

Package: spStack (via r-universe)

October 12, 2026

Type Package

Version 1.1.3.99

Title Bayesian Geostatistics Using Predictive Stacking

Description Fits Bayesian hierarchical spatial and spatial-temporal process models for point-referenced Gaussian, Poisson, binomial, and binary data using stacking of predictive densities. It involves sampling from analytically available posterior distributions conditional upon candidate values of the spatial process parameters and, subsequently assimilate inference from these individual posterior distributions using Bayesian predictive stacking. Our algorithm is highly parallelizable and hence, much faster than traditional Markov chain Monte Carlo algorithms while delivering competitive predictive performance. See Zhang, Tang, and Banerjee (2025) <[doi:10.1080/01621459.2025.2566449](https://doi.org/10.1080/01621459.2025.2566449)>, and, Pan, Zhang, Bradley, and Banerjee (2025) <[doi:10.1214/25-BA1582](https://doi.org/10.1214/25-BA1582)> for details.

Imports CVXR, future, future.apply, ggplot2, MBA, rlang, rstudioapi

NeedsCompilation yes

License GPL-3

Encoding UTF-8

LazyData true

Suggests dplyr, knitr, loo, patchwork, rmarkdown, spelling, testthat
(>= 3.0.0), tidy

Config/testthat/edition 3

Roxygen list(markdown = TRUE)

Depends R (>= 4.4)

VignetteBuilder knitr

URL <https://span-18.github.io/spStack-dev/>

BugReports <https://github.com/SPan-18/spStack-dev/issues>

Language en-US

Config/roxygen2/version 8.1.0

Config/pak/sysreqs cmake libgmp3-dev make pkg-config libclang-dev

Repository <https://span-18.r-universe.dev>

Date/Publication 2026-10-12 22:34:40 UTC

RemoteUrl <https://github.com/span-18/spstack-dev>

RemoteRef HEAD

RemoteSha eaffc5c69971cb223864b53566685b9d2154fee4

Contents

spStack-package	2
candidateModels	4
cholUpdate	5
get_stacking_weights	7
iDist	8
posteriorPredict	9
recoverGLMscale	11
simSpaceTime	12
simSpatial	13
spGLMexact	15
spGLMstack	19
spLMexact	22
spLMstack	25
stackedSampler	29
stvcGLMexact	30
stvcGLMstack	34
stvcLMexact	37
stvcLMstack	41
surfaceplot	44
surfaceplot2	46
svcGLMexact	47
svcGLMstack	50
svcLMexact	53
svcLMstack	56
Index	60

Description

This package delivers functions to fit Bayesian hierarchical spatial process models for point-referenced Gaussian, Poisson, binomial, and binary data using stacking of predictive densities. It involves sampling from analytically available posterior distributions conditional upon some candidate values of the spatial process parameters for both Gaussian response model as well as non-Gaussian responses, and, subsequently assimilate inference from these individual posterior distributions using Bayesian predictive stacking. Our algorithm is highly parallelizable and hence, much faster than traditional Markov chain Monte Carlo algorithms while delivering competitive predictive performance.

In context of inference for spatial point-referenced data, Bayesian hierarchical models involve latent spatial processes characterized by spatial process parameters, which besides lacking substantive relevance in scientific contexts, are also weakly identified and hence, impedes convergence of MCMC algorithms. This motivates us to build methodology that involves fast sampling from posterior distributions conditioned on a grid of the weakly identified model parameters and combine the inference by stacking of predictive densities (Yao *et. al* 2018). We exploit the Bayesian conjugate linear modeling framework for the Gaussian case (Zhang, Tang and Banerjee 2025) and the generalized conjugate multivariate distribution theory (Pan, Zhang, Bradley and Banerjee 2025) to analytically derive the individual posterior distributions.

Details

Package: spStack
 Type: Package
 Version: 1.1.0
 License: GPL-3

Accepts a formula, e.g., $y \sim x_1 + x_2$, for most regression models accompanied by candidate values of spatial process parameters, and returns posterior samples of the regression coefficients and the latent spatial random effects. Posterior inference or prediction of any quantity of interest proceed from these samples. Main functions are -

`spLMexact()`
`spGLMexact()`
`spLMstack()`
`spGLMstack()`

Author(s)

Maintainer: Soumyakanti Pan <span18@ucla.edu> ([ORCID](#))

Authors:

- Soumyakanti Pan <span18@ucla.edu> ([ORCID](#))
- Sudipto Banerjee <sudipto@ucla.edu>

References

Zhang L, Tang W, Banerjee S (2025). "Bayesian Geostatistics Using Predictive Stacking." *Journal of the American Statistical Association*, **In press**. doi:[10.1080/01621459.2025.2566449](https://doi.org/10.1080/01621459.2025.2566449).

Pan S, Zhang L, Bradley JR, Banerjee S (2025). "Bayesian Inference for Spatial-temporal Non-Gaussian Data Using Predictive Stacking." *Bayesian Analysis*, **In Press**. doi:10.1214/25BA1582.

Yao Y, Vehtari A, Simpson D, Gelman A (2018). "Using Stacking to Average Bayesian Predictive Distributions (with Discussion)." *Bayesian Analysis*, **13**(3), 917-1007. doi:10.1214/17BA1091.

See Also

Useful links:

- <https://span-18.github.io/spStack-dev/>
- Report bugs at <https://github.com/SPan-18/spStack-dev/issues>

candidateModels

Create a collection of candidate models for stacking

Description

Creates an object of class 'candidateModels' that contains a list of candidate models for stacking. The function takes a list of candidate values for each model parameter and returns a list of possible combinations of these values based on either simple aggregation or Cartesian product of individual candidate values.

Usage

```
candidateModels(params_list, aggregation = "simple")
```

Arguments

params_list	a list of candidate values for each model parameter. See examples for details.
aggregation	a character string specifying the type of aggregation to be used. Options are 'simple' and 'cartesian'. Default is 'simple'.

Value

an object of class 'candidateModels'

Author(s)

Soumyakanti Pan span18@ucla.edu,
Sudipto Banerjee sudipto@ucla.edu

See Also

[stvcGLMstack\(\)](#)

Examples

```

m1 <- candidateModels(list(phi_s = c(1, 1), phi_t = c(1, 2)), "simple")
m1
m2 <- candidateModels(list(phi_s = c(1, 1), phi_t = c(1, 2)), "cartesian")
m2
m3 <- candidateModels(list(phi_s = list(c(1, 1), c(1, 2)),
                             phi_t = list(c(1, 3), c(2, 3)),
                             boundary = c(0.5, 0.75)),
                      "simple")

```

cholUpdate

*Different Cholesky factor updates***Description**

Provides functions that implements different types of updates of a Cholesky factor that includes rank-one update, single row/column deletion update and a block deletion update.

Usage

```

cholUpdateRankOne(A, v, alpha, beta, lower = TRUE)

cholUpdateDel(A, del.index, lower = TRUE)

cholUpdateDelBlock(A, del.start, del.end, lower = TRUE)

```

Arguments

A	an $n \times n$ triangular matrix
v	an $n \times 1$ matrix/vector
alpha	scalar; if not supplied, default is 1
beta	scalar; if not supplied, default is 1
lower	logical; if A is lower-triangular or not
del.index	an integer from 1 to n indicating the row/column to be deleted
del.start	an integer from 1 to n indicating the first row/column of a block to be deleted, must be at least 1 less than del.end
del.end	an integer from 1 to n indicating the last row/column of a block to be deleted, must be at least 1 more than del.start

Details

Suppose $B = AA^\top$ is a $n \times n$ matrix with A being its lower-triangular Cholesky factor. Then rank-one update corresponds to finding the Cholesky factor of the matrix $C = \alpha B + \beta vv^\top$ for some $\alpha, \beta \in \mathbb{R}$ given A (see, Krause and Igel 2015). Similarly, single row/column deletion update corresponds to finding the Cholesky factor of the $(n-1) \times (n-1)$ matrix B_i which is obtained by removing the i -th row and column of B , given A for some $i = 1, \dots, n$. Lastly, block deletion corresponds to finding the Cholesky factor of the $(n-n_k) \times (n-n_k)$ matrix B_I for a subset I of $\{1, \dots, n\}$ containing n_k consecutive indices, given the factor A .

Value

An $m \times m$ lower-triangular matrix with $m = n$ in case of `cholUpdateRankOne()`, $m = n - 1$ in case of `cholUpdateDel()`, and, $m = n - n_k$ in case of `cholUpdateDelBlock()` where n_k is the size of the block removed.

Author(s)

Soumyakanti Pan span18@ucla.edu,
Sudipto Banerjee sudipto@ucla.edu

References

Oswin Krause and Christian Igel. 2015. "A More Efficient Rank-one Covariance Matrix Update for Evolution Strategies". In *Proceedings of the 2015 ACM Conference on Foundations of Genetic Algorithms XIII* (FOGA '15). Association for Computing Machinery, New York, NY, USA, 129-136. doi:[10.1145/2725494.2725496](https://doi.org/10.1145/2725494.2725496).

Examples

```
n <- 10
A <- matrix(rnorm(n^2), n, n)
A <- crossprod(A)
cholA <- chol(A)

## Rank-1 update
v <- 1:n
AplusvvT <- A + tcrossprod(v)
cholA1 <- t(chol(AplusvvT))
cholA2 <- cholUpdateRankOne(cholA, v, lower = FALSE)
print(all(abs(cholA1 - cholA2) < 1E-9))

## Single Row-deletion update
ind <- 2
A1 <- A[-ind, -ind]
cholA1 <- t(chol(A1))
cholA2 <- cholUpdateDel(cholA, del.index = ind, lower = FALSE)
print(all(abs(cholA1 - cholA2) < 1E-9))

## Block-deletion update
start_ind <- 2
end_ind <- 6
del_ind <- c(start_ind:end_ind)
A1 <- A[-del_ind, -del_ind]
cholA1 <- t(chol(A1))
cholA2 <- cholUpdateDelBlock(cholA, start_ind, end_ind, lower = FALSE)
print(all(abs(cholA1 - cholA2) < 1E-9))
```

get_stacking_weights *Optimal stacking weights*

Description

Obtains optimal stacking weights given leave-one-out predictive densities for each candidate model.

Usage

```
get_stacking_weights(log_loopd, solver = NULL, verbose = TRUE)
```

Arguments

log_loopd	an $n \times M$ matrix with i -th row containing the leave-one-out predictive densities for the i -th data point for the M candidate models.
solver	specifies the solver to use for obtaining optimal weights. Default is "CLARABEL". Internally calls <code>CVXR::psolve()</code> .
verbose	if TRUE, prints output of optimization routine.

Details

The weights maximize the log score of the stacked leave-one-out predictive densities (Yao *et al.* 2018) over the simplex, using the CVXR solvers in the order given above. If none of them reaches an optimal solution, `loo::stacking_weights()` is used as a fallback when the package **loo** is installed; otherwise the weights are returned as NA with status "failed".

Value

A list with elements:

weights	optimal stacking weights as a numeric vector of length M (NA if no solver succeeded, see Details).
status	solver status, returns "optimal" if solver succeeded, and "failed" if no solver succeeded.
solver	name of the solver used ("none" if no solver succeeded).
details	a list with the installed CVXR solvers (installed), the requested solver(s) (requested) and those of them not installed (missing.requested), the order in which the solvers were tried (search.order), a data frame of the attempts with the status or error of each solver (attempts), and whether the fallback <code>loo::stacking_weights()</code> was used (fallback).

Author(s)

Soumyakanti Pan span18@ucla.edu,
 Sudipto Banerjee sudipto@ucla.edu

References

Yao Y, Vehtari A, Simpson D, Gelman A (2018). "Using Stacking to Average Bayesian Predictive Distributions (with Discussion)." *Bayesian Analysis*, **13**(3), 917-1007. doi:[10.1214/17BA1091](https://doi.org/10.1214/17BA1091).

See Also

`CVXR::psolve()`, `splMstack()`, `spGLMstack()`

Examples

```
set.seed(1234)
data(simSpatial)
dat <- simSpatial[1:100, ]

cand.mod <- candidateModels(list(phi = c(2, 4, 6),
                                nu = c(1, 1.5, 1.75),
                                noise_sp_ratio = c(0.1)), "cartesian")

mod1 <- splMstack(y_gauss ~ x1 + x2, data = dat,
                 coords = as.matrix(dat[, c("s1", "s2")]),
                 cor.fn = "matern",
                 candidate.models = cand.mod,
                 n.samples = 1000, loopd.method = "exact",
                 parallel = FALSE, verbose = TRUE)

loopd_mat <- do.call('cbind', mod1$loopd)
w_hat <- get_stacking_weights(loopd_mat)
print(round(w_hat$weights, 4))
print(w_hat$solver)
print(w_hat$status)
```

iDist

Calculate distance matrix

Description

Computes the inter-site Euclidean distance matrix for one or two sets of points.

Usage

```
iDist(coords.1, coords.2, ...)
```

Arguments

<code>coords.1</code>	an $n \times p$ matrix with each row corresponding to a point in p -dimensional space.
<code>coords.2</code>	an $m \times p$ matrix with each row corresponding to a point in p dimensional space. If this is missing then <code>coords.1</code> is used.
<code>...</code>	currently no additional arguments.

Value

The $n \times n$ or $n \times m$ inter-site Euclidean distance matrix.

Author(s)

Soumyakanti Pan span18@ucla.edu,
 Sudipto Banerjee sudipto@ucla.edu

Examples

```
n <- 10
p1 <- cbind(runif(n), runif(n))
m <- 5
p2 <- cbind(runif(m), runif(m))
D <- iDist(p1, p2)
```

posteriorPredict	<i>Prediction of latent process at new spatial or temporal locations</i>
------------------	--

Description

A function to sample from the posterior predictive distribution of the latent spatial or spatial-temporal process.

Usage

```
posteriorPredict(mod_out, coords_new, covars_new, joint = FALSE, nBinom_new)
```

Arguments

mod_out	an object returned by any model fit under fixed hyperparameters or using predictive stacking, i.e., <code>spLMexact()</code> , <code>spLMstack()</code> , <code>spGLMexact()</code> , <code>spGLMstack()</code> , <code>svGLMexact()</code> , <code>svGLMstack()</code> , <code>svcGLMexact()</code> , <code>svcGLMstack()</code> , <code>stvcGLMexact()</code> , <code>stvcGLMstack()</code> , <code>stvcLMexact()</code> or <code>stvcLMstack()</code> .
coords_new	new spatial coordinates (an $n_{new} \times 2$ matrix) or, for the spatial-temporal models, a list with tags <code>sp</code> (an $n_{new} \times 2$ matrix) and <code>time</code> (a vector or one-column matrix) at which the latent process, the mean, and the response is to be predicted.
covars_new	new covariates at the new coordinates: a matrix or, for the varying coefficients models (svc and stvc), a list with tags <code>fixed</code> (covariates with fixed effects) and <code>vc</code> (covariates with spatially or spatially-temporally varying coefficients). See examples for the structure of this list.
joint	a logical value indicating whether to return the joint posterior predictive samples of the latent process at the new locations or times. Defaults to FALSE.
nBinom_new	a vector of the number of trials for each new prediction location or time. Only required if the model family is "binomial". Defaults to a vector of ones, indicating one trial for each new prediction.

Value

A modified object with the class name preceeded by the identifier `pp` separated by a `..`. For example, if input is of class `spLMstack`, then the output of this prediction function would be `pp.spLMstack`. The entry with the tag `samples` is updated and will include samples from the posterior predictive distribution of the latent process, the mean, and the response at the new locations or times. An entry with the tag `prediction` is added and contains the new coordinates and covariates, and whether the joint posterior predictive samples were requested.

Author(s)

Soumyakanti Pan span18@ucla.edu,
Sudipto Banerjee sudipto@ucla.edu

See Also

[spLMexact\(\)](#), [spLMstack\(\)](#), [spGLMexact\(\)](#), [spGLMstack\(\)](#), [svcLMexact\(\)](#), [svcLMstack\(\)](#), [svcGLMexact\(\)](#), [svcGLMstack\(\)](#), [stvcGLMexact\(\)](#), [stvcGLMstack\(\)](#), [stvcLMexact\(\)](#), [stvcLMstack\(\)](#)

Examples

```
set.seed(1234)
# training and test data sizes
n_train <- 100
n_pred <- 10

# Example 1: Spatial linear model
# split the data into training and prediction sets
data(simSpatial)
dat_train <- simSpatial[1:n_train, ]
dat_pred <- simSpatial[n_train + 1:n_pred, ]

# fit a spatial linear model using predictive stacking
cand.mod <- candidateModels(list(phi = c(2, 4, 6), nu = c(1, 1.5, 1.75),
                                noise_sp_ratio = c(0.1, 0.25)), "cartesian")
mod1 <- spLMstack(y_gauss ~ x1 + x2, data = dat_train,
                  coords = as.matrix(dat_train[, c("s1", "s2")]),
                  cor.fn = "matern",
                  candidate.models = cand.mod,
                  n.samples = 1000, loopd.method = "exact",
                  parallel = FALSE, verbose = TRUE)

# prepare new coordinates and covariates for prediction
sp_pred <- as.matrix(dat_pred[, c("s1", "s2")])
X_new <- cbind(1, dat_pred$x1, dat_pred$x2)

# carry out posterior prediction
mod.pred <- posteriorPredict(mod1, coords_new = sp_pred, covars_new = X_new,
                             joint = TRUE)

# sample from the stacked posterior and posterior predictive distribution
post_samps <- stackedSampler(mod.pred)
```

```
# compare the predicted spatial effects with the truth
z_pred <- apply(post_samps$z.pred, 1, median)
cor(z_pred, dat_pred$z_true)
```

recoverGLMscale	<i>Recover posterior samples of scale parameters of spatial/spatial-temporal generalized linear models</i>
-----------------	--

Description

A function to recover posterior samples of scale parameters that were marginalized out during model fit. This is only applicable for spatial or, spatial-temporal generalized linear models. This function applies on outputs of functions that fits a spatial/spatial-temporal generalized linear model, such as `spGLMexact()`, `spGLMstack()`, `svcGLMexact()`, `svcGLMstack()`, `stvcGLMexact()`, and `stvcGLMstack()`.

Usage

```
recoverGLMscale(mod_out)
```

Arguments

`mod_out` an object returned by a fitting a spatial or spatial-temporal GLM.

Value

An object of the same class as input, and updates the list tagged samples with the posterior samples of the scale parameters. The new tags are `sigmasq.beta` and `z.scale`.

Author(s)

Soumyakanti Pan span18@ucla.edu,
Sudipto Banerjee sudipto@ucla.edu

See Also

`spGLMexact()`, `spGLMstack()`, `svcGLMexact()`, `svcGLMstack()`, `stvcGLMexact()`, `stvcGLMstack()`

Examples

```
set.seed(1234)
data(simSpatial)
dat <- simSpatial[1:100, ]
cand.mod <- candidateModels(list(phi = c(2, 4, 8), nu = c(1.5, 1.75),
                                boundary = c(0.5)), "cartesian")

mod1 <- spGLMstack(y_pois ~ x1 + x2, data = dat, family = "poisson",
                  coords = as.matrix(dat[, c("s1", "s2")]), cor.fn = "matern",
```

```

candidate.models = cand.mod,
n.samples = 100,
loopd.controls = list(method = "CV", CV.K = 10, nMC = 500),
verbose = TRUE)

# Recover posterior samples of scale parameters
mod1.1 <- recoverGLMscales(mod1)

# sample from the stacked posterior distribution
post_samps <- stackedSampler(mod1.1)

```

simSpaceTime

Synthetic point-referenced spatial-temporal data

Description

Dataset of size 500 with spatial coordinates sampled uniformly from the unit square and temporal coordinates sampled uniformly from the unit interval, two covariates, an intercept and a slope of x_1 that vary over space and time, and Gaussian and Poisson responses that share them.

Usage

```
data(simSpaceTime)
```

Format

a data.frame object with 500 rows and columns

s_1 , s_2 2-D coordinates in the unit square.

t_coords temporal coordinates in the unit interval.

x_1 , x_2 covariates sampled from the standard normal distribution.

y_gauss Gaussian response.

y_pois Poisson count response.

$z1_true$ true spatial-temporal effect associated with the intercept.

$z2_true$ true spatial-temporal effect associated with x_1 .

Details

With $\ell = (s, t)$, the varying intercept is a wave travelling across space over time and the varying slope of x_1 changes with s_2 and t ,

$$z_1(\ell) = \sin(2\pi s_1 - \pi t), \quad z_2(\ell) = \cos(2\pi s_2) \cos(\pi t),$$

deterministic surfaces that are not draws from Gaussian processes. The responses are generated as

$$y_{gauss}(\ell) \sim N(2 + 5x_1(\ell) - x_2(\ell) + z_1(\ell) + x_1(\ell)z_2(\ell), 0.5^2),$$

$$y_{pois}(\ell) \sim \text{Poisson}(\exp\{2 - 0.5x_1(\ell) + 0.3x_2(\ell) + z_1(\ell) + x_1(\ell)z_2(\ell)\}).$$

This data can be generated with the code given in the example.

Author(s)Soumyakanti Pan span18@ucla.edu**See Also**[simSpatial](#)**Examples**

```

set.seed(1726)
n <- 500
s1 <- runif(n)
s2 <- runif(n)
t_coords <- runif(n)
x1 <- rnorm(n)
x2 <- rnorm(n)
z1 <- sin(2 * pi * s1 - pi * t_coords)
z2 <- cos(2 * pi * s2) * cos(pi * t_coords)
dat <- data.frame(
  s1 = s1, s2 = s2, t_coords = t_coords, x1 = x1, x2 = x2,
  y_gauss = rnorm(n, 2 + 5 * x1 - x2 + z1 + x1 * z2, sd = 0.5),
  y_pois = rpois(n, exp(2 - 0.5 * x1 + 0.3 * x2 + z1 + x1 * z2)),
  z1_true = z1, z2_true = z2
)
all.equal(dat, simSpaceTime)

```

simSpatial

*Synthetic point-referenced spatial data***Description**

Dataset of size 500 with spatial coordinates sampled uniformly from the unit square, two covariates, a spatial effect with a checkerboard pattern and one response for each of the Gaussian, Poisson, binomial and binary families, plus a Gaussian and a Poisson response with spatially varying coefficients (a varying intercept with a ring pattern and a varying slope of x_1). All the responses share the covariates and the spatial effects, so that every spatial model in the package can be fitted to the same data.

Usage

```
data(simSpatial)
```

Format

a `data.frame` object with 500 rows and 13 columns

`s1`, `s2` 2-D coordinates in the unit square.

`x1`, `x2` covariates sampled from the standard normal distribution.

`y_gauss` Gaussian response.
`y_pois` Poisson count response.
`y_binom` binomial count response, out of `n_trials` trials.
`n_trials` number of trials of the binomial response (5 to 20).
`y_bin` binary response.
`y_svc` Gaussian response with the spatially varying intercept `z2` and the spatially varying slope `5 + z` of `x1`.
`y_svc_pois` Poisson count response with the spatially varying intercept `z2` and the spatially varying slope `-0.5 + z` of `x1`.
`z_true` true spatial effect (checkerboard) that generated the data; the spatially varying part of the slope of `x1` in `y_svc` and `y_svc_pois`.
`z2_true` true spatially varying intercept (rings) of `y_svc` and `y_svc_pois`.

Details

The spatial effect is the checkerboard

$$z(s) = 1.5 \sin(2\pi s_1) \sin(2\pi s_2),$$

and the spatially varying intercept is the rings

$$z_2(s) = \cos(4\pi \|s - s_0\|) - c, \quad s_0 = (0.5, 0.5),$$

where the constant $c \approx 0.129$ makes the mean of z_2 over the unit square zero (the mean of z is zero by symmetry, and the two surfaces are orthogonal on the unit square). Both are deterministic surfaces, not draws from a Gaussian process. The responses are generated as

$$\begin{aligned}
 y_{\text{gauss}}(s) &\sim N(2 + 5x_1(s) - x_2(s) + z(s), 0.5^2), \\
 y_{\text{pois}}(s) &\sim \text{Poisson}(\exp\{2 - 0.5x_1(s) + 0.3x_2(s) + z(s)\}), \\
 y_{\text{binom}}(s) &\sim \text{Binomial}(m(s), \text{ilogit}\{0.5 - 0.5x_1(s) + 0.5x_2(s) + z(s)\}), \\
 y_{\text{bin}}(s) &\sim \text{Bernoulli}(\text{ilogit}\{0.5 - 0.5x_1(s) + 0.5x_2(s) + z(s)\}), \\
 y_{\text{svc}}(s) &\sim N(2 + z_2(s) + \{5 + z(s)\}x_1(s) - x_2(s), 0.5^2), \\
 y_{\text{svc_pois}}(s) &\sim \text{Poisson}(\exp\{2 + z_2(s) + \{-0.5 + z(s)\}x_1(s) + 0.3x_2(s)\}),
 \end{aligned}$$

where $m(s)$ is the number of trials. This data can be generated with the code given in the example.

Author(s)

Soumyakanti Pan span18@ucla.edu

See Also

[simSpaceTime](#)

Examples

```

set.seed(1729)
n <- 500
s1 <- runif(n)
s2 <- runif(n)
x1 <- rnorm(n)
x2 <- rnorm(n)
z <- 1.5 * sin(2 * pi * s1) * sin(2 * pi * s2)
ring <- function(s1, s2) cos(4 * pi * sqrt((s1 - 0.5)^2 + (s2 - 0.5)^2))
g <- (seq_len(1000) - 0.5) / 1000
z2 <- ring(s1, s2) - mean(outer(g, g, ring))
n_trials <- sample(5:20, n, replace = TRUE)
dat <- data.frame(
  s1 = s1, s2 = s2, x1 = x1, x2 = x2,
  y_gauss = rnorm(n, 2 + 5 * x1 - x2 + z, sd = 0.5),
  y_pois = rpois(n, exp(2 - 0.5 * x1 + 0.3 * x2 + z)),
  y_binom = rbinom(n, n_trials, plogis(0.5 - 0.5 * x1 + 0.5 * x2 + z)),
  n_trials = n_trials,
  y_bin = rbinom(n, 1, plogis(0.5 - 0.5 * x1 + 0.5 * x2 + z)),
  y_svc = rnorm(n, 2 + z2 + (5 + z) * x1 - x2, sd = 0.5),
  y_svc_pois = rpois(n, exp(2 + z2 + (-0.5 + z) * x1 + 0.3 * x2)),
  z_true = z,
  z2_true = z2
)
all.equal(dat, simSpatial)

```

spGLMexact

Univariate Bayesian spatial generalized linear model

Description

Fits a Bayesian spatial generalized linear model with fixed values of spatial process parameters and some auxiliary model parameters. The output contains posterior samples of the fixed effects, spatial random effects and, if required, finds leave-one-out predictive densities.

Usage

```

spGLMexact(
  formula,
  data = parent.frame(),
  family,
  coords,
  cor.fn,
  priors,
  spParams,
  boundary = 0.5,
  n.samples,
  loopd = FALSE,

```

```

    loopd.method = "exact",
    CV.K = 10,
    loopd.nMC = 500,
    verbose = TRUE,
    ...
)

```

Arguments

formula	a symbolic description of the regression model to be fit. See example below.
data	an optional data frame containing the variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> , typically the environment from which <code>spGLMexact</code> is called.
family	Specifies the distribution of the response as a member of the exponential family. Supported options are 'poisson', 'binomial' and 'binary'.
coords	an $n \times 2$ matrix of the observation coordinates in $\mathbb{R}^2$ (e.g., easting and northing).
cor.fn	a quoted keyword that specifies the correlation function used to model the spatial dependence structure among the observations. Supported covariance model key words are: 'exponential' and 'matern'. See below for details.
priors	(optional) a list with each tag corresponding to a hyperparameter name and containing hyperprior details. Valid tags include <code>V.beta</code> , <code>nu.beta</code> , <code>nu.z</code> and <code>sigmaSq.xi</code> . Values of <code>nu.beta</code> and <code>nu.z</code> must be at least 2.1. If not supplied, uses defaults.
spParams	fixed values of spatial process parameters.
boundary	Specifies the boundary adjustment parameter. Must be a real number between 0 and 1. Default is 0.5.
n.samples	number of posterior samples to be generated.
loopd	logical. If <code>loopd=TRUE</code> , returns leave-one-out predictive densities, using method as given by <code>loopd.method</code> . Default is <code>FALSE</code> .
loopd.method	character. Ignored if <code>loopd=FALSE</code> . If <code>loopd=TRUE</code> , valid inputs are 'exact', 'CV' and 'PSIS'. The option 'exact' corresponds to exact leave-one-out predictive densities which requires computation almost equivalent to fitting the model n times. The options 'CV' and 'PSIS' are faster and they implement K -fold cross validation and Pareto-smoothed importance sampling to find approximate leave-one-out predictive densities (Vehtari <i>et al.</i> 2017).
CV.K	An integer between 10 and 20. Considered only if <code>loopd.method='CV'</code> . Default is 10 (as recommended in Vehtari <i>et. al</i> 2017).
loopd.nMC	Number of Monte Carlo samples to be used to evaluate leave-one-out predictive densities when <code>loopd.method</code> is set to either 'exact' or 'CV'.
verbose	logical. If <code>verbose = TRUE</code> , prints model description.
...	currently no additional argument.

Details

With this function, we fit a Bayesian hierarchical spatial generalized linear model by sampling exactly from the joint posterior distribution utilizing the generalized conjugate multivariate distribution theory (Bradley and Clinch 2024). Suppose $\chi = (s_1, \dots, s_n)$ denotes the n spatial locations the response y is observed. Let $y(s)$ be the outcome at location s endowed with a probability law from the natural exponential family, which we denote by

$$y(s) \sim \text{EF}(x(s)^\top \beta + z(s); b, \psi)$$

for some positive parameter $b > 0$ and unit log partition function ψ . We consider the following response models based on the input supplied to the argument family.

'poisson' It considers point-referenced Poisson responses $y(s) \sim \text{Poisson}(e^{x(s)^\top \beta + z(s)})$. Here, $b = 1$ and $\psi(t) = e^t$.

'binomial' It considers point-referenced binomial counts $y(s) \sim \text{Binomial}(m(s), \pi(s))$ where, $m(s)$ denotes the total number of trials and probability of success $\pi(s) = \text{ilogit}(x(s)^\top \beta + z(s))$ at location s . Here, $b = m(s)$ and $\psi(t) = \log(1 + e^t)$.

'binary' It considers point-referenced binary data (0 or, 1) i.e., $y(s) \sim \text{Bernoulli}(\pi(s))$, where probability of success $\pi(s) = \text{ilogit}(x(s)^\top \beta + z(s))$ at location s . Here, $b = 1$ and $\psi(t) = \log(1 + e^t)$.

The hierarchical model is given as

$$\begin{aligned} y(s_i) \mid \beta, z, \xi &\sim \text{EF}(x(s_i)^\top \beta + z(s_i) + \xi_i - \mu_i; b_i, \psi_y), i = 1, \dots, n \\ \xi \mid \beta, z, \sigma_\xi^2, \alpha_\epsilon &\sim \text{GCM}_c(\dots), \\ \beta \mid \sigma_\beta^2 &\sim N(0, \sigma_\beta^2 V_\beta), \quad \sigma_\beta^2 \sim \text{IG}(\nu_\beta/2, \nu_\beta/2) \\ z \mid \sigma_z^2 &\sim N(0, \sigma_z^2 R(\chi; \phi, \nu)), \quad \sigma_z^2 \sim \text{IG}(\nu_z/2, \nu_z/2), \end{aligned}$$

where $\mu = (\mu_1, \dots, \mu_n)^\top$ denotes the discrepancy parameter. We fix the spatial process parameters ϕ and ν and the hyperparameters $V_\beta, \nu_\beta, \nu_z$ and σ_ξ^2 . The term ξ is known as the fine-scale variation term which is given a conditional generalized conjugate multivariate distribution as prior. For more details, see Pan *et al.* 2025. Default values for $V_\beta, \nu_\beta, \nu_z, \sigma_\xi^2$ are diagonal with each diagonal element 100, 2.1, 2.1 and 0.1 respectively.

Value

An object of class spGLMexact, which is a list with the following tags -

priors details of the priors used, containing the values of the boundary adjustment parameter (boundary), the variance parameter of the fine-scale variation term (simasq.xi) and others.

samples a list of length 3, containing posterior samples of fixed effects (beta), spatial effects (z) and the fine-scale variation term (xi).

loopd If loopd=TRUE, contains leave-one-out predictive densities.

model.params Values of the fixed parameters that includes phi (spatial decay), nu (spatial smoothness; NA for the exponential correlation function).

diagnostics a list of diagnostics. Element `numerical` is a data frame with one row and columns `min.pivot` (the smallest relative Cholesky pivot of the $n \times n$ correlation matrix; values below $1e-8$ indicate a nearly singular correlation matrix), `min.cor` and `max.cor` (the correlations of the two farthest-apart and of the two closest locations; values of `min.cor` above 0.95 suggest an effective range far exceeding the extent of the data, values of `max.cor` below 0.05 suggest uncorrelated locations), obtained from quantities the fit computes anyway. If `verbose = TRUE`, a "Diagnostics" section is printed when any threshold is crossed.

The return object might include additional data that can be used for subsequent prediction and/or model fit evaluation.

Author(s)

Soumyakanti Pan span18@ucla.edu

References

Bradley JR, Clinch M (2024). "Generating Independent Replicates Directly from the Posterior Distribution for a Class of Spatial Hierarchical Models." *Journal of Computational and Graphical Statistics*, **0**(0), 1-17. doi:[10.1080/10618600.2024.2365728](https://doi.org/10.1080/10618600.2024.2365728).

Pan S, Zhang L, Bradley JR, Banerjee S (2025). "Bayesian Inference for Spatial-temporal Non-Gaussian Data Using Predictive Stacking." *Bayesian Analysis*, **In Press**. doi:[10.1214/25BA1582](https://doi.org/10.1214/25BA1582).

Vehtari A, Gelman A, Gabry J (2017). "Practical Bayesian Model Evaluation Using Leave-One-out Cross-Validation and WAIC." *Statistics and Computing*, **27**(5), 1413-1432. ISSN 0960-3174. doi:[10.1007/s1122201696964](https://doi.org/10.1007/s1122201696964).

See Also

[spLMexact\(\)](#)

Examples

```
data(simSpatial)
dat <- simSpatial[1:100, ]
coords <- as.matrix(dat[, c("s1", "s2")])

# Example 1: spatial Poisson count data
mod1 <- spGLMexact(y_pois ~ x1 + x2, data = dat, family = "poisson",
  coords = coords, cor.fn = "matern",
  spParams = list(phi = 4, nu = 1.75),
  n.samples = 100, verbose = TRUE)
post_beta <- mod1$samples$beta
print(t(apply(post_beta, 1, function(x) quantile(x, c(0.025, 0.5, 0.975))))))

# Example 2: spatial binomial count data
mod2 <- spGLMexact(cbind(y_binom, n_trials) ~ x1 + x2, data = dat,
  family = "binomial", coords = coords, cor.fn = "matern",
  spParams = list(phi = 4, nu = 1.75),
  n.samples = 100, verbose = FALSE)

# Example 3: spatial binary data
```

```
mod3 <- spGLMexact(y_bin ~ x1 + x2, data = dat, family = "binary",
  coords = coords, cor.fn = "matern",
  spParams = list(phi = 4, nu = 1.75),
  n.samples = 100, verbose = FALSE)
```

spGLMstack

Bayesian spatial generalized linear model using predictive stacking

Description

Fits Bayesian spatial generalized linear model on a collection of candidate models constructed based on some candidate values of some model parameters specified by the user and subsequently combines inference by stacking predictive densities. See Pan, Zhang, Bradley, and Banerjee (2025) for more details.

Usage

```
spGLMstack(
  formula,
  data = parent.frame(),
  family,
  coords,
  cor.fn,
  priors,
  candidate.models,
  n.samples,
  loopd.controls,
  parallel = FALSE,
  solver = NULL,
  verbose = TRUE,
  ...
)
```

Arguments

<code>formula</code>	a symbolic description of the regression model to be fit. See example below.
<code>data</code>	an optional data frame containing the variables in the model. If not found in <code>data</code> , the variables are taken from <code>environment(formula)</code> , typically the environment from which <code>spLMstack</code> is called.
<code>family</code>	Specifies the distribution of the response as a member of the exponential family. Supported options are 'poisson', 'binomial' and 'binary'.
<code>coords</code>	an $n \times 2$ matrix of the observation coordinates in $\mathbb{R}^2$ (e.g., easting and northing).
<code>cor.fn</code>	a quoted keyword that specifies the correlation function used to model the spatial dependence structure among the observations. Supported covariance model keywords are: 'exponential' and 'matern'. See below for details.

priors	(optional) a list with each tag corresponding to a parameter name and containing prior details. Valid tags include V.beta, nu.beta, nu.z and sigmaSq.xi.
candidate.models	an object of class candidateModels containing a list of candidate models for stacking. See <code>candidateModels()</code> for details.
n.samples	number of posterior samples to be generated.
loopd.controls	a list with details on how leave-one-out predictive densities (LOO-PD) are to be calculated. Valid tags include method, CV.K, nMC and CV.update. The tag method can be either 'exact' or 'CV'. If sample size is more than 100, then the default is 'CV' with CV.K equal to its default value 10 (Gelman <i>et al.</i> 2024). The tag nMC decides how many Monte Carlo samples will be used to evaluate the leave-one-out predictive densities, which must be at least 500 (default). The tag CV.update is an advanced option, used only if method = 'CV', that decides how the pre-processing of the model fit on each fold is obtained, and should be changed with care as the faster choice depends on the BLAS library that R is linked with (see <code>sessionInfo()</code>). CV.update = 'update' obtains it by deletion updates of the full-data Cholesky factors, which is the faster choice with the reference BLAS that R ships with. CV.update = 'direct' recomputes it on each fold, which is faster only with an optimized BLAS such as OpenBLAS, Intel MKL or Apple Accelerate (vecLib), and is slower otherwise. The default CV.update = 'auto' uses 'direct' if such an optimized BLAS is detected from the library paths reported by R and 'update' otherwise; set it explicitly if the BLAS is not detected correctly (for example, an optimized BLAS installed in place of Rblas.dll on Windows). Both choices give the same results up to floating-point rounding, so only the run time is affected.
parallel	logical. If parallel=FALSE, the parallelization plan, if set up by the user, is ignored. If parallel=TRUE, the function inherits the parallelization plan that is set by the user via the function <code>future::plan()</code> only. Depending on the parallel backend available, users may choose their own plan. More details are available at https://cran.R-project.org/package=future .
solver	(optional) Specifies the name of the solver that will be used to obtain optimal stacking weights for each candidate model. Default order is c("CLARABEL", "ECOS", "SCS"). Users can use other solvers supported by the CVXR-package .
verbose	logical. If TRUE, prints model-specific optimal stacking weights.
...	currently no additional argument.

Details

Instead of assigning a prior on the process parameters ϕ and ν , the boundary adjustment parameter ϵ , we consider a set of candidate models based on some candidate values of these parameters supplied by the user. Suppose the set of candidate models is $\mathcal{M} = \{M_1, \dots, M_G\}$. Then for each $g = 1, \dots, G$, we sample from the posterior distribution $p(\sigma^2, \beta, z \mid y, M_g)$ under the model M_g and

find leave-one-out predictive densities $p(y_i | y_{-i}, M_g)$. Then we solve the optimization problem

$$\begin{aligned} \max_{w_1, \dots, w_G} \quad & \frac{1}{n} \sum_{i=1}^n \log \sum_{g=1}^G w_g p(y_i | y_{-i}, M_g) \\ \text{subject to} \quad & w_g \geq 0, \sum_{g=1}^G w_g = 1 \end{aligned}$$

to find the optimal stacking weights $\hat{w}_1, \dots, \hat{w}_G$.

Value

An object of class `spGLMstack`, which is a list including the following tags -

`family` the distribution of the responses as indicated in the function call

`samples` a list of length equal to total number of candidate models with each entry corresponding to a list of length 3, containing posterior samples of fixed effects (beta), spatial effects (z) and fine-scale variation term (xi) for that particular model.

`loopy` a list of length equal to total number of candidate models with each entry containing leave-one-out predictive densities under that particular model.

`loopy.method` a list containing details of the algorithm used for calculation of leave-one-out predictive densities. For K-fold cross-validation, its tag `cv.update` records the pre-processing method ('update' or 'direct') that was used.

`n.models` number of candidate models that are fit.

`model.params` a list with one element per candidate model, each a named list of its parameters: `phi`, `nu` (NA for the exponential correlation function) and `boundary` (boundary adjustment parameter).

`stacking.summary` a matrix with one row per candidate model, containing its parameters and its optimal stacking weight, for display.

`stacking.weights` a numeric vector of length equal to the number of candidate models storing the optimal stacking weights.

`run.time` a `proc_time` object with runtime details.

`diagnostics` a list of diagnostics. Element `numerical` is a data frame with one row per candidate model and columns `min.pivot` (the smallest relative Cholesky pivot of the $n \times n$ correlation matrix; values below $1e-8$ indicate a nearly singular correlation matrix), `min.cor` and `max.cor` (the correlations of the two farthest-apart and of the two closest locations; values of `min.cor` above 0.95 suggest an effective range far exceeding the extent of the data, values of `max.cor` below 0.05 nearly uncorrelated locations), obtained from quantities the fit computes anyway. Element `solver` describes the optimization for the stacking weights: the solver used (`used`) and its status (`status`), the installed and requested solvers, the search order, the attempts with their status, and whether the fallback `loo::stacking_weights()` was used. If `verbose = TRUE`, a "Diagnostics" section is printed if there is an issue: numerical flags of candidate models with stacking weight above 0.05 (extreme candidates with negligible weight are expected in a stacking grid and are only counted), and solver problems (a requested solver not installed, an inaccurate solution, or the fallback).

The return object might include additional data that is useful for subsequent prediction, model fit evaluation and other utilities.

Author(s)

Soumyakanti Pan span18@ucla.edu,
 Sudipto Banerjee sudipto@ucla.edu

References

Pan S, Zhang L, Bradley JR, Banerjee S (2025). "Bayesian Inference for Spatial-temporal Non-Gaussian Data Using Predictive Stacking." *Bayesian Analysis*, **In Press**. doi:[10.1214/25BA1582](https://doi.org/10.1214/25BA1582).
 Vehtari A, Simpson D, Gelman A, Yao Y, Gabry J (2024). "Pareto Smoothed Importance Sampling." *Journal of Machine Learning Research*, **25**(72), 1-58. URL <https://jmlr.org/papers/v25/19-556.html>.

See Also

[spGLMexact\(\)](#), [spLMstack\(\)](#)

Examples

```
set.seed(1234)
data(simSpatial)
dat <- simSpatial[1:150, ]
cand.mod <- candidateModels(list(phi = c(2, 4, 8),
                                nu = c(1.5, 1.75),
                                boundary = c(0.75)), "cartesian")

mod1 <- spGLMstack(y_pois ~ x1 + x2, data = dat, family = "poisson",
                  coords = as.matrix(dat[, c("s1", "s2")]), cor.fn = "matern",
                  candidate.models = cand.mod,
                  n.samples = 500,
                  loopd.controls = list(method = "CV", CV.K = 10, nMC = 500),
                  parallel = FALSE, verbose = TRUE)

post_samps <- stackedSampler(mod1)
post_beta <- post_samps$beta
print(t(apply(post_beta, 1, function(x) quantile(x, c(0.025, 0.5, 0.975))))))

# compare the posterior medians of the spatial effects with the truth
cor(apply(post_samps$z, 1, median), dat$z_true)
```

 spLMexact

Univariate Bayesian spatial linear model

Description

Fits a Bayesian spatial linear model with spatial process parameters and the noise-to-spatial variance ratio fixed to a value supplied by the user. The output contains posterior samples of the fixed effects, variance parameter, spatial random effects and, if required, leave-one-out predictive densities.

Usage

```

spLMexact(
  formula,
  data = parent.frame(),
  coords,
  cor.fn,
  priors = "flat",
  spParams,
  noise_sp_ratio,
  n.samples,
  loopd = FALSE,
  loopd.method = "exact",
  verbose = TRUE,
  ...
)

```

Arguments

formula	a symbolic description of the regression model to be fit. See example below.
data	an optional data frame containing the variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> , typically the environment from which <code>spLMexact</code> is called.
coords	an $n \times 2$ matrix of the observation coordinates in $\mathbb{R}^2$ (e.g., easting and northing).
cor.fn	a quoted keyword that specifies the correlation function used to model the spatial dependence structure among the observations. Supported covariance model keywords are: 'exponential' and 'matern'. See below for details.
priors	either "flat" (default), which assigns the prior $p(\beta, \sigma^2) \propto 1/\sigma^2$, or a list with tags <code>beta.norm</code> (a list containing μ_β and V_β) and/or <code>sigma.sq.ig</code> (a vector containing a_σ and b_σ). A component not supplied in the list receives its flat prior, $p(\beta) \propto 1$ or $p(\sigma^2) \propto 1/\sigma^2$.
spParams	fixed value of spatial process parameters.
noise_sp_ratio	noise-to-spatial variance ratio.
n.samples	number of posterior samples to be generated.
loopd	logical. If <code>loopd=TRUE</code> , returns leave-one-out predictive densities, using method as given by <code>loopd.method</code> . Default is <code>FALSE</code> .
loopd.method	character. Ignored if <code>loopd=FALSE</code> . If <code>loopd=TRUE</code> , valid inputs are 'exact' and 'PSIS'. The option 'exact' corresponds to exact leave-one-out predictive densities which requires computation almost equivalent to fitting the model n times. The option 'PSIS' is faster and finds approximate leave-one-out predictive densities using Pareto-smoothed importance sampling (Gelman <i>et al.</i> 2024).
verbose	logical. If <code>verbose = TRUE</code> , prints model description.
...	currently no additional argument.

Details

Suppose $\chi = (s_1, \dots, s_n)$ denotes the n spatial locations the response y is observed. With this function, we fit a conjugate Bayesian hierarchical spatial model

$$\begin{aligned} y \mid z, \beta, \sigma^2 &\sim N(X\beta + z, \sigma^2 I_n), \quad z \mid \sigma_z^2 \sim N(0, \sigma_z^2 R(\chi; \phi, \nu)), \\ \beta \mid \sigma^2 &\sim N(\mu_\beta, \sigma^2 V_\beta), \quad \sigma^2 \sim \text{IG}(a_\sigma, b_\sigma) \end{aligned}$$

where we fix the noise-to-spatial variance ratio $\delta^2 = \sigma^2 / \sigma_z^2$, the spatial process parameters ϕ and ν , and the hyperparameters μ_β , V_β , a_σ and b_σ . If `priors = "flat"`, we instead assign the prior $p(\beta, \sigma^2) \propto 1/\sigma^2$. We utilize a composition sampling strategy to sample the model parameters from their joint posterior distribution which can be written as

$$p(\sigma^2, \beta, z \mid y) = p(\sigma^2 \mid y) \times p(\beta \mid \sigma^2, y) \times p(z \mid \beta, \sigma^2, y).$$

We proceed by first sampling σ^2 from its marginal posterior, then given the samples of σ^2 , we sample β and subsequently, we sample z conditioned on the posterior samples of β and σ^2 (Banerjee 2020). Posterior samples of the spatial variance are obtained as $\sigma_z^2 = \sigma^2 / \delta^2$.

Value

An object of class `spLMexact`, which is a list with the following tags -

samples a list of length 4, containing posterior samples of fixed effects (`beta`), measurement error variance (`sigmaSq`), spatial variance (`sigmaSq.z`), and spatial effects (`z`).

loopd If `loopd=TRUE`, contains leave-one-out predictive densities.

model.params Values of the fixed parameters that includes `phi` (spatial decay), `nu` (spatial smoothness; NA for the exponential correlation function) and `noise_sp_ratio` (noise-to-spatial variance ratio).

diagnostics a list of fit diagnostics, obtained from quantities the fit computes anyway. Element `numerical` is a data frame with one row and columns `min.pivot` (the smallest relative Cholesky pivot of the $n \times n$ factorizations; values below $1e-8$ indicate a nearly singular covariance matrix), `min.cor` and `max.cor` (the correlations of the two farthest-apart and of the two closest locations; values of `min.cor` above 0.95 suggest an effective range far exceeding the extent of the data, values of `max.cor` below 0.05 nearly uncorrelated locations). If `loopd.method = 'PSIS'`, element `pareto` is a list with the Pareto k diagnostic values of the leave-one-out predictive densities (k), the threshold above which they are unreliable (`threshold`, Vehtari *et al.* 2024) and the number of values above it (`n.high`). If `verbose = TRUE`, a "Diagnostics" section is printed when any threshold is crossed.

The return object might include additional data used for subsequent prediction and/or model fit evaluation.

Author(s)

Soumyakanti Pan span18@ucla.edu,
Sudipto Banerjee sudipto@ucla.edu

References

- Banerjee S (2020). "Modeling massive spatial datasets using a conjugate Bayesian linear modeling framework." *Spatial Statistics*, **37**, 100417. ISSN 2211-6753. doi:10.1016/j.spasta.2020.100417.
- Vehtari A, Simpson D, Gelman A, Yao Y, Gabry J (2024). "Pareto Smoothed Importance Sampling." *Journal of Machine Learning Research*, **25**(72), 1-58. URL <https://jmlr.org/papers/v25/19-556.html>.

See Also

[spLMstack\(\)](#)

Examples

```
data(simSpatial)
dat <- simSpatial[1:100, ]

# setup prior list
muBeta <- c(0, 0, 0)
VBeta <- diag(100, 3)
sigmaSqIGa <- 2
sigmaSqIGb <- 0.1
prior_list <- list(beta.norm = list(muBeta, VBeta),
                   sigma.sq.ig = c(sigmaSqIGa, sigmaSqIGb))

mod1 <- spLMexact(y_gauss ~ x1 + x2, data = dat,
                  coords = as.matrix(dat[, c("s1", "s2")]),
                  cor.fn = "matern",
                  priors = prior_list,
                  spParams = list(phi = 4, nu = 1.75),
                  noise_sp_ratio = 0.1,
                  n.samples = 100,
                  loopd = TRUE, loopd.method = "exact")

post_beta <- mod1$samples$beta
print(t(apply(post_beta, 1, function(x) quantile(x, c(0.025, 0.5, 0.975))))))

# compare the posterior medians of the spatial effects with the truth
cor(apply(mod1$samples$z, 1, median), dat$z_true)
```

Description

Fits Bayesian spatial linear model on a collection of candidate models constructed based on some candidate values of some model parameters specified by the user and subsequently combines inference by stacking predictive densities. See Zhang, Tang and Banerjee (2025) for more details.

Usage

```

spLMstack(
  formula,
  data = parent.frame(),
  coords,
  cor.fn,
  priors = "flat",
  candidate.models,
  n.samples,
  loopd.method,
  parallel = FALSE,
  solver = NULL,
  verbose = TRUE,
  ...
)

```

Arguments

formula	a symbolic description of the regression model to be fit. See example below.
data	an optional data frame containing the variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> , typically the environment from which <code>spLMstack</code> is called.
coords	an $n \times 2$ matrix of the observation coordinates in $\mathbb{R}^2$ (e.g., easting and northing).
cor.fn	a quoted keyword that specifies the correlation function used to model the spatial dependence structure among the observations. Supported covariance model key words are: 'exponential' and 'matern'. See below for details.
priors	either "flat" (default), which assigns the prior $p(\beta, \sigma^2) \propto 1/\sigma^2$, or a list with tags <code>beta.norm</code> (a list containing μ_β and V_β) and/or <code>sigma.sq.ig</code> (a vector containing a_σ and b_σ). A component not supplied in the list receives its flat prior, $p(\beta) \propto 1$ or $p(\sigma^2) \propto 1/\sigma^2$.
candidate.models	an object of class <code>candidateModels</code> containing a list of candidate models for stacking. See <code>candidateModels()</code> for details.
n.samples	number of posterior samples to be generated.
loopd.method	character. Valid inputs are 'exact' and 'PSIS'. The option 'exact' corresponds to exact leave-one-out predictive densities. The option 'PSIS' is faster, as it finds approximate leave-one-out predictive densities using Pareto-smoothed importance sampling (Gelman <i>et al.</i> 2024).
parallel	logical. If <code>parallel=FALSE</code> , the parallelization plan, if set up by the user, is ignored. If <code>parallel=TRUE</code> , the function inherits the parallelization plan that is set by the user via the function <code>future::plan()</code> only. Depending on the parallel backend available, users may choose their own plan. More details are available at https://cran.R-project.org/package=future .
solver	(optional) Specifies the name of the solver that will be used to obtain optimal stacking weights for each candidate model. Default order is <code>c("CLARABEL",</code>

"ECOS", "SCS"). Users can use other solvers supported by the [CVXR-package](#) package.

verbose logical. If TRUE, prints model-specific optimal stacking weights.

... currently no additional argument.

Details

Instead of assigning a prior on the process parameters ϕ and ν , noise-to-spatial variance ratio δ^2 , we consider a set of candidate models based on some candidate values of these parameters supplied by the user. Suppose the set of candidate models is $\mathcal{M} = \{M_1, \dots, M_G\}$. Then for each $g = 1, \dots, G$, we sample from the posterior distribution $p(\sigma^2, \beta, z \mid y, M_g)$ under the model M_g and find leave-one-out predictive densities $p(y_i \mid y_{-i}, M_g)$. Then we solve the optimization problem

$$\begin{aligned} \max_{w_1, \dots, w_G} \quad & \frac{1}{n} \sum_{i=1}^n \log \sum_{g=1}^G w_g p(y_i \mid y_{-i}, M_g) \\ \text{subject to} \quad & w_g \geq 0, \sum_{g=1}^G w_g = 1 \end{aligned}$$

to find the optimal stacking weights $\hat{w}_1, \dots, \hat{w}_G$.

Value

An object of class `spLMstack`, which is a list including the following tags -

`samples` a list of length equal to total number of candidate models with each entry corresponding to a list of length 4, containing posterior samples of fixed effects (beta), measurement error variance (sigmaSq), spatial variance (sigmaSq.z), and spatial effects (z) for that model.

`loopd` a list of length equal to total number of candidate models with each entry containing leave-one-out predictive densities under that particular model.

`n.models` number of candidate models that are fit.

`model.params` a list with one element per candidate model, each a named list of its parameters: `phi`, `nu` (NA for the exponential correlation function) and `noise_sp_ratio` (noise-to-spatial variance ratio).

`stacking.summary` a matrix with one row per candidate model, containing its parameters and its optimal stacking weight, for display.

`stacking.weights` a numeric vector of length equal to the number of candidate models storing the optimal stacking weights.

`run.time` a `proc_time` object with runtime details.

`diagnostics` a list of diagnostics. Element `numerical` is a data frame with one row per candidate model and columns `min.pivot` (the smallest relative Cholesky pivot of the $n \times n$ factorizations; values below $1e-8$ indicate a nearly singular covariance matrix), `min.cor` and `max.cor` (the correlations of the two farthest-apart and of the two closest locations; values of `min.cor` above 0.95 suggest an effective range far exceeding the extent of the data, values of `max.cor` below 0.05 nearly uncorrelated locations), obtained from quantities the fits compute anyway. If `loopd.method = 'PSIS'`, element `pareto` is a list with the Pareto k diagnostic values of

each candidate model (k), the threshold above which they are unreliable (`threshold`) and the number of values above it for each model (`n.high`). `Element solver` describes the optimization for the stacking weights: the solver used (`used`) and its status (`status`), the installed and requested solvers, the search order, the attempts with their status, and whether the fallback `loo::stacking_weights()` was used. If `verbose = TRUE`, a "Diagnostics" section is printed if there is an issue: numerical flags of candidate models with stacking weight above 0.05 (extreme candidates with negligible weight are expected in a stacking grid and are only counted), Pareto k values above the threshold for any model, and solver problems (a requested solver not installed, an inaccurate solution, or the fallback).

The return object might include additional data that is useful for subsequent prediction, model fit evaluation and other utilities.

Author(s)

Soumyakanti Pan span18@ucla.edu,
Sudipto Banerjee sudipto@ucla.edu

References

Vehtari A, Simpson D, Gelman A, Yao Y, Gabry J (2024). "Pareto Smoothed Importance Sampling." *Journal of Machine Learning Research*, **25**(72), 1-58. URL <https://jmlr.org/papers/v25/19-556.html>.

Zhang L, Tang W, Banerjee S (2025). "Bayesian Geostatistics Using Predictive Stacking." *Journal of the American Statistical Association*, **In press**. doi:10.1080/01621459.2025.2566449.

See Also

[spLMexact\(\)](#), [spGLMstack\(\)](#)

Examples

```
set.seed(1234)
data(simSpatial)
dat <- simSpatial[1:100, ]

cand.mod <- candidateModels(list(phi = c(2, 4, 6),
                                   nu = c(1, 1.5, 1.75),
                                   noise_sp_ratio = c(0.1, 0.25)),
                             "cartesian")

mod1 <- spLMstack(y_gauss ~ x1 + x2, data = dat,
                  coords = as.matrix(dat[, c("s1", "s2")]),
                  cor.fn = "matern",
                  candidate.models = cand.mod,
                  n.samples = 1000, loopd.method = "exact",
                  parallel = FALSE, verbose = TRUE)

post_samps <- stackedSampler(mod1)
post_beta <- post_samps$beta
print(t(apply(post_beta, 1, function(x) quantile(x, c(0.025, 0.5, 0.975))))))
```

```
# compare the posterior medians of the spatial effects with the truth
cor(apply(post_samps$z, 1, median), dat$z_true)
```

stackedSampler	<i>Sample from the stacked posterior distribution</i>
----------------	---

Description

A helper function to sample from the stacked posterior distribution to obtain final posterior samples that can be used for subsequent analysis. This function applies on outputs of functions [sPLMstack\(\)](#), [spGLMstack\(\)](#), [svcLMstack\(\)](#), [svcGLMstack\(\)](#), [stvcGLMstack\(\)](#) and [stvcLMstack\(\)](#).

Usage

```
stackedSampler(mod_out, n.samples)
```

Arguments

mod_out	an object that is an output of a model fit or a prediction task, i.e., the class should be either <code>sPLMstack</code> , <code>pp.sPLMstack</code> , <code>spGLMstack</code> , <code>pp.spGLMstack</code> , <code>stvcGLMstack</code> , <code>pp.stvcGLMstack</code> , <code>stvcLMstack</code> , <code>pp.stvcLMstack</code> , <code>svcLMstack</code> , <code>pp.svcLMstack</code> , <code>svcGLMstack</code> or <code>pp.svcGLMstack</code> .
n.samples	(optional) If missing, inherits the number of posterior samples from the original output. Otherwise, it specifies number of posterior samples to draw from the stacked posterior. If it exceeds the number of posterior draws used in the original function, then a message is thrown and the samples are obtained by resampling. We recommend running the original model fit/prediction with enough samples.

Details

After obtaining the optimal stacking weights $\hat{w}_1, \dots, \hat{w}_G$, posterior inference of quantities of interest subsequently proceed from the *stacked* posterior,

$$\tilde{p}(\cdot | y) = \sum_{g=1}^G \hat{w}_g p(\cdot | y, M_g),$$

where $\mathcal{M} = \{M_1, \dots, M_g\}$ is the collection of candidate models.

Value

An object of class `stacked_posterior`, which is a list that includes the following tags -

beta samples of the fixed effect from the stacked joint posterior.

z samples of the spatial random effects from the stacked joint posterior.

The list may also include other scale parameters corresponding to the model.

Author(s)

Soumyakanti Pan span18@ucla.edu,
 Sudipto Banerjee sudipto@ucla.edu

See Also

[spLMstack\(\)](#), [spGLMstack\(\)](#), [svclMstack\(\)](#), [svcGLMstack\(\)](#), [stvcGLMstack\(\)](#), [stvcLMstack\(\)](#)

Examples

```
set.seed(1234)
data(simSpatial)
dat <- simSpatial[1:100, ]
cand.mod <- candidateModels(list(phi = c(2, 4, 6),
                                   nu = c(1, 1.5, 1.75),
                                   noise_sp_ratio = c(0.1)), "cartesian")

mod1 <- spLMstack(y_gauss ~ x1 + x2, data = dat,
                  coords = as.matrix(dat[, c("s1", "s2")]),
                  cor.fn = "matern",
                  candidate.models = cand.mod,
                  n.samples = 1000, loopd.method = "exact",
                  parallel = FALSE, verbose = TRUE)
print(mod1$diagnostics$solver$used)
print(mod1$diagnostics$solver$status)
print(mod1$run.time)

post_samps <- stackedSampler(mod1)
post_beta <- post_samps$beta
print(t(apply(post_beta, 1, function(x) quantile(x, c(0.025, 0.5, 0.975))))))
```

stvcGLMexact

Bayesian spatially-temporally varying generalized linear model

Description

Fits a Bayesian generalized linear model with spatially-temporally varying coefficients under fixed values of spatial-temporal process parameters and some auxiliary model parameters. The output contains posterior samples of the fixed effects, spatial-temporal random effects and, if required, finds leave-one-out predictive densities.

Usage

```
stvcGLMexact(
  formula,
  data = parent.frame(),
  family,
  sp_coords,
```

```

    time_coords,
    cor.fn,
    process.type,
    sptParams,
    priors,
    boundary = 0.5,
    n.samples,
    loopd = FALSE,
    loopd.method = "exact",
    CV.K = 10,
    loopd.nMC = 500,
    verbose = TRUE,
    ...
)

```

Arguments

<code>formula</code>	a symbolic description of the regression model to be fit. Variables in parenthesis are assigned spatially-temporally varying coefficients. See examples.
<code>data</code>	an optional data frame containing the variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> , typically the environment from which <code>stvcGLMexact</code> is called.
<code>family</code>	Specifies the distribution of the response as a member of the exponential family. Supported options are 'poisson', 'binomial' and 'binary'.
<code>sp_coords</code>	an $n \times 2$ matrix of the observation spatial coordinates in $\mathbb{R}^2$ (e.g., easting and northing).
<code>time_coords</code>	an $n \times 1$ matrix of the observation temporal coordinates in $\mathcal{T} \subseteq [0, \infty)$.
<code>cor.fn</code>	a quoted keyword that specifies the correlation function used to model the spatial-temporal dependence structure among the observations. Supported covariance model key words are: 'gneiting-decay' (Gneiting and Guttorp 2010). See below for details.
<code>process.type</code>	a quoted keyword specifying the model for the spatial-temporal process. Supported keywords are 'independent' which indicates independent processes for each varying coefficients characterized by different process parameters, <code>independent.shared</code> implies independent processes for the varying coefficients that shares common process parameters, and <code>multivariate</code> implies correlated processes for the varying coefficients modeled by a multivariate Gaussian process with an inverse-Wishart prior on the correlation matrix. The input for <code>sptParams</code> and <code>priors</code> must be given accordingly.
<code>sptParams</code>	fixed values of spatial-temporal process parameters in usually a list of length 2. If <code>cor.fn='gneiting-decay'</code> , then it is a list of length 2 with tags <code>phi_s</code> and <code>phi_t</code> . If <code>process.type='independent'</code> , then <code>phi_s</code> and <code>phi_t</code> contain fixed values of the r spatial-temporal processes, otherwise they will contain scalars. See examples below.
<code>priors</code>	(optional) a list with each tag corresponding to a hyperparameter name and containing hyperprior details. Valid tags include <code>V.beta</code> , <code>nu.beta</code> , <code>nu.z</code> , <code>sigmaSq.xi</code>

	and IW.scale. Values of nu.beta and nu.z must be at least 2.1. If not supplied, uses defaults.
boundary	Specifies the boundary adjustment parameter. Must be a real number between 0 and 1. Default is 0.5.
n.samples	number of posterior samples to be generated.
loopd	logical. If loopd=TRUE, returns leave-one-out predictive densities, using method as given by loopd.method. Default is FALSE.
loopd.method	character. Ignored if loopd=FALSE. If loopd=TRUE, valid inputs are 'exact', 'CV'. The option 'exact' corresponds to exact leave-one-out predictive densities which requires computation almost equivalent to fitting the model n times. The options 'CV' is faster as it implements K -fold cross validation to find approximate leave-one-out predictive densities (Vehtari <i>et al.</i> 2017).
CV.K	An integer between 10 and 20. Considered only if loopd.method='CV'. Default is 10 (as recommended in Vehtari <i>et al.</i> 2017).
loopd.nMC	Number of Monte Carlo samples to be used to evaluate leave-one-out predictive densities when loopd.method is set to either 'exact' or 'CV'.
verbose	logical. If verbose = TRUE, prints model description.
...	currently no additional argument.

Details

With this function, we fit a Bayesian hierarchical spatially-temporally varying generalized linear model by sampling exactly from the joint posterior distribution utilizing the generalized conjugate multivariate distribution theory (Bradley and Clinch 2024). Suppose $\chi = (\ell_1, \dots, \ell_n)$ denotes the n spatial-temporal co-ordinates in $\mathcal{L} = \mathcal{S} \times \mathcal{T}$, the response y is observed. Let $y(\ell)$ be the outcome at the co-ordinate ℓ endowed with a probability law from the natural exponential family, which we denote by

$$y(\ell) \sim \text{EF}(x(\ell)^\top \beta + \tilde{x}(\ell)^\top z(\ell); b(\ell), \psi)$$

for some positive parameter $b(\ell) > 0$ and unit log partition function ψ . Here, $\tilde{x}(\ell)$ denotes co-variates with spatially-temporally varying coefficients. We consider the following response models based on the input supplied to the argument family.

'poisson' It considers point-referenced Poisson responses $y(\ell) \sim \text{Poisson}(e^{x(\ell)^\top \beta + \tilde{x}(\ell)^\top z(\ell)})$. Here, $b(\ell) = 1$ and $\psi(t) = e^t$.

'binomial' It considers point-referenced binomial counts $y(\ell) \sim \text{Binomial}(m(\ell), \pi(\ell))$ where, $m(\ell)$ denotes the total number of trials and probability of success $\pi(\ell) = \text{ilogit}(x(\ell)^\top \beta + \tilde{x}(\ell)^\top z(\ell))$ at spatial-temporal co-ordinate ℓ . Here, $b = m(\ell)$ and $\psi(t) = \log(1 + e^t)$.

'binary' It considers point-referenced binary data (0 or, 1) i.e., $y(\ell) \sim \text{Bernoulli}(\pi(\ell))$, where probability of success $\pi(\ell) = \text{ilogit}(x(\ell)^\top \beta + \tilde{x}(\ell)^\top z(\ell))$ at spatial-temporal co-ordinate ℓ . Here, $b(\ell) = 1$ and $\psi(t) = \log(1 + e^t)$.

The hierarchical model is given as

$$\begin{aligned}
y(\ell_i) \mid \beta, z, \xi &\sim \text{EF}(x(\ell_i)^\top \beta + \tilde{x}(\ell_i)^\top z(s_i) + \xi_i - \mu_i; b_i, \psi_y), i = 1, \dots, n \\
\xi \mid \beta, z, \sigma_\xi^2, \alpha_\epsilon &\sim \text{GCM}_c(\dots), \\
\beta \mid \sigma_\beta^2 &\sim N(0, \sigma_\beta^2 V_\beta), \quad \sigma_\beta^2 \sim \text{IG}(\nu_\beta/2, \nu_\beta/2) \\
z_j \mid \sigma_{z_j}^2 &\sim N(0, \sigma_{z_j}^2 R(\chi; \phi_s, \phi_t)), \quad \sigma_{z_j}^2 \sim \text{IG}(\nu_z/2, \nu_z/2), j = 1, \dots, r
\end{aligned}$$

where $\mu = (\mu_1, \dots, \mu_n)^\top$ denotes the discrepancy parameter. We fix the spatial-temporal process parameters ϕ_s and ϕ_t and the hyperparameters $V_\beta, \nu_\beta, \nu_z$ and σ_ξ^2 . The term ξ is known as the fine-scale variation term which is given a conditional generalized conjugate multivariate distribution as prior. For more details, see Pan *et al.* 2025. Default values for $V_\beta, \nu_\beta, \nu_z, \sigma_\xi^2$ are diagonal with each diagonal element 100, 2.1, 2.1 and 0.1 respectively. If `process.type = 'multivariate'`, the $r \times r$ cross-covariance matrix Σ of the processes has the prior $\Sigma \sim \text{IW}(\nu_z + 2r, \Psi)$, with Ψ given by `IW.scale`; its default $\Psi = (\nu_z + r - 1)I_r$ gives $E[\Sigma] = I_r$. The widths of the posterior credible intervals of the processes grow with the prior scale.

Value

An object of class `stvcGLMexact`, which is a list with the following tags -

priors details of the priors used, containing the values of the boundary adjustment parameter (boundary), the variance parameter of the fine-scale variation term (`simasq.xi`) and others.

samples a list of length 3, containing posterior samples of fixed effects (beta), spatial-temporal effects (z) and the fine-scale variation term (xi). The element with tag z will again be a list of length r, each containing posterior samples of the spatial-temporal random effects corresponding to each varying coefficient.

loopd If `loopd=TRUE`, contains leave-one-out predictive densities.

model.params Values of the fixed parameters that includes `phi_s` (spatial decay), `phi_t` (temporal smoothness).

diagnostics a list of diagnostics. Element `numerical` is a data frame with one row (one row per process, with columns `model` and `process`, if `process.type = 'independent'`) and columns `min.pivot` (the smallest relative Cholesky pivot of the $n \times n$ spatial-temporal correlation matrix; values below 1e-8 indicate a nearly singular correlation matrix), `min.cor` and `max.cor` (the correlations of the two farthest-apart and of the two closest space-time locations; values of `min.cor` above 0.95 suggest an effective range far exceeding the extent of the data, values of `max.cor` below 0.05 nearly uncorrelated space-time locations), obtained from quantities the fit computes anyway. If `verbose = TRUE`, a "Diagnostics" section is printed when any threshold is crossed.

The return object might include additional data that can be used for subsequent prediction and/or model fit evaluation.

Author(s)

Soumyakanti Pan span18@ucla.edu

References

- Bradley JR, Clinch M (2024). "Generating Independent Replicates Directly from the Posterior Distribution for a Class of Spatial Hierarchical Models." *Journal of Computational and Graphical Statistics*, 0(0), 1-17. doi:[10.1080/10618600.2024.2365728](https://doi.org/10.1080/10618600.2024.2365728).
- T. Gneiting and P. Guttorm (2010). "Continuous-parameter spatio-temporal processes." In A.E. Gelfand, P.J. Diggle, M. Fuentes, and P. Guttorm, editors, *Handbook of Spatial Statistics*, Chapman & Hall CRC Handbooks of Modern Statistical Methods, p 427–436. Taylor and Francis.

Pan S, Zhang L, Bradley JR, Banerjee S (2025). "Bayesian Inference for Spatial-temporal Non-Gaussian Data Using Predictive Stacking." *Bayesian Analysis*, **In Press**. doi:10.1214/25BA1582.

Vehtari A, Gelman A, Gabry J (2017). "Practical Bayesian Model Evaluation Using Leave-One-out Cross-Validation and WAIC." *Statistics and Computing*, **27**(5), 1413-1432. ISSN 0960-3174. doi:10.1007/s1122201696964.

See Also

[spGLMexact\(\)](#)

Examples

```
data(simSpaceTime)
dat <- simSpaceTime[1:100, ]

# Fit a spatial-temporal varying coefficient Poisson GLM
mod1 <- stvcGLMexact(y_pois ~ x1 + x2 + (x1), data = dat, family = "poisson",
                    sp_coords = as.matrix(dat[, c("s1", "s2")]),
                    time_coords = as.matrix(dat[, "t_coords"]),
                    cor.fn = "gneiting-decay",
                    process.type = "independent",
                    sptParams = list(phi_s = c(3, 3), phi_t = c(4, 4)),
                    verbose = FALSE, n.samples = 100)

# rows 1, ..., n of z hold the varying intercept, rows n+1, ..., 2n the
# varying slope of x1
n <- nrow(dat)
z_hat <- apply(mod1$samples$z, 1, median)
cor(z_hat[1:n], dat$z1_true)
```

stvcGLMstack

Bayesian spatially-temporally varying coefficients generalized linear model using predictive stacking

Description

Fits Bayesian spatial-temporal generalized linear model with spatially-temporally varying coefficients on a collection of candidate models constructed based on some candidate values of some model parameters specified by the user and subsequently combines inference by stacking predictive densities. See Pan, Zhang, Bradley, and Banerjee (2025) for more details.

Usage

```
stvcGLMstack(
  formula,
  data = parent.frame(),
  family,
  sp_coords,
```

```

    time_coords,
    cor.fn,
    process.type,
    priors,
    candidate.models,
    n.samples,
    loopd.controls,
    parallel = FALSE,
    solver = NULL,
    verbose = TRUE,
    ...
)

```

Arguments

<code>formula</code>	a symbolic description of the regression model to be fit. Variables in parenthesis are assigned spatially-temporally varying coefficients. See examples.
<code>data</code>	an optional data frame containing the variables in the model. If not found in <code>data</code> , the variables are taken from <code>environment(formula)</code> , typically the environment from which <code>stvcGLMstack</code> is called.
<code>family</code>	Specifies the distribution of the response as a member of the exponential family. Supported options are 'poisson', 'binomial' and 'binary'.
<code>sp_coords</code>	an $n \times 2$ matrix of the observation spatial coordinates in $\mathbb{R}^2$ (e.g., easting and northing).
<code>time_coords</code>	an $n \times 1$ matrix of the observation temporal coordinates in $\mathcal{T} \subseteq [0, \infty)$.
<code>cor.fn</code>	a quoted keyword that specifies the correlation function used to model the spatial-temporal dependence structure among the observations. Supported covariance model key words are: 'gneiting-decay' (Gneiting and Guttorp 2010). See below for details.
<code>process.type</code>	a quoted keyword specifying the model for the spatial-temporal process. Supported keywords are 'independent' which indicates independent processes for each varying coefficients characterized by different process parameters, <code>independent.shared</code> implies independent processes for the varying coefficients that shares common process parameters, and <code>multivariate</code> implies correlated processes for the varying coefficients modeled by a multivariate Gaussian process with an inverse-Wishart prior on the correlation matrix. The input for <code>sptParams</code> and <code>priors</code> must be given accordingly.
<code>priors</code>	(optional) a list with each tag corresponding to a hyperparameter name and containing hyperprior details. Valid tags include <code>V.beta</code> , <code>nu.beta</code> , <code>nu.z</code> , <code>sigmaSq.xi</code> and <code>IW.scale</code> . Values of <code>nu.beta</code> and <code>nu.z</code> must be at least 2.1. If not supplied, uses defaults.
<code>candidate.models</code>	an object of class <code>candidateModels</code> containing a list of candidate models for stacking. See candidateModels() for details.
<code>n.samples</code>	number of samples to be drawn from the posterior distribution.

<code>loopd.controls</code>	a list with details on how leave-one-out predictive densities (LOO-PD) are to be calculated. Valid tags include <code>method</code> , <code>CV.K</code> , <code>nMC</code> and <code>CV.update</code> . The tag <code>method</code> can be either <code>'exact'</code> or <code>'CV'</code> . If sample size is more than 100, then the default is <code>'CV'</code> with <code>CV.K</code> equal to its default value 10 (Gelman <i>et al.</i> 2024). The tag <code>nMC</code> decides how many Monte Carlo samples will be used to evaluate the leave-one-out predictive densities, which must be at least 500 (default). The tag <code>CV.update</code> is an advanced option, used only if <code>method = 'CV'</code> , that decides how the pre-processing of the model fit on each fold is obtained, and should be changed with care as the faster choice depends on the BLAS library that R is linked with (see <code>sessionInfo()</code>). <code>CV.update = 'update'</code> obtains it by deletion updates of the full-data Cholesky factors, which is the faster choice with the reference BLAS that R ships with. <code>CV.update = 'direct'</code> recomputes it on each fold, which is faster only with an optimized BLAS such as OpenBLAS, Intel MKL or Apple Accelerate (vecLib), and is slower otherwise. The default <code>CV.update = 'auto'</code> uses <code>'direct'</code> if such an optimized BLAS is detected from the library paths reported by R and <code>'update'</code> otherwise; set it explicitly if the BLAS is not detected correctly (for example, an optimized BLAS installed in place of <code>Rblas.dll</code> on Windows). Both choices give the same results up to floating-point rounding, so only the run time is affected.
<code>parallel</code>	logical. If <code>parallel=FALSE</code> , the parallelization plan, if set up by the user, is ignored. If <code>parallel=TRUE</code> , the function inherits the parallelization plan that is set by the user via the function <code>future::plan()</code> only. Depending on the parallel backend available, users may choose their own plan. More details are available at https://cran.R-project.org/package=future .
<code>solver</code>	(optional) Specifies the name of the solver that will be used to obtain optimal stacking weights for each candidate model. Default order is <code>c("CLARABEL", "ECOS", "SCS")</code> . Users can use other solvers supported by the CVXR-package package.
<code>verbose</code>	logical. If <code>TRUE</code> , prints model-specific optimal stacking weights.
<code>...</code>	currently no additional argument.

Value

An object of class `stvcGLMstack`, which is a list including the following tags -

- `samples` a list of length equal to total number of candidate models with each entry corresponding to a list of length 3, containing posterior samples of fixed effects (beta), spatial effects (z), and fine scale variation xi for that model.
- `loopd` a list of length equal to total number of candidate models with each entry containing leave-one-out predictive densities under that particular model.
- `n.models` number of candidate models that are fit.
- `model.params` a list with one element per candidate model, each a named list of its parameters: `phi_s`, `phi_t` (vectors of length *r* if `process.type = 'independent'`) and boundary.
- `stacking.summary` a matrix with one row per candidate model, containing its parameters and its optimal stacking weight, for display.
- `stacking.weights` a numeric vector of length equal to the number of candidate models storing the optimal stacking weights.

`run.time` a `proc_time` object with runtime details.

`diagnostics` a list of diagnostics. Element `numerical` is a data frame with one row per candidate model (per candidate model and process, with columns `model` and `process`, if `process.type = 'independent'`) and columns `min.pivot` (the smallest relative Cholesky pivot of the $n \times n$ spatial-temporal correlation matrix; values below $1e-8$ indicate a nearly singular correlation matrix), `min.cor` and `max.cor` (the correlations of the two farthest-apart and of the two closest space-time locations; values of `min.cor` above 0.95 suggest an effective range far exceeding the extent of the data, values of `max.cor` below 0.05 nearly uncorrelated space-time locations), obtained from quantities the fit computes anyway. Element `solver` describes the optimization for the stacking weights: the solver used (`used`) and its status (`status`), the installed and requested solvers, the search order, the attempts with their status, and whether the fallback `loo::stacking_weights()` was used. If `verbose = TRUE`, a "Diagnostics" section is printed if there is an issue: numerical flags of candidate models with stacking weight above 0.05 (extreme candidates with negligible weight are expected in a stacking grid and are only counted), and solver problems (a requested solver not installed, an inaccurate solution, or the fallback).

This object can be further used to recover posterior samples of the scale parameters in the model, and subsequently, to make predictions at new locations or times using the function `posteriorPredict()`.

Examples

```
set.seed(1234)
data(simSpaceTime)
dat <- simSpaceTime[1:100, ]

# create list of candidate models (multivariate)
mod.list2 <- candidateModels(list(phi_s = list(2, 3, 4),
                                     phi_t = list(4, 8),
                                     boundary = c(0.75)), "cartesian")

# fit a spatial-temporal varying coefficient model using predictive stacking
mod1 <- stvcGLMstack(y_pois ~ x1 + x2 + (x1), data = dat, family = "poisson",
                    sp_coords = as.matrix(dat[, c("s1", "s2")]),
                    time_coords = as.matrix(dat[, "t_coords"]),
                    cor.fn = "gneiting-decay",
                    process.type = "multivariate",
                    candidate.models = mod.list2,
                    loopd.controls = list(method = "CV", CV.K = 10, nMC = 500),
                    n.samples = 500)
post_samps <- stackedSampler(mod1)
```

stvcLMexact

Bayesian spatially-temporally varying coefficients linear model

Description

Fits a Bayesian linear model with spatially-temporally varying coefficients for a Gaussian response, with the spatial-temporal process parameters and the noise-to-spatial variance ratios fixed to values supplied by the user. The output contains exact posterior samples of the fixed effects, the noise and

process variances, the spatial-temporal random effects and, if required, leave-one-out predictive densities.

Usage

```
stvcLMexact(
  formula,
  data = parent.frame(),
  sp_coords,
  time_coords,
  cor.fn,
  process.type,
  sptParams,
  noise_sp_ratio,
  priors = "flat",
  n.samples,
  loopd = FALSE,
  loopd.method = "exact",
  verbose = TRUE,
  ...
)
```

Arguments

formula	a symbolic description of the regression model to be fit. Variables in parenthesis are assigned spatially-temporally varying coefficients. See examples.
data	an optional data frame containing the variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> , typically the environment from which <code>stvcLMexact</code> is called.
sp_coords	an $n \times 2$ matrix of the observation spatial coordinates in $\mathbb{R}^2$ (e.g., easting and northing).
time_coords	an $n \times 1$ matrix of the observation temporal coordinates in $\mathcal{T} \subseteq [0, \infty)$.
cor.fn	a quoted keyword that specifies the correlation function used to model the spatial-temporal dependence structure among the observations. Supported covariance model key words are: 'gneiting-decay' (Gneiting and Guttorp 2010).
process.type	a quoted keyword specifying the model for the spatial-temporal processes of the varying coefficients. Supported keywords are 'independent', independent processes with their own process parameters and noise-to-spatial variance ratios, and 'independent.shared', independent processes that share common process parameters and a common noise-to-spatial variance ratio.
sptParams	fixed values of the spatial-temporal process parameters, a list with tags <code>phi_s</code> and <code>phi_t</code> . If <code>process.type = 'independent'</code> , each is a vector of length r , otherwise a scalar.
noise_sp_ratio	noise-to-spatial variance ratio(s) δ_j^2 : a vector of length r if <code>process.type = 'independent'</code> , otherwise a scalar. Default is 1.
priors	either "flat" (default), which assigns the prior $p(\beta, \sigma^2) \propto 1/\sigma^2$, or a list with tags <code>beta.norm</code> (a list containing μ_β and V_β) and/or <code>sigma.sq.ig</code> (a vector

	containing a_σ and b_σ). A component not supplied in the list receives its flat prior, $p(\beta) \propto 1$ or $p(\sigma^2) \propto 1/\sigma^2$.
n.samples	number of posterior samples to be generated.
loopd	logical. If loopd=TRUE, returns leave-one-out predictive densities, using method as given by loopd.method. Default is FALSE.
loopd.method	character. Ignored if loopd=FALSE. If loopd=TRUE, valid inputs are 'exact' and 'PSIS'. The option 'exact' finds the exact leave-one-out predictive densities in closed form, at the cost of about one additional $n \times n$ triangular inversion. The option 'PSIS' finds approximate leave-one-out predictive densities using Pareto-smoothed importance sampling (Vehtari <i>et al.</i> 2024); with many latent effects (nr), its Pareto k diagnostics are often high and 'exact' is recommended.
verbose	logical. If verbose = TRUE, prints model description.
...	currently no additional argument.

Details

Suppose $\chi = (\ell_1, \dots, \ell_n)$ denotes the n spatial-temporal co-ordinates in $\mathcal{L} = \mathcal{S} \times \mathcal{T}$ at which the response y is observed. With this function, we fit the conjugate Bayesian hierarchical model

$$\begin{aligned}
y(\ell) &= x(\ell)^\top \beta + \tilde{x}(\ell)^\top z(\ell) + \epsilon(\ell), \quad \epsilon(\ell) \sim N(0, \sigma^2), \\
z_j \mid \sigma^2 &\sim N(0, \sigma_{z_j}^2 R(\chi; \phi_{s,j}, \phi_{t,j})), \quad \sigma_{z_j}^2 = \sigma^2 / \delta_j^2, \quad j = 1, \dots, r, \\
\beta \mid \sigma^2 &\sim N(\mu_\beta, \sigma^2 V_\beta), \quad \sigma^2 \sim \text{IG}(a_\sigma, b_\sigma),
\end{aligned}$$

where $\tilde{x}(\ell)$ denotes the r covariates with spatially-temporally varying coefficients, the processes $z_1, \dots, z_r$ are independent, and $R(\chi; \phi_s, \phi_t)$ is the spatial-temporal correlation matrix of the Gneiting (2002) family. We fix the noise-to-spatial variance ratios $\delta_j^2 = \sigma^2 / \sigma_{z_j}^2$, the process parameters $\phi_{s,j}$ and $\phi_{t,j}$, and the hyperparameters μ_β , V_β , a_σ and b_σ . If process.type = 'independent', each process has its own $(\phi_{s,j}, \phi_{t,j}, \delta_j^2)$; if process.type = 'independent.shared', they share one. If priors = "flat", we instead assign the prior $p(\beta, \sigma^2) \propto 1/\sigma^2$.

The joint posterior distribution is available in closed form and is sampled exactly by composition,

$$p(\sigma^2, \beta, z \mid y) = p(\sigma^2 \mid y) \times p(\beta \mid \sigma^2, y) \times p(z \mid \beta, \sigma^2, y),$$

where $\sigma^2 \mid y$ is inverse-gamma and the other two are Gaussian. All of them depend on the data through the $n \times n$ matrix $V_y = I_n + \sum_j \delta_j^{-2} D_j R_j D_j$, with $D_j = \text{diag}(\tilde{x}_j)$. The nr -dimensional vector z is drawn by a prior draw followed by a kriging correction (Matheron's rule; Bhattacharya, Chakraborty and Mallick 2016), which needs only $n \times n$ Cholesky factorizations. Posterior samples of the process variances are obtained as $\sigma_{z_j}^2 = \sigma^2 / \delta_j^2$. The exact leave-one-out predictive densities are obtained in closed form from the same factorizations.

Value

An object of class stvcLMexact, which is a list with the following tags -

samples a list of length 4, containing posterior samples of fixed effects (beta, a $p \times \text{n.samples}$ matrix), the noise variance (sigmaSq), the process variances (sigmaSq.z, an $r \times \text{n.samples}$

matrix if `process.type = 'independent'`, otherwise a vector), and the spatial-temporal effects (z , an $nr \times n$.samples matrix whose rows $(j-1)n+1, \dots, jn$ correspond to the j -th varying coefficient).

loopd If `loopd=TRUE`, contains leave-one-out predictive densities.

model.params Values of the fixed parameters: `phi_s`, `phi_t` and `noise_sp_ratio`.

diagnostics a list of fit diagnostics, obtained from quantities the fit computes anyway. Element `numerical` is a data frame with one row (one row per process, if `process.type = 'independent'`) and columns `min.pivot` (the smallest relative Cholesky pivot of the spatial-temporal correlation matrix and of V_y ; values below $1e-8$ indicate a nearly singular matrix), `min.cor` and `max.cor` (the correlations of the two farthest-apart and of the two closest space-time locations; values of `min.cor` above 0.95 suggest an effective range far exceeding the extent of the data, values of `max.cor` below 0.05 nearly uncorrelated space-time locations). If `loopd.method = 'PSIS'`, element `pareto` is a list with the Pareto k diagnostic values (k), the threshold above which they are unreliable (`threshold`) and the number of values above it (`n.high`). If `verbose = TRUE`, a "Diagnostics" section is printed when any threshold is crossed.

The return object might include additional data used for subsequent prediction and/or model fit evaluation.

Author(s)

Soumyakanti Pan span18@ucla.edu,
Sudipto Banerjee sudipto@ucla.edu

References

- Bhattacharya A, Chakraborty A, Mallick BK (2016). "Fast sampling with Gaussian scale mixture priors in high-dimensional regression." *Biometrika*, **103**(4), 985-991. doi:[10.1093/biomet/asw042](https://doi.org/10.1093/biomet/asw042).
- Gneiting T (2002). "Nonseparable, Stationary Covariance Functions for Space-Time Data." *Journal of the American Statistical Association*, **97**(458), 590-600. doi:[10.1198/016214502760047113](https://doi.org/10.1198/016214502760047113).
- T. Gneiting and P. Guttorp (2010). "Continuous-parameter spatio-temporal processes." In A.E. Gelfand, P.J. Diggle, M. Fuentes, and P. Guttorp, editors, *Handbook of Spatial Statistics*, Chapman & Hall CRC Handbooks of Modern Statistical Methods, p 427-436. Taylor and Francis.
- Vehtari A, Simpson D, Gelman A, Yao Y, Gabry J (2024). "Pareto Smoothed Importance Sampling." *Journal of Machine Learning Research*, **25**(72), 1-58. URL <https://jmlr.org/papers/v25/19-556.html>.

See Also

[stvcLMstack\(\)](#), [stvcGLMexact\(\)](#), [spLMexact\(\)](#)

Examples

```
data(simSpaceTime)
dat <- simSpaceTime[1:100, ]

mod1 <- stvcLMexact(y_gauss ~ x1 + x2 + (x1), data = dat,
  sp_coords = as.matrix(dat[, c("s1", "s2")]),
```

```

time_coords = as.matrix(dat[, "t_coords"]),
cor.fn = "gneiting-decay",
process.type = "independent",
sptParams = list(phi_s = c(2, 2), phi_t = c(4, 4)),
noise_sp_ratio = c(0.1, 0.25),
n.samples = 500, loopd = TRUE, verbose = FALSE)

# rows 1, ..., n of z hold the varying intercept, rows n+1, ..., 2n the
# varying slope of x1
n <- nrow(dat)
z_hat <- apply(mod1$samples$z, 1, median)
cor(z_hat[1:n], dat$z1_true)

```

stvcLMstack	<i>Bayesian spatially-temporally varying coefficients linear model using predictive stacking</i>
-------------	--

Description

Fits Bayesian linear models with spatially-temporally varying coefficients for a Gaussian response on a collection of candidate models, constructed from candidate values of the spatial-temporal process parameters and the noise-to-spatial variance ratios supplied by the user, and combines inference by stacking predictive densities.

Usage

```

stvcLMstack(
  formula,
  data = parent.frame(),
  sp_coords,
  time_coords,
  cor.fn,
  process.type,
  priors = "flat",
  candidate.models,
  n.samples,
  loopd.method = "exact",
  parallel = FALSE,
  solver = NULL,
  verbose = TRUE,
  ...
)

```

Arguments

formula	a symbolic description of the regression model to be fit. Variables in parenthesis are assigned spatially-temporally varying coefficients. See examples.
---------	--

<code>data</code>	an optional data frame containing the variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> , typically the environment from which <code>stvcLMstack</code> is called.
<code>sp_coords</code>	an $n \times 2$ matrix of the observation spatial coordinates in $\mathbb{R}^2$ (e.g., easting and northing).
<code>time_coords</code>	an $n \times 1$ matrix of the observation temporal coordinates in $\mathcal{T} \subseteq [0, \infty)$.
<code>cor.fn</code>	a quoted keyword that specifies the correlation function used to model the spatial-temporal dependence structure among the observations. Supported covariance model key words are: 'gneiting-decay' (Gneiting and Guttorp 2010).
<code>process.type</code>	a quoted keyword specifying the model for the spatial-temporal processes of the varying coefficients: 'independent' or 'independent.shared'. See <code>stvcLMexact()</code> .
<code>priors</code>	either "flat" (default), which assigns the prior $p(\beta, \sigma^2) \propto 1/\sigma^2$, or a list with tags <code>beta.norm</code> (a list containing μ_β and V_β) and/or <code>sigma.sq.ig</code> (a vector containing a_σ and b_σ). A component not supplied in the list receives its flat prior.
<code>candidate.models</code>	an object of class <code>candidateModels</code> containing a list of candidate models for stacking, each with tags <code>phi_s</code> , <code>phi_t</code> and <code>noise_sp_ratio</code> : vectors of length r if <code>process.type = 'independent'</code> (use <code>list()</code> entries in <code>candidateModels()</code>), otherwise scalars. See <code>candidateModels()</code> for details.
<code>n.samples</code>	number of posterior samples to be generated.
<code>loopd.method</code>	character. Valid inputs are 'exact' (default) and 'PSIS'. The option 'exact' finds the exact leave-one-out predictive densities in closed form. The option 'PSIS' uses Pareto-smoothed importance sampling (Vehtari <i>et al.</i> 2024); with many latent effects its Pareto k diagnostics are often high, so 'exact' is recommended.
<code>parallel</code>	logical. If <code>parallel=FALSE</code> , the parallelization plan, if set up by the user, is ignored. If <code>parallel=TRUE</code> , the function inherits the parallelization plan that is set by the user via the function <code>future::plan()</code> only. Depending on the parallel backend available, users may choose their own plan. More details are available at https://cran.R-project.org/package=future .
<code>solver</code>	(optional) Specifies the name of the solver that will be used to obtain optimal stacking weights for each candidate model. Default order is <code>c("CLARABEL", "ECOS", "SCS")</code> . Users can use other solvers supported by the <code>CVXR-package</code> .
<code>verbose</code>	logical. If <code>TRUE</code> , prints model-specific optimal stacking weights.
<code>...</code>	currently no additional argument.

Details

Instead of assigning priors on the process parameters ϕ_s, ϕ_t and the noise-to-spatial variance ratios δ^2 , we consider a set of candidate models $\mathcal{M} = \{M_1, \dots, M_G\}$ based on candidate values of these parameters. For each g , we sample exactly from the posterior distribution $p(\sigma^2, \beta, z \mid y, M_g)$ (see

`stvcLMexact()` and find the leave-one-out predictive densities $p(y_i | y_{-i}, M_g)$. The stacking weights solve

$$\begin{aligned} \max_{w_1, \dots, w_G} \quad & \frac{1}{n} \sum_{i=1}^n \log \sum_{g=1}^G w_g p(y_i | y_{-i}, M_g) \\ \text{subject to} \quad & w_g \geq 0, \sum_{g=1}^G w_g = 1. \end{aligned}$$

Candidate models that share the process parameters (ϕ_s, ϕ_t) are fitted together, building and factorizing the spatial-temporal correlation matrices only once.

Value

An object of class `stvcLMstack`, which is a list including the following tags -

`samples` a list of length equal to total number of candidate models with each entry corresponding to a list of length 4, containing posterior samples of fixed effects (`beta`), the noise variance (`sigmaSq`), the process variances (`sigmaSq.z`) and the spatial-temporal effects (`z`) for that model, as in `stvcLMexact()`.

`loopd` a list of length equal to total number of candidate models with each entry containing leave-one-out predictive densities under that particular model.

`n.models` number of candidate models that are fit.

`model.params` a list with one element per candidate model, each a named list of its parameters: `phi_s`, `phi_t` and `noise_sp_ratio` (vectors of length r if `process.type = 'independent'`).

`stacking.summary` a matrix with one row per candidate model, containing its parameters and its optimal stacking weight, for display.

`stacking.weights` a numeric vector of length equal to the number of candidate models storing the optimal stacking weights.

`run.time` a `proc_time` object with runtime details.

`diagnostics` a list of diagnostics. Element `numerical` is a data frame with one row per candidate model (per candidate model and process, with columns `model` and `process`, if `process.type = 'independent'`) and columns `min.pivot`, `min.cor` and `max.cor`, as in `stvcLMexact()`. If `loopd.method = 'PSIS'`, element `pareto` contains the Pareto k diagnostic values of each candidate model (k), the threshold above which they are unreliable (`threshold`) and the number of values above it for each model (`n.high`). Element `solver` describes the optimization for the stacking weights: the solver used (`used`) and its status (`status`), the installed and requested solvers, the search order, the attempts with their status, and whether the fallback `loo::stacking_weights()` was used. If `verbose = TRUE`, a "Diagnostics" section is printed if there is an issue: numerical flags of candidate models with stacking weight above 0.05 (extreme candidates with negligible weight are expected in a stacking grid and are only counted), Pareto k values above the threshold for any model, and solver problems.

This object can be used to make predictions at new locations or times with `posteriorPredict()` and to sample from the stacked posterior with `stackedSampler()`.

Author(s)

Soumyakanti Pan span18@ucla.edu,
Sudipto Banerjee sudipto@ucla.edu

References

- Vehtari A, Simpson D, Gelman A, Yao Y, Gabry J (2024). "Pareto Smoothed Importance Sampling." *Journal of Machine Learning Research*, **25**(72), 1-58. URL <https://jmlr.org/papers/v25/19-556.html>.
- Zhang L, Tang W, Banerjee S (2025). "Bayesian Geostatistics Using Predictive Stacking." *Journal of the American Statistical Association*, **In press**. doi:10.1080/01621459.2025.2566449.

See Also

[stvcLMexact\(\)](#), [stvcGLMstack\(\)](#), [sPLMstack\(\)](#)

Examples

```
set.seed(1234)
data(simSpaceTime)
dat <- simSpaceTime[1:100, ]

# processes sharing (phi_s, phi_t, noise_sp_ratio): scalar candidates
mod.list <- candidateModels(list(phi_s = c(1, 2, 3), phi_t = c(2, 4, 8),
                                noise_sp_ratio = c(0.1, 0.25)), "cartesian")
mod1 <- stvcLMstack(y_gauss ~ x1 + x2 + (x1), data = dat,
                   sp_coords = as.matrix(dat[, c("s1", "s2")]),
                   time_coords = as.matrix(dat[, "t_coords"]),
                   cor.fn = "gneiting-decay",
                   process.type = "independent.shared",
                   candidate.models = mod.list,
                   n.samples = 500)
post_samps <- stackedSampler(mod1)
n <- nrow(dat)
cor(apply(post_samps$z[1:n, ], 1, median), dat$z1_true)

# independent processes (r = 2): vector-valued candidates via list()
mod.list2 <- candidateModels(list(phi_s = list(c(2, 2), c(3, 3)),
                                   phi_t = list(c(4, 4), c(8, 8)),
                                   noise_sp_ratio = list(c(0.1, 0.25), c(0.25, 0.5))),
                              "cartesian")
mod2 <- stvcLMstack(y_gauss ~ x1 + x2 + (x1), data = dat,
                   sp_coords = as.matrix(dat[, c("s1", "s2")]),
                   time_coords = as.matrix(dat[, "t_coords"]),
                   cor.fn = "gneiting-decay",
                   process.type = "independent",
                   candidate.models = mod.list2,
                   n.samples = 500, verbose = FALSE)
```

surfaceplot

Make a surface plot

Description

Make a surface plot

Usage

```
surfaceplot(
  tab,
  coords_name,
  var_name,
  h = 8,
  col.pal,
  mark_points = FALSE,
  title = NULL
)
```

Arguments

tab	a data-frame containing spatial co-ordinates and the variable to plot
coords_name	name of the two columns that contains the co-ordinates of the points, also used as the axis labels
var_name	name of the column containing the variable to be plotted
h	integer; (optional) controls smoothness of the spatial interpolation as appearing in the MBA::mba.surf() function. Default is 8.
col.pal	Optional; color palette, preferably divergent, use colorRampPalette function from grDevices. Default is the colorblind-friendly diverging palette 'RdBu' from ColorBrewer.
mark_points	Logical; if TRUE, the input points are marked. Default is FALSE.
title	(optional) a title for the plot. Default is no title.

Value

a ggplot object containing the surface plot

Author(s)

Soumyakanti Pan span18@ucla.edu,
Sudipto Banerjee sudipto@ucla.edu

Examples

```
data(simSpatial)
plot1 <- surfaceplot(simSpatial, coords_name = c("s1", "s2"),
                     var_name = "z_true")
plot1

# try your favourite color palette
plot2 <- surfaceplot(simSpatial, coords_name = c("s1", "s2"),
                     var_name = "z_true",
                     col.pal = hcl.colors(100, "PuOr", rev = TRUE))
plot2
```

surfaceplot2

Make two side-by-side surface plots

Description

Make two side-by-side surface plots, particularly useful towards a comparative study of two spatial surfaces.

Usage

```
surfaceplot2(
  tab,
  coords_name,
  var1_name,
  var2_name,
  h = 8,
  col.pal,
  mark_points = FALSE,
  titles = c(var1_name, var2_name)
)
```

Arguments

tab	a data-frame containing spatial co-ordinates and the variables to plot
coords_name	name of the two columns that contains the co-ordinates of the points, also used as the axis labels
var1_name	name of the column containing the first variable to be plotted
var2_name	name of the column containing the second variable to be plotted
h	integer; (optional) controls smoothness of the spatial interpolation as appearing in the MBA::mba.surf() function. Default is 8.
col.pal	Optional; color palette, preferably divergent, use <code>colorRampPalette</code> function from <code>grDevices</code> . Default is the colorblind-friendly diverging palette 'RdBu' from <code>ColorBrewer</code> .
mark_points	Logical; if TRUE, the input points are marked. Default is FALSE.
titles	a character vector of length 2, the titles of the two plots. Default is the names of the two variables, <code>c(var1_name, var2_name)</code> ; use NULL for no titles.

Value

a list containing two ggplot objects, which share the colour scale (one legend serves both, e.g. with `patchwork::wrap_plots(plots) + patchwork::plot_layout(guides = "collect")`)

Author(s)

Soumyakanti Pan span18@ucla.edu,
 Sudipto Banerjee sudipto@ucla.edu

Examples

```
data(simSpatial)
plots_2 <- surfaceplot2(simSpatial, coords_name = c("s1", "s2"),
                        var1_name = "z_true", var2_name = "y_gauss",
                        titles = c("True spatial effect", "Gaussian response"))

plots_2
```

 svcGLMexact

Bayesian spatially varying coefficients generalized linear model

Description

Fits a Bayesian generalized linear model with spatially varying coefficients under fixed values of the spatial process parameters and some auxiliary model parameters. The output contains posterior samples of the fixed effects, the spatially varying coefficients and, if required, finds leave-one-out predictive densities.

Usage

```
svcGLMexact(
  formula,
  data = parent.frame(),
  family,
  coords,
  cor.fn,
  process.type,
  spParams,
  priors,
  boundary = 0.5,
  n.samples,
  loopd = FALSE,
  loopd.method = "exact",
  CV.K = 10,
  loopd.nMC = 500,
  verbose = TRUE,
  ...
)
```

Arguments

formula	a symbolic description of the regression model to be fit. Variables in parenthesis are assigned spatially varying coefficients. See examples.
data	an optional data frame containing the variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> , typically the environment from which <code>svcGLMexact</code> is called.
family	Specifies the distribution of the response as a member of the exponential family. Supported options are 'poisson', 'binomial' and 'binary'.

<code>coords</code>	an $n \times 2$ matrix of the observation coordinates in $\mathbb{R}^2$ (e.g., easting and northing).
<code>cor.fn</code>	a quoted keyword that specifies the correlation function used to model the spatial dependence structure among the observations. Supported covariance model keywords are: 'exponential' and 'matern'.
<code>process.type</code>	a quoted keyword specifying the model for the spatial processes: 'independent', 'independent.shared' or 'multivariate'. See details.
<code>spParams</code>	fixed values of the spatial process parameters, a list with tags <code>phi</code> (spatial decay) and, if <code>cor.fn = 'matern'</code> , <code>nu</code> (smoothness). If <code>process.type = 'independent'</code> , each is a vector of length r , otherwise a scalar.
<code>priors</code>	(optional) a list with each tag corresponding to a hyperparameter name and containing hyperprior details. Valid tags include <code>V.beta</code> , <code>nu.beta</code> , <code>nu.z</code> , <code>sigmaSq.xi</code> and <code>IW.scale</code> . Values of <code>nu.beta</code> and <code>nu.z</code> must be at least 2.1. If not supplied, uses defaults.
<code>boundary</code>	Specifies the boundary adjustment parameter. Must be a real number between 0 and 1. Default is 0.5.
<code>n.samples</code>	number of posterior samples to be generated.
<code>loopd</code>	logical. If <code>loopd=TRUE</code> , returns leave-one-out predictive densities, using method as given by <code>loopd.method</code> . Default is FALSE.
<code>loopd.method</code>	character. Ignored if <code>loopd=FALSE</code> . If <code>loopd=TRUE</code> , valid inputs are 'exact', 'CV'. The option 'exact' corresponds to exact leave-one-out predictive densities which requires computation almost equivalent to fitting the model n times. The options 'CV' is faster as it implements K -fold cross validation to find approximate leave-one-out predictive densities (Vehtari <i>et al.</i> 2017).
<code>CV.K</code>	An integer between 10 and 20. Considered only if <code>loopd.method='CV'</code> . Default is 10 (as recommended in Vehtari <i>et. al</i> 2017).
<code>loopd.nMC</code>	Number of Monte Carlo samples to be used to evaluate leave-one-out predictive densities when <code>loopd.method</code> is set to either 'exact' or 'CV'.
<code>verbose</code>	logical. If <code>verbose = TRUE</code> , prints model description.
<code>...</code>	currently no additional argument.

Details

With this function, we fit a Bayesian hierarchical spatially varying coefficients generalized linear model by sampling exactly from the joint posterior distribution utilizing the generalized conjugate multivariate distribution theory (Bradley and Clinch 2024). Suppose $\chi = (s_1, \dots, s_n)$ denotes the n spatial locations in $\mathcal{S} \subset \mathbb{R}^2$ at which the response y is observed, with $y(s)$ endowed with a probability law from the natural exponential family

$$y(s) \sim \text{EF}(x(s)^\top \beta + \tilde{x}(s)^\top z(s); b(s), \psi)$$

for some positive parameter $b(s) > 0$ and unit log partition function ψ . Here, $\tilde{x}(s)$ denotes the r covariates with spatially varying coefficients. The supported families ('poisson', 'binomial' and 'binary') and the hierarchical model are those of `stvcGLMexact()`, with

$$z_j \mid \sigma_{z_j}^2 \sim N(0, \sigma_{z_j}^2 R(\chi; \phi_j, \nu_j)), \quad \sigma_{z_j}^2 \sim \text{IG}(\nu_z/2, \nu_z/2), \quad j = 1, \dots, r,$$

where $R(\chi; \phi, \nu)$ is the exponential or Matern spatial correlation matrix with decay ϕ and smoothness ν . If `process.type = 'independent'`, each process has its own (ϕ_j, ν_j) ; if `process.type = 'independent.shared'`, they share one. If `process.type = 'multivariate'`, $z = (z_1, \dots, z_r)$ is a multivariate Gaussian process with correlation matrix $R(\chi; \phi, \nu)$ and an $r \times r$ cross-covariance matrix with an inverse-Wishart prior. We fix the process parameters and the hyperparameters V_β , ν_β , ν_z and σ_ξ^2 ; their defaults are as in `stvcGLMexact()`.

Value

An object of class `svcGLMexact`, which is a list with the following tags -

priors details of the priors used, containing the values of the boundary adjustment parameter (boundary), the variance parameter of the fine-scale variation term (`sigmaSq.xi`) and others.

samples a list of length 3, containing posterior samples of fixed effects (beta), the spatially varying coefficients (z, an $nr \times n$.samples matrix whose rows $(j-1)n+1, \dots, jn$ correspond to the j -th varying coefficient) and the fine-scale variation term (xi).

loopd If `loopd=TRUE`, contains leave-one-out predictive densities.

model.params Values of the fixed parameters: phi and nu (NA for the exponential correlation function).

diagnostics a list of diagnostics. Element `numerical` is a data frame with one row (one row per process, if `process.type = 'independent'`) and columns `min.pivot` (the smallest relative Cholesky pivot of the $n \times n$ spatial correlation matrix; values below $1e-8$ indicate a nearly singular correlation matrix), `min.cor` and `max.cor` (the correlations of the two farthest-apart and of the two closest locations; values of `min.cor` above 0.95 suggest an effective range far exceeding the extent of the data, values of `max.cor` below 0.05 nearly uncorrelated locations). If `verbose = TRUE`, a "Diagnostics" section is printed when any threshold is crossed.

The return object might include additional data that can be used for subsequent prediction and/or model fit evaluation.

Author(s)

Soumyakanti Pan span18@ucla.edu,
Sudipto Banerjee sudipto@ucla.edu

References

- Bradley JR, Clinch M (2024). "Generating Independent Replicates Directly from the Posterior Distribution for a Class of Spatial Hierarchical Models." *Journal of Computational and Graphical Statistics*, **0**(0), 1-17. doi:[10.1080/10618600.2024.2365728](https://doi.org/10.1080/10618600.2024.2365728).
- Pan S, Zhang L, Bradley JR, Banerjee S (2025). "Bayesian Inference for Spatial-temporal Non-Gaussian Data Using Predictive Stacking." *Bayesian Analysis*, **In Press**. doi:[10.1214/25BA1582](https://doi.org/10.1214/25BA1582).
- Vehtari A, Gelman A, Gabry J (2017). "Practical Bayesian Model Evaluation Using Leave-One-out Cross-Validation and WAIC." *Statistics and Computing*, **27**(5), 1413-1432. ISSN 0960-3174. doi:[10.1007/s1122201696964](https://doi.org/10.1007/s1122201696964).

See Also

[svcGLMstack\(\)](#), [svclMexact\(\)](#), [spGLMexact\(\)](#), [stvcGLMexact\(\)](#)

Examples

```
data(simSpatial)
dat <- simSpatial[1:150, ]

mod1 <- svcGLMexact(y_svc_pois ~ x1 + x2 + (x1), data = dat,
  family = "poisson",
  coords = as.matrix(dat[, c("s1", "s2")]),
  cor.fn = "matern", process.type = "independent",
  spParams = list(phi = c(8, 4), nu = c(1.5, 1.5)),
  n.samples = 100, verbose = FALSE)

# rows 1, ..., n of z hold the varying intercept, rows n+1, ..., 2n the
# varying slope of x1
n <- nrow(dat)
z_hat <- apply(mod1$samples$z, 1, median)
cor(z_hat[n + 1:n], dat$z_true)
```

svcGLMstack

Bayesian spatially varying coefficients generalized linear model using predictive stacking

Description

Fits Bayesian generalized linear models with spatially varying coefficients on a collection of candidate models constructed from candidate values of the spatial process parameters and the boundary adjustment parameter supplied by the user, and combines inference by stacking predictive densities. See Pan, Zhang, Bradley, and Banerjee (2025) for more details.

Usage

```
svcGLMstack(
  formula,
  data = parent.frame(),
  family,
  coords,
  cor.fn,
  process.type,
  priors,
  candidate.models,
  n.samples,
  loopd.controls,
  parallel = FALSE,
  solver = NULL,
  verbose = TRUE,
```

...
)

Arguments

formula	a symbolic description of the regression model to be fit. Variables in parenthesis are assigned spatially varying coefficients. See examples.
data	an optional data frame containing the variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> , typically the environment from which <code>svcGLMstack</code> is called.
family	Specifies the distribution of the response as a member of the exponential family. Supported options are 'poisson', 'binomial' and 'binary'.
coords	an $n \times 2$ matrix of the observation coordinates in $\mathbb{R}^2$ (e.g., easting and northing).
cor.fn	a quoted keyword that specifies the correlation function used to model the spatial dependence structure among the observations. Supported covariance model key words are: 'exponential' and 'matern'.
process.type	a quoted keyword specifying the model for the spatial processes: 'independent', 'independent.shared' or 'multivariate'. See svcGLMexact() .
priors	(optional) a list with each tag corresponding to a hyperparameter name and containing hyperprior details. Valid tags include <code>V.beta</code> , <code>nu.beta</code> , <code>nu.z</code> , <code>sigmaSq.xi</code> and <code>IW.scale</code> . See svcGLMexact() . If not supplied, uses defaults.
candidate.models	an object of class <code>candidateModels</code> containing a list of candidate models for stacking, each with tags <code>phi</code> , <code>nu</code> (only if <code>cor.fn = 'matern'</code>), vectors of length r if <code>process.type = 'independent'</code> (use <code>list()</code> entries in candidateModels()) and scalars otherwise, and <code>boundary</code> , a scalar. See candidateModels() for details.
n.samples	number of samples to be drawn from the posterior distribution.
loopd.controls	a list with details on how leave-one-out predictive densities (LOO-PD) are to be calculated, with tags <code>method</code> ('exact' or 'CV'), <code>CV.K</code> , <code>nMC</code> and <code>CV.update</code> , as in stvcGLMstack() . If sample size is more than 100, then the default is 'CV' with <code>CV.K = 10</code> .
parallel	logical. If <code>parallel=FALSE</code> , the parallelization plan, if set up by the user, is ignored. If <code>parallel=TRUE</code> , the function inherits the parallelization plan that is set by the user via the function future::plan() only. Depending on the parallel backend available, users may choose their own plan. More details are available at https://cran.R-project.org/package=future .
solver	(optional) Specifies the name of the solver that will be used to obtain optimal stacking weights for each candidate model. Default order is <code>c("CLARABEL", "ECOS", "SCS")</code> . Users can use other solvers supported by the CVXR-package package.
verbose	logical. If <code>TRUE</code> , prints model-specific optimal stacking weights.
...	currently no additional argument.

Details

Candidate models that share the process parameters (ϕ, ν) are fitted together: the spatial correlation matrices and all the pre-processing for the leave-one-out predictive densities are computed once and shared across their boundary values, and the groups of such candidates are the units of parallel work.

Value

An object of class `svcGLMstack`, which is a list including the following tags -

`samples` a list of length equal to total number of candidate models with each entry corresponding to a list of length 3, containing posterior samples of fixed effects (`beta`), the spatially varying coefficients (`z`), and fine scale variation `xi` for that model.

`loopd` a list of length equal to total number of candidate models with each entry containing leave-one-out predictive densities under that particular model.

`n.models` number of candidate models that are fit.

`model.params` a list with one element per candidate model, each a named list of its parameters: `phi`, `nu` (NA for the exponential correlation function) and `boundary`.

`stacking.summary` a matrix with one row per candidate model, containing its parameters and its optimal stacking weight, for display.

`stacking.weights` a numeric vector of length equal to the number of candidate models storing the optimal stacking weights.

`run.time` a `proc_time` object with runtime details.

`diagnostics` a list of diagnostics, as in `stvcGLMstack()`: element `numerical` (columns `min.pivot`, `min.cor` and `max.cor`, see `svcGLMexact()`) and element `solver`. If `verbose = TRUE`, a "Diagnostics" section is printed if there is an issue.

This object can be further used to recover posterior samples of the scale parameters in the model with `recoverGLMscale()`, to make predictions at new locations with `posteriorPredict()`, and to sample from the stacked posterior with `stackedSampler()`.

Author(s)

Soumyakanti Pan span18@ucla.edu,
Sudipto Banerjee sudipto@ucla.edu

References

Pan S, Zhang L, Bradley JR, Banerjee S (2025). "Bayesian Inference for Spatial-temporal Non-Gaussian Data Using Predictive Stacking." *Bayesian Analysis*, **In Press**. doi:10.1214/25BA1582.

See Also

[svcGLMexact\(\)](#), [svGLMstack\(\)](#), [spGLMstack\(\)](#), [stvcGLMstack\(\)](#)

Examples

```
set.seed(1234)
data(simSpatial)
dat <- simSpatial[1:100, ]

# processes sharing (phi, nu): scalar candidates
mod.list <- candidateModels(list(phi = c(4, 8), nu = c(1.5),
                                boundary = c(0.5, 0.75)), "cartesian")
mod1 <- svcGLMstack(y_svc_pois ~ x1 + x2 + (x1), data = dat,
                   family = "poisson",
                   coords = as.matrix(dat[, c("s1", "s2")]),
                   cor.fn = "matern", process.type = "independent.shared",
                   candidate.models = mod.list,
                   loopd.controls = list(method = "CV", CV.K = 10, nMC = 500),
                   n.samples = 500)
post_samps <- stackedSampler(mod1)
```

svcLMexact

Bayesian spatially varying coefficients linear model

Description

Fits a Bayesian linear model with spatially varying coefficients for a Gaussian response, with the spatial process parameters and the noise-to-spatial variance ratios fixed to values supplied by the user. The output contains exact posterior samples of the fixed effects, the noise and process variances, the spatially varying coefficients and, if required, leave-one-out predictive densities.

Usage

```
svcLMexact(
  formula,
  data = parent.frame(),
  coords,
  cor.fn,
  process.type,
  spParams,
  noise_sp_ratio,
  priors = "flat",
  n.samples,
  loopd = FALSE,
  loopd.method = "exact",
  verbose = TRUE,
  ...
)
```

Arguments

<code>formula</code>	a symbolic description of the regression model to be fit. Variables in parenthesis are assigned spatially varying coefficients. See examples.
<code>data</code>	an optional data frame containing the variables in the model. If not found in <code>data</code> , the variables are taken from <code>environment(formula)</code> , typically the environment from which <code>svcLMexact</code> is called.
<code>coords</code>	an $n \times 2$ matrix of the observation coordinates in $\mathbb{R}^2$ (e.g., easting and northing).
<code>cor.fn</code>	a quoted keyword that specifies the correlation function used to model the spatial dependence structure among the observations. Supported covariance model keywords are: 'exponential' and 'matern'. See below for details.
<code>process.type</code>	a quoted keyword specifying the model for the spatial processes of the varying coefficients. Supported keywords are 'independent', independent processes with their own process parameters and noise-to-spatial variance ratios, and 'independent.shared', independent processes that share common process parameters and a common noise-to-spatial variance ratio.
<code>spParams</code>	fixed values of the spatial process parameters, a list with tags <code>phi</code> (spatial decay) and, if <code>cor.fn = 'matern'</code> , <code>nu</code> (smoothness). If <code>process.type = 'independent'</code> , each is a vector of length r , otherwise a scalar.
<code>noise_sp_ratio</code>	noise-to-spatial variance ratio(s) δ_j^2 : a vector of length r if <code>process.type = 'independent'</code> , otherwise a scalar. Default is 1.
<code>priors</code>	either "flat" (default), which assigns the prior $p(\beta, \sigma^2) \propto 1/\sigma^2$, or a list with tags <code>beta.norm</code> (a list containing μ_β and V_β) and/or <code>sigma.sq.ig</code> (a vector containing a_σ and b_σ). A component not supplied in the list receives its flat prior, $p(\beta) \propto 1$ or $p(\sigma^2) \propto 1/\sigma^2$.
<code>n.samples</code>	number of posterior samples to be generated.
<code>loopd</code>	logical. If <code>loopd=TRUE</code> , returns leave-one-out predictive densities, using method as given by <code>loopd.method</code> . Default is FALSE.
<code>loopd.method</code>	character. Ignored if <code>loopd=FALSE</code> . If <code>loopd=TRUE</code> , valid inputs are 'exact' and 'PSIS'. The option 'exact' finds the exact leave-one-out predictive densities in closed form, at the cost of about one additional $n \times n$ triangular inversion. The option 'PSIS' finds approximate leave-one-out predictive densities using Pareto-smoothed importance sampling (Vehtari <i>et al.</i> 2024); with many latent effects (nr), its Pareto k diagnostics are often high and 'exact' is recommended.
<code>verbose</code>	logical. If <code>verbose = TRUE</code> , prints model description.
<code>...</code>	currently no additional argument.

Details

Suppose $\chi = (s_1, \dots, s_n)$ denotes the n spatial locations in $\mathcal{S} \subset \mathbb{R}^2$ at which the response y is observed. With this function, we fit the conjugate Bayesian hierarchical model

$$\begin{aligned}
 y(s) &= x(s)^\top \beta + \tilde{x}(s)^\top z(s) + \epsilon(s), \quad \epsilon(s) \sim N(0, \sigma^2), \\
 z_j \mid \sigma^2 &\sim N(0, \sigma_{z_j}^2 R(\chi; \phi_j, \nu_j)), \quad \sigma_{z_j}^2 = \sigma^2 / \delta_j^2, \quad j = 1, \dots, r, \\
 \beta \mid \sigma^2 &\sim N(\mu_\beta, \sigma^2 V_\beta), \quad \sigma^2 \sim \text{IG}(a_\sigma, b_\sigma),
 \end{aligned}$$

where $\tilde{x}(s)$ denotes the r covariates with spatially varying coefficients, the processes $z_1, \dots, z_r$ are independent, and $R(\chi; \phi, \nu)$ is the exponential or Matern spatial correlation matrix with decay ϕ and smoothness ν . We fix the noise-to-spatial variance ratios $\delta_j^2 = \sigma^2 / \sigma_{z_j}^2$, the process parameters ϕ_j and ν_j , and the hyperparameters μ_β , V_β , a_σ and b_σ . If `process.type = 'independent'`, each process has its own $(\phi_j, \nu_j, \delta_j^2)$; if `process.type = 'independent.shared'`, they share one. If `priors = "flat"`, we instead assign the prior $p(\beta, \sigma^2) \propto 1/\sigma^2$.

The joint posterior distribution is available in closed form and is sampled exactly by composition,

$$p(\sigma^2, \beta, z \mid y) = p(\sigma^2 \mid y) \times p(\beta \mid \sigma^2, y) \times p(z \mid \beta, \sigma^2, y),$$

where $\sigma^2 \mid y$ is inverse-gamma and the other two are Gaussian. All of them depend on the data through the $n \times n$ matrix $V_y = I_n + \sum_j \delta_j^{-2} D_j R_j D_j$, with $D_j = \text{diag}(\tilde{x}_j)$. The nr -dimensional vector z is drawn by a prior draw followed by a kriging correction (Matheron's rule; Bhattacharya, Chakraborty and Mallick 2016), which needs only $n \times n$ Cholesky factorizations. The exact leave-one-out predictive densities are obtained in closed form from the same factorizations. The computations are those of `stvclMexact()`, with a spatial correlation function in place of the spatial-temporal one.

Value

An object of class `svcLMexact`, which is a list with the following tags -

samples a list of length 4, containing posterior samples of fixed effects (beta, a $p \times n$.samples matrix), the noise variance (sigmaSq), the process variances (sigmaSq.z, an $r \times n$.samples matrix if `process.type = 'independent'`, otherwise a vector), and the spatially varying coefficients (z, an $nr \times n$.samples matrix whose rows $(j-1)n+1, \dots, jn$ correspond to the j -th varying coefficient).

loopd If `loopd=TRUE`, contains leave-one-out predictive densities.

model.params Values of the fixed parameters: phi, nu (NA for the exponential correlation function) and noise_sp_ratio.

diagnostics a list of fit diagnostics, obtained from quantities the fit computes anyway. Element `numerical` is a data frame with one row (one row per process, if `process.type = 'independent'`) and columns `min.pivot` (the smallest relative Cholesky pivot of the spatial correlation matrix and of V_y ; values below $1e-8$ indicate a nearly singular matrix), `min.cor` and `max.cor` (the correlations of the two farthest-apart and of the two closest locations; values of `min.cor` above 0.95 suggest an effective range far exceeding the extent of the data, values of `max.cor` below 0.05 nearly uncorrelated locations). If `loopd.method = 'PSIS'`, element `pareto` is a list with the Pareto k diagnostic values (`k`), the threshold above which they are unreliable (`threshold`) and the number of values above it (`n.high`). If `verbose = TRUE`, a "Diagnostics" section is printed when any threshold is crossed.

The return object might include additional data used for subsequent prediction and/or model fit evaluation.

Author(s)

Soumyakanti Pan span18@ucla.edu,
Sudipto Banerjee sudipto@ucla.edu

References

- Bhattacharya A, Chakraborty A, Mallick BK (2016). "Fast sampling with Gaussian scale mixture priors in high-dimensional regression." *Biometrika*, **103**(4), 985-991. doi:[10.1093/biomet/asw042](https://doi.org/10.1093/biomet/asw042).
- Vehtari A, Simpson D, Gelman A, Yao Y, Gabry J (2024). "Pareto Smoothed Importance Sampling." *Journal of Machine Learning Research*, **25**(72), 1-58. URL <https://jmlr.org/papers/v25/19-556.html>.

See Also

[svcLMstack\(\)](#), [svcGLMexact\(\)](#), [splMexact\(\)](#), [stvcLMexact\(\)](#)

Examples

```
data(simSpatial)
dat <- simSpatial[1:200, ]

# varying intercept and varying slope of x1, each with its own parameters
mod1 <- svcLMexact(y_svc ~ x1 + x2 + (x1), data = dat,
                  coords = as.matrix(dat[, c("s1", "s2")]),
                  cor.fn = "matern", process.type = "independent",
                  spParams = list(phi = c(8, 4), nu = c(1.5, 1.5)),
                  noise_sp_ratio = c(0.25, 0.25),
                  n.samples = 500, loopd = TRUE, verbose = FALSE)

# rows 1, ..., n of z hold the varying intercept, rows n+1, ..., 2n the
# varying slope of x1
n <- nrow(dat)
z_hat <- apply(mod1$samples$z, 1, median)
cor(z_hat[1:n], dat$z2_true)
cor(z_hat[n + 1:n], dat$z_true)
```

svcLMstack

Bayesian spatially varying coefficients linear model using predictive stacking

Description

Fits Bayesian linear models with spatially varying coefficients for a Gaussian response on a collection of candidate models, constructed from candidate values of the spatial process parameters and the noise-to-spatial variance ratios supplied by the user, and combines inference by stacking predictive densities.

Usage

```
svcLMstack(
  formula,
  data = parent.frame(),
  coords,
```

```

cor.fn,
process.type,
priors = "flat",
candidate.models,
n.samples,
loopd.method = "exact",
parallel = FALSE,
solver = NULL,
verbose = TRUE,
...
)

```

Arguments

formula	a symbolic description of the regression model to be fit. Variables in parenthesis are assigned spatially varying coefficients. See examples.
data	an optional data frame containing the variables in the model. If not found in data, the variables are taken from <code>environment(formula)</code> , typically the environment from which <code>svcLMstack</code> is called.
coords	an $n \times 2$ matrix of the observation coordinates in $\mathbb{R}^2$ (e.g., easting and northing).
cor.fn	a quoted keyword that specifies the correlation function used to model the spatial dependence structure among the observations. Supported covariance model keywords are: 'exponential' and 'matern'.
process.type	a quoted keyword specifying the model for the spatial processes of the varying coefficients: 'independent' or 'independent.shared'. See <code>svcLMexact()</code> .
priors	either "flat" (default), which assigns the prior $p(\beta, \sigma^2) \propto 1/\sigma^2$, or a list with tags <code>beta.norm</code> (a list containing μ_β and V_β) and/or <code>sigma.sq.ig</code> (a vector containing a_σ and b_σ). A component not supplied in the list receives its flat prior.
candidate.models	an object of class <code>candidateModels</code> containing a list of candidate models for stacking, each with tags <code>phi</code> , <code>nu</code> (only if <code>cor.fn = 'matern'</code>) and <code>noise_sp_ratio</code> : vectors of length r if <code>process.type = 'independent'</code> (use <code>list()</code> entries in <code>candidateModels()</code>), otherwise scalars. See <code>candidateModels()</code> for details.
n.samples	number of posterior samples to be generated.
loopd.method	character. Valid inputs are 'exact' (default) and 'PSIS'. The option 'exact' finds the exact leave-one-out predictive densities in closed form. The option 'PSIS' uses Pareto-smoothed importance sampling (Vehtari <i>et al.</i> 2024); with many latent effects its Pareto k diagnostics are often high, so 'exact' is recommended.
parallel	logical. If <code>parallel=FALSE</code> , the parallelization plan, if set up by the user, is ignored. If <code>parallel=TRUE</code> , the function inherits the parallelization plan that is set by the user via the function <code>future::plan()</code> only. Depending on the parallel backend available, users may choose their own plan. More details are available at https://cran.R-project.org/package=future .

solver	(optional) Specifies the name of the solver that will be used to obtain optimal stacking weights for each candidate model. Default order is <code>c("CLARABEL", "ECOS", "SCS")</code> . Users can use other solvers supported by the CVXR-package package.
verbose	logical. If TRUE, prints model-specific optimal stacking weights.
...	currently no additional argument.

Details

Instead of assigning priors on the process parameters ϕ , ν and the noise-to-spatial variance ratios δ^2 , we consider a set of candidate models $\mathcal{M} = \{M_1, \dots, M_G\}$ based on candidate values of these parameters. For each g , we sample exactly from the posterior distribution $p(\sigma^2, \beta, z \mid y, M_g)$ (see [svcLMexact\(\)](#)) and find the leave-one-out predictive densities $p(y_i \mid y_{-i}, M_g)$. The stacking weights solve

$$\begin{aligned} \max_{w_1, \dots, w_G} \quad & \frac{1}{n} \sum_{i=1}^n \log \sum_{g=1}^G w_g p(y_i \mid y_{-i}, M_g) \\ \text{subject to} \quad & w_g \geq 0, \sum_{g=1}^G w_g = 1. \end{aligned}$$

Candidate models that share the process parameters (ϕ, ν) are fitted together, building and factorizing the spatial correlation matrices only once, and the groups of such candidates are the units of parallel work.

Value

An object of class `svcLMstack`, which is a list including the following tags -

- `samples` a list of length equal to total number of candidate models with each entry corresponding to a list of length 4, containing posterior samples of fixed effects (beta), the noise variance (sigmaSq), the process variances (sigmaSq.z) and the spatially varying coefficients (z) for that model, as in [svcLMexact\(\)](#).
- `loopd` a list of length equal to total number of candidate models with each entry containing leave-one-out predictive densities under that particular model.
- `n.models` number of candidate models that are fit.
- `model.params` a list with one element per candidate model, each a named list of its parameters: `phi`, `nu` (NA for the exponential correlation function) and `noise_sp_ratio` (vectors of length `r` if `process.type = 'independent'`).
- `stacking.summary` a matrix with one row per candidate model, containing its parameters and its optimal stacking weight, for display.
- `stacking.weights` a numeric vector of length equal to the number of candidate models storing the optimal stacking weights.
- `run.time` a `proc_time` object with runtime details.
- `diagnostics` a list of diagnostics. Element `numerical` is a data frame with one row per candidate model (per candidate model and process, with columns `model` and `process`, if `process.type = 'independent'`) and columns `min.pivot`, `min.cor` and `max.cor`, as in [svcLMexact\(\)](#).

If `loopd.method = 'PSIS'`, element `pareto` contains the Pareto k diagnostic values of each candidate model (k), the threshold above which they are unreliable (`threshold`) and the number of values above it for each model (`n.high`). Element `solver` describes the optimization for the stacking weights: the solver used (`used`) and its status (`status`), the installed and requested solvers, the search order, the attempts with their status, and whether the fallback `loo::stacking_weights()` was used. If `verbose = TRUE`, a "Diagnostics" section is printed if there is an issue: numerical flags of candidate models with stacking weight above 0.05 (extreme candidates with negligible weight are expected in a stacking grid and are only counted), Pareto k values above the threshold for any model, and solver problems.

This object can be used to make predictions at new locations with `posteriorPredict()` and to sample from the stacked posterior with `stackedSampler()`.

Author(s)

Soumyakanti Pan span18@ucla.edu,
Sudipto Banerjee sudipto@ucla.edu

References

Vehtari A, Simpson D, Gelman A, Yao Y, Gabry J (2024). "Pareto Smoothed Importance Sampling." *Journal of Machine Learning Research*, **25**(72), 1-58. URL <https://jmlr.org/papers/v25/19-556.html>.
Zhang L, Tang W, Banerjee S (2025). "Bayesian Geostatistics Using Predictive Stacking." *Journal of the American Statistical Association*, **In press**. doi:10.1080/01621459.2025.2566449.

See Also

`svcLMexact()`, `svcGLMstack()`, `spLMstack()`, `stvcLMstack()`

Examples

```
set.seed(1234)
data(simSpatial)
dat <- simSpatial[1:200, ]

# independent processes (r = 2): vector-valued candidates via list()
mod.list <- candidateModels(list(phi = list(c(4, 4), c(8, 4), c(8, 8)),
  nu = list(c(1.5, 1.5)),
  noise_sp_ratio = list(c(0.1, 0.1), c(0.25, 0.25))),
  "cartesian")
mod1 <- svcLMstack(y_svc ~ x1 + x2 + (x1), data = dat,
  coords = as.matrix(dat[, c("s1", "s2")]),
  cor.fn = "matern", process.type = "independent",
  candidate.models = mod.list, n.samples = 500)
post_samps <- stackedSampler(mod1)
n <- nrow(dat)
cor(apply(post_samps$z[n + 1:n, ], 1, median), dat$z_true)
```

Index

- * **datasets**
 - simSpaceTime, [12](#)
 - simSpatial, [13](#)
- * **utilities**
 - iDist, [8](#)
- candidateModels, [4](#)
- candidateModels(), [20](#), [26](#), [35](#), [42](#), [51](#), [57](#)
- cholUpdate, [5](#)
- cholUpdateDel (cholUpdate), [5](#)
- cholUpdateDelBlock (cholUpdate), [5](#)
- cholUpdateRankOne (cholUpdate), [5](#)
- CVXR-package, [20](#), [27](#), [36](#), [42](#), [51](#), [58](#)
- CVXR::psolve(), [7](#), [8](#)
- future::plan(), [20](#), [26](#), [36](#), [42](#), [51](#), [57](#)
- get_stacking_weights, [7](#)
- iDist, [8](#)
- MBA::mba.surf(), [45](#), [46](#)
- posteriorPredict, [9](#)
- posteriorPredict(), [37](#), [43](#), [52](#), [59](#)
- recoverGLMscale, [11](#)
- recoverGLMscale(), [52](#)
- simSpaceTime, [12](#), [14](#)
- simSpatial, [13](#), [13](#)
- spGLMexact, [15](#)
- spGLMexact(), [3](#), [9–11](#), [22](#), [34](#), [50](#)
- spGLMstack, [19](#)
- spGLMstack(), [3](#), [8–11](#), [28–30](#), [52](#)
- spLMexact, [22](#)
- spLMexact(), [3](#), [9](#), [10](#), [18](#), [28](#), [40](#), [56](#)
- spLMstack, [25](#)
- spLMstack(), [3](#), [8–10](#), [22](#), [25](#), [29](#), [30](#), [44](#), [59](#)
- spStack (spStack-package), [2](#)
- spStack-package, [2](#)
- stackedSampler, [29](#)
- stackedSampler(), [43](#), [52](#), [59](#)
- stvcGLMexact, [30](#)
- stvcGLMexact(), [9–11](#), [40](#), [48–50](#)
- stvcGLMstack, [34](#)
- stvcGLMstack(), [4](#), [9–11](#), [29](#), [30](#), [44](#), [51](#), [52](#)
- stvcLMexact, [37](#)
- stvcLMexact(), [9](#), [10](#), [42–44](#), [55](#), [56](#)
- stvcLMstack, [41](#)
- stvcLMstack(), [9](#), [10](#), [29](#), [30](#), [40](#), [59](#)
- surfaceplot, [44](#)
- surfaceplot2, [46](#)
- svcGLMexact, [47](#)
- svcGLMexact(), [9–11](#), [51](#), [52](#), [56](#)
- svcGLMstack, [50](#)
- svcGLMstack(), [9–11](#), [29](#), [30](#), [50](#), [59](#)
- svclMexact, [53](#)
- svclMexact(), [9](#), [10](#), [50](#), [57–59](#)
- svclMstack, [56](#)
- svclMstack(), [9](#), [10](#), [29](#), [30](#), [52](#), [56](#)