in the SAR model, rho, is assigned a uniform prior distribution. By default, the prior will be uniform over all permissible values as determined by the eigenvalues of the spatial weights matrix. The range of permissible values for rho is printed to the console by prep_sar_data.

tau The scale parameter for any varying intercepts (a.k.a exchangeable random effects, or partial pooling) terms. This scale parameter, tau, is assigned a Student's t prior (constrained to be positive).

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.

prior_only
Logical value; if TRUE, draw samples only from the prior distributions of parameters.

censor_point
Integer value indicating the maximum censored value; this argument is for modeling censored (suppressed) outcome data, typically disease case counts or deaths.

chains
Number of MCMC chains to use.

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.

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

`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 SAR 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.

log_lambda_mu Linear predictor inside the SAR model (for Poisson and binomial models)

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

Using `drop = c('fitted', 'log_lik', 'alpha_re', 'x_true')` is equivalent to `slim = TRUE`. Note that if `slim = TRUE`, then `drop` will be ignored—so only use one or the other.

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

`...` Other arguments passed to `sampling`.

Details

Discussions of SAR models may be found in Cliff and Ord (1981), Cressie (2015, Ch. 6), LeSage and Pace (2009), and LeSage (2014). The Stan implementation draws from Donegan (2021).

The general scheme of the SAR model for numeric vector y is

$$y = \mu + (I - \rho W)^{-1} \epsilon$$

$$\epsilon \sim Gauss(0, \sigma^2 I)$$

where W is the spatial weights matrix, I is the n-by-n identity matrix, and ρ is a spatial autocorrelation parameter. In words, the errors of the regression equation are spatially autocorrelated.

Re-arranging terms, the model can also be written as follows:

$$y = \mu + \rho W (y - \mu) + \epsilon$$

which perhaps shows more intuitively the implicit spatial trend component, $\rho W(y-\mu)$.

Most often, this model is applied directly to observations (referred to below as the auto-Gaussian model). The SAR model can also be applied to a vector of parameters inside a hierarchical model. The latter enables spatial autocorrelation to be modeled when the observations are discrete counts (e.g., disease incidence data).

A note on terminology: the spatial statistics literature conceptualizes the simultaneously-specified spatial autoregressive model (SAR) in relation to the conditionally-specified spatial autoregressive model (CAR) (see stan_car) (see Cliff and Ord 1981). The spatial econometrics literature, by contrast, refers to the simultaneously-specified spatial autoregressive (SAR) model as the spatial error model (SEM), and they contrast the SEM with the spatial lag model (which contains a spatially-lagged dependent variable on the right-hand-side of the regression equation) (see LeSage 2014).

Auto-Gaussian:

When `family = auto_gaussian()`, the SAR model is specified as follows:

$$y \sim Gauss(\mu, \Sigma)$$
$$\Sigma = \sigma^2 (I - \rho W)^{-1}(I - \rho W')^{-1}$$

where μ is the mean vector (with intercept, covariates, etc.), W is a spatial weights matrix (usually row-standardized), and σ is a scale parameter.

The SAR model contains an implicit spatial trend (i.e., spatial autocorrelation) component ϕ which is calculated as follows:

$$\phi = \rho W (y - \mu)$$

This term can be extracted from a fitted auto-Gaussian model using the spatial method.

When applied to a fitted auto-Gaussian model, the residuals.geostan_fit method returns 'de-trended' residuals R by default. That is,

$$R = y - \mu - \rho W (y - \mu).$$

To obtain "raw" residuals $(y - \mu)$, use `residuals(fit, detrend = FALSE)`. Similarly, the fitted values obtained from the fitted.geostan_fit will include the spatial trend term by default.

Poisson:

For `family = poisson()`, the model is specified as:

$$y \sim Poisson(e^{O+\lambda})$$
$$\lambda \sim Gauss(\mu, \Sigma)$$
$$\Sigma = \sigma^2 (I - \rho W)^{-1}(I - \rho W')^{-1}.$$

If the raw outcome consists of a rate $\frac{y}{p}$ with observed counts y and denominator p (often this will be the size of the population at risk), then the offset term $O = log(p)$ is the log of the denominator.

This is often written (equivalently) as:

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

$$\Sigma = \sigma^2 (I - \rho W)^{-1} (I - \rho W')^{-1}.$$

For Poisson models, the spatial method returns the parameter vector ϕ.

In the Poisson SAR model, ϕ contains a latent spatial trend as well as additional variation around it. If you would like to extract the latent/implicit spatial trend from ϕ, you can do so by calculating:

$$\rho W \phi.$$

Binomial:

For `family = binomial()`, the model is specified as:

$$y \sim Binomial(N, \lambda)$$
$$logit(\lambda) \sim Gauss(\mu, \Sigma)$$
$$\Sigma = \sigma^2 (I - \rho W)^{-1} (I - \rho W')^{-1}.$$

where outcome data y are counts, N is the number of trials, and λ is the rate of 'success'. Note that the model formula should be structured as: `cbind(sucesses, failures) ~ 1` (for an intercept-only model), such that `trials = successes + failures`.

For fitted Binomial models, the `spatial` method will return the parameter vector `phi`, equivalent to:

$$\phi = logit(\lambda) - \mu.$$

As is also the case for the Poisson model, ϕ contains a latent spatial trend as well as additional variation around it. If you would like to extract the latent/implicit spatial trend from ϕ, you can do so by calculating:

$$\rho W \phi.$$

Additional functionality:

The SAR 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

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

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

slx The `slx` formula

re A list containing `re`, the varying intercepts (`re`) 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.

spatial A data frame with the name of the spatial component parameter (either "phi" or, for auto Gaussian models, "trend") and method ("SAR")

ME A list indicating if the object contains an ME model; if so, the user-provided ME list is also stored here.

C Spatial weights matrix (in sparse matrix format).

Author(s)

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

Source

Cliff, A D and Ord, J K (1981). *Spatial Processes: Models and Applications*. Pion.

Cressie, Noel (2015 (1993)). *Statistics for Spatial Data*. Wiley Classics, Revised Edition.

Cressie, Noel and Wikle, Christopher (2011). *Statistics for Spatio-Temporal Data*. Wiley.

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

LeSage, James (2014). What Regional Scientists Need to Know about Spatial Econometrics. *The Review of Regional Science* 44: 13-32 (2014 Southern Regional Science Association Fellows Address).

LeSage, James, & Pace, Robert Kelley (2009). *Introduction to Spatial Econometrics*. Chapman and Hall/CRC.

Examples

```
# model mortality risk
data(georgia)
W <- shape2mat(georgia, style = "W")

fit <- stan_sar(log(rate.male) ~ 1,
                C = W,
                data = georgia,
                chains = 1, # for ex. speed only
                iter = 700
                )

rstan::stan_rhat(fit$stanfit)
rstan::stan_mcse(fit$stanfit)
print(fit)
plot(fit)
sp_diag(fit, georgia)
```

```
 # a more appropriate model for count data:
fit2 <- stan_sar(deaths.male ~ offset(log(pop.at.risk.male)),
                C = W,
                data = georgia,
                family = poisson(),
                chains = 1, # for ex. speed only
                iter = 700
                 )
sp_diag(fit2, georgia)
```

---

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