Package 'sfclust'

May 9, 2026

Title Bayesian Spatial Functional Clustering

Version 1.0.1

Description Bayesian clustering of spatial regions with similar functional shapes using spanning trees and latent Gaussian models. The method enforces spatial contiguity within clusters and supports a wide range of latent Gaussian models, including non-Gaussian likelihoods, via the R-INLA framework. The algorithm is based on Zhong, R., Chacón-Montalván, E. A., and Moraga, P. (2024) <doi:10.48550/arXiv.2407.12633>, extending the approach of Zhang, B., Sang, H., Luo, Z. T., and Huang, H. (2023) <doi:10.1214/22-AOAS1643>. The package includes tools for model fitting, convergence diagnostics, visualization, and summarization of clustering results.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

Imports cubelyr, igraph, sf, SparseM, stars, dplyr, methods, Matrix

Suggests ggplot2, ggraph, class, fda, purrr, INLA, knitr, rmarkdown, testthat (>= 3.0.0)

Additional_repositories https://inla.r-inla-download.org/R/stable

Depends R (>= 4.1.0)

Config/testthat/edition 3

VignetteBuilder knitr

LazyData true

NeedsCompilation no

Author Erick A. Chacón-Montalván [aut, cre] (ORCID: <https://orcid.org/0000-0001-8068-1034>), Ruiman Zhong [aut] (ORCID: <https://orcid.org/0000-0002-4681-9199>), Paula Moraga [aut] (ORCID: <https://orcid.org/0000-0001-5266-0201>)

Maintainer Erick A. Chacón-Montalván <erick.chaconmontalvan@kaust.edu.sa>

Repository CRAN

Date/Publication 2025-05-19 13:20:02 UTC

Contents

`data_all` *Prepare data in long format*

Description

Convert spatio-temporal data to long format with indices for time and spatial location.

Usage

```
data_all(stdata, stnames = c("geometry", "time"))
```

Arguments

`stdata`	A stars object containing spatial-temporal dimensions defined in `stnames`.
`stnames`	The names of the `spatial` and `temporal` dimensions.

Value

A long-format data frame with ids for each observation and for spatial and time indexing.

Examples

```
library(sfclust)
library(stars)

dims <- st_dimensions(
  geometry = st_sfc(lapply(1:5, function(i) st_point(c(i, i)))),
  time = seq(as.Date("2024-01-01"), by = "1 day", length.out = 3)
)
stdata <- st_as_stars(cases = array(1:15, dim = c(5, 3)), dimensions = dims)

data_all(stdata)
```

data_each *Prepare data for a cluster*

Description

Subset a spatio-temporal dataset for a cluster and convert it to a long format with indices for time and spatial location.

Usage

```
data_each(k, membership, stdata, stnames = c("geometry", "time"))
```

Arguments

k	The cluster number to subset.
membership	A vector defining the cluster membership for each region.
stdata	A stars object containing spatial-temporal dimensions defined in stnames.
stnames	The names of the spatial and temporal dimensions.

Value

A long-format data frame with ids for each observation and for spatial and time indexing.

Examples

```
library(sfclust)
library(stars)

dims <- st_dimensions(
  geometry = st_sfc(lapply(1:5, function(i) st_point(c(i, i)))),
  time = seq(as.Date("2024-01-01"), by = "1 day", length.out = 3)
)
stdata <- st_as_stars(cases = array(1:15, dim = c(5, 3)), dimensions = dims)

data_each(k = 2, membership = c(1, 1, 1, 2, 2), stdata)
```

fitted.sfclust *Fitted Values for* sfclust *Objects*

Description

This function calculates the fitted values for a specific clustering sample in an sfclust object, based on the estimated models for each cluster. The fitted values are computed using the membership assignments and model parameters associated with the selected clustering sample.

Usage

```
## S3 method for class 'sfclust'
fitted(object, sample = object$clust$id, sort = FALSE, aggregate = FALSE, ...)
```

Arguments

object	An object of class 'sfclust', containing clustering results and models.
sample	An integer specifying the clustering sample number for which the fitted values should be computed. The default is the id of the current clustering. The value must be between 1 and the total number of clustering (membership) samples.
sort	Logical value indicating if clusters should be relabel based on number of elements.
aggregate	Logical value indicating if fitted values are desired at cluster level.
...	Additional arguments, currently not used.

Details

The function first checks if the provided sample value is valid (i.e., it is within the range of available clustering samples). If the specified sample does not match the current clustering id, the sfclust object is updated accordingly. It then retrieves the membership assignments and cluster models for the selected sample, calculates the linear predictions for each cluster, and combines them into a matrix of fitted values.

Value

A stars object with linear predictor fitted values at regions levels. In case aggregate = TRUE, the output produces an stars objecto at cluster levels.

Examples

```
library(sfclust)

data(stgaus)
result <- sfclust(stgaus, formula = y ~ f(idt, model = "rw1"), niter = 10,
  nmessage = 1)

# Estimated values ordering clusters by size
df_est <- fitted(result, sort = TRUE)

# Estimated values aggregated by cluster
df_est <- fitted(result, aggregate = TRUE)

# Estimated values using a particular clustering sample
df_est <- fitted(result, sample = 3)
```

genclust *Generate clusters for spatial clustering*

Description

Creates an undirected graph from spatial polygonal data, computes its minimum spanning tree (MST), and generates `nclust` clusters. This function is used to initialize cluster membership in a clustering algorithm, such as `sfclust`.

Usage

```
genclust(x, nclust = 10, weights = NULL)
```

Arguments

`x`	An `sf` or `sfc` object representing spatial polygonal data. It can also be a `matrix` or `Matrix` object with non-zero values representing weighted connectivity between units.
`nclust`	Integer, specifying the initial number of clusters.
`weights`	Optional `numeric` vector or `matrix` of weights between units in `x`. It should have dimensions `n^2`, where `n` is the number of units in `x`. If NULL, random weights are assigned.

Value

A list with three elements:

- `graph`: The undirected graph object representing spatial contiguity.
- `mst`: The minimum spanning tree.
- `membership`: The cluster membership for elements in `x`.

Examples

```
library(sfclust)
library(sf)

x <- st_make_grid(cellsize = c(1, 1), offset = c(0, 0), n = c(3, 2))

# using distance between geometries
clust <- genclust(x, nclust = 3, weights = st_distance(st_centroid(x)))
print(clust)
plot(st_sf(x, cluster = factor(clust$membership)))

# using increasing weights
cluster_ini <- genclust(x, nclust = 3, weights = 1:36)
print(cluster_ini)
plot(st_sf(x, cluster = factor(cluster_ini$membership)))
```

```
# using on random weights
cluster_ini <- genclust(x, nclust = 3, weights = runif(36))
print(cluster_ini)
plot(st_sf(x, cluster = factor(cluster_ini$membership)))
```

`log_mlik_all` *Fit models and compute the log marginal likelihood for all clusters*

Description

Fit the specified INLA model to each cluster and compute the log marginal likelihood for each cluster specified in the membership vector.

Usage

```
log_mlik_all(
  membership,
  stdata,
  stnames = c("geometry", "time"),
  correction = TRUE,
  detailed = FALSE,
  ...
)
```

Arguments

`membership`	Integer, character or factor vector indicating the cluster membership for each spatial unit.
`stdata`	A stars object with spatial-temporal dimensions defined in `stnames`, and including predictors and response variables.
`stnames`	The names of the `spatial` and `temporal` dimensions of the `stdata` object.
`correction`	Logical value indicating whether a correction for dispersion.
`detailed`	Logical value indicating whether to return the INLA model instead of the log marginal likelihood. The argument `correction` is not applied in this case.
`...`	Arguments passed to the `inla` function (eg. `family`, `formula` and `E`).

Value

A numeric vector containing the log marginal likelihood for each cluster or the the fitted INLA model for each cluster when `detailed = TRUE`.

Examples

```
library(sfclust)
library(stars)

dims <- st_dimensions(
  geometry = st_sfc(lapply(1:5, function(i) st_point(c(i, i)))),
  time = seq(as.Date("2024-01-01"), by = "1 day", length.out = 3)
)
stdata <- st_as_stars(
  cases = array(rpois(15, 100 * exp(1)), dim = c(5, 3)),
  temperature = array(runif(15, 15, 20), dim = c(5, 3)),
  expected = array(100, dim = c(5, 3)),
  dimensions = dims
)

log_mlik_all(c(1, 1, 1, 2, 2), stdata,
  formula = cases ~ temperature, family = "poisson", E = expected)

models = log_mlik_all(c(1, 1, 1, 2, 2), stdata, detailed = TRUE,
  formula = cases ~ temperature, family = "poisson", E = expected)
lapply(models, summary)
```

---

plot.sfclust *Plot function for* sfclust *objects*

Description

This function visualizes the estimated clusters from an `sfclust` object. It can display: (1) a map of regions colored by their assigned cluster, (2) the functional shapes of the linear predictors for each cluster, and (3) a traceplot of the log marginal likelihood. A conditional legend is added if the number of clusters is less than 10.

Usage

```
## S3 method for class 'sfclust'
plot(
  x,
  sample = x$clust$id,
  which = 1:3,
  clusters = NULL,
  sort = FALSE,
  legend = FALSE,
  ...
)
```

Arguments

x	An sfclust object containing the clustering results, including the cluster assignments and model parameters.
sample	Integer specifying the clustering sample number to summarize. Defaults to the last sample.
which	Integer vector indicating which plot to display. Options are: - 1: Map of regions colored by cluster assignment. - 2: Functional shapes of the linear predictors for each cluster. - 3: Traceplot of the log marginal likelihood.
clusters	Optional vector specifying which clusters to plot. If NULL, all clusters are included.
sort	Logical value indicating whether clusters should be relabeled based on the number of elements. Default is FALSE.
legend	Logical value indicating whether a legend should be included in the plot. Default is FALSE.
...	Additional arguments passed to the underlying plotting functions.

Value

A plot displaying the selected subgraphs as specified by which.

---

print.sfclust *Print method for sfclust objects*

Description

Prints details of an sfclust object, including the (i) within-cluster formula; (ii) hyperparameters used for the MCMC sample such as the number of clusters penalty (q) and the movement probabilities (move_prob); (iii) the number of movement type dones during the MCMC sampling; and (iv) the log marginal likelihood of the model of the last clustering sample.

Usage

```
## S3 method for class 'sfclust'
print(x, ...)
```

Arguments

x	An object of class 'sfclust'.
...	Additional arguments passed to print.default.

Value

Invisibly returns the input `sfclust` object `x`. The function also prints a summary of:

- the within-cluster model formula,
- clustering hyperparameters,
- movement counts from the MCMC sampler,
- and the log marginal likelihood of the selected sample.

`sfclust` *Bayesian spatial functional clustering*

Description

Bayesian detection of neighboring spatial regions with similar functional shapes using spanning trees and latent Gaussian models. It ensures spatial contiguity in the clusters, handles a large family of latent Gaussian models supported by `inla`, and allows to work with non-Gaussian likelihoods.

Usage

```
sfclust(
  stdata,
  graphdata = NULL,
  stnames = c("geometry", "time"),
  move_prob = c(0.425, 0.425, 0.1, 0.05),
  q = 0.5,
  correction = TRUE,
  niter = 100,
  burnin = 0,
  thin = 1,
  nmessage = 10,
  path_save = NULL,
  nsave = nmessage,
  ...
)
```

Arguments

`stdata`	A stars object containing response variables, covariates, and other necessary data.
`graphdata`	A list containing the initial graph used for the Bayesian model. It should include components like `graph`, `mst`, and `membership` (default is `NULL`).
`stnames`	A character vector specifying the spatio-temporal dimension names of `stdata` that represent spatial geometry and time, respectively (default is `c("geometry", "time")`).

move_prob	A numeric vector of probabilities for different types of moves in the MCMC process: birth, death, change, and hyperparameter moves (default is c(0.425, 0.425, 0.1, 0.05)).
q	A numeric value representing the penalty for the number of clusters (default is 0.5).
correction	A logical indicating whether correction to compute the marginal likelihoods should be applied (default is TRUE). This depend of the type of effect inclused in the INLA model.
niter	An integer specifying the number of MCMC iterations to perform (default is 100).
burnin	An integer specifying the number of burn-in iterations to discard (default is 0).
thin	An integer specifying the thinning interval for recording the results (default is 1).
nmessage	An integer specifying how often progress messages should be printed (default is 10).
path_save	A character string specifying the file path to save the results (default is NULL).
nsave	An integer specifying the number of iterations between saved results in the chain (default is nmessage).
...	Additional arguments such as formula, family, and others that are passed to the inla function.

Details

This implementation draws inspiration from the methods described in the paper: *"Bayesian Clustering of Spatial Functional Data with Application to a Human Mobility Study During COVID-19"* by Bohai Zhang, Huiyan Sang, Zhao Tang Luo, and Hui Huang, published in *The Annals of Applied Statistics*, 2023. For further details on the methodology, please refer to:

- The paper: doi:10.1214/22AOAS1643
- Supplementary material: doi:10.1214/22AOAS1643SUPPB

The MCMC algorithm in this implementation is largely based on the supplementary material provided in the paper. However, we have generalized the computation of the marginal likelihood ratio by leveraging INLA (Integrated Nested Laplace Approximation). This generalization enables integration over all parameters and hyperparameters, allowing for inference within a broader family of distribution functions and model terms, thereby extending the scope and flexibility of the original approach. Further details of our approach can be found in our paper *"Bayesian spatial functional data clustering: applications in disease surveillance"* by Ruiman Zhong, Erick A. Chacón-Montalván, Paula Moraga:

- The paper: https://arxiv.org/abs/2407.12633

Value

An sfclust object containing two main lists: samples and clust.

- The samples list includes details from the sampling process, such as:

  - membership: The cluster membership assignments for each sample.
  - log_marginal_likelihood: The log marginal likelihood for each sample.
  - move_counts: The counts of each type of move during the MCMC process.
- The clust list contains information about the selected clustering, including:
  - id: The identifier of the selected sample (default is the last sample).
  - membership: The cluster assignments for the selected sample.
  - models: The fitted models for each cluster in the selected sample.

Author(s)

Ruiman Zhong <ruiman.zhong@kaust.edu.sa>, Erick A. Chacón-Montalván <erick.chaconmontalvan@kaust.edu.sa>
Paula Moraga <paula.moraga@kaust.edu.sa>

Examples

```
library(sfclust)

# Clustering with Gaussian data
data(stgaus)
result <- sfclust(stgaus, formula = y ~ f(idt, model = "rw1"),
  niter = 10, nmessage = 1)
print(result)
summary(result)
plot(result)

# Clustering with binomial data
data(stbinom)
result <- sfclust(stbinom, formula = cases ~ poly(time, 2) + f(id),
  family = "binomial", Ntrials = population, niter = 10, nmessage = 1)
print(result)
summary(result)
plot(result)
```

---

stbinom *Spatio-temporal Binomial data*

Description

A simulated stars object containing binomial response data with a functional clustering pattern defined by polynomial fixed effects. This dataset includes the variables cases and population observed across 100 simulated spatial regions over 91 time points.

Usage

```
data(stbinom)
```

Format

A `stars` object with:

cases Number of observed cases (integer)

population Population at risk (integer)

dimensions Two dimensions: `geometry` (spatial features) and `time` (daily observations)

Examples

```
library(sfclust)

data(stbinom)
stbinom
plot(stbinom["cases"])
```

---

stgaus *Spatio-temporal Gaussian data*

Description

A simulated `stars` object containing Gaussian response data with a functional clustering pattern uging random walk processes. This dataset includes the response variable `y` observed across 100 simulated spatial regions over 91 time points.

Usage

```
data(stgaus)
```

Format

A `stars` object with:

y Response variable

Examples

```
library(sfclust)

data(stgaus)
stgaus
plot(stgaus["y"])
```

summary.sfclust *Summary method for sfclust objects*

Description

This function summarizes the cluster assignments from the desired clustering `sample`.

Usage

```
## S3 method for class 'sfclust'
summary(object, sample = object$clust$id, sort = FALSE, ...)
```

Arguments

`object`	An object of class 'sfclust'.
`sample`	An integer specifying the clustering sample number to be summarized (default is the last sample).
`sort`	Logical value indicating if clusters should be relabel based on number of elements.
`...`	Additional arguments passed to `print.default`.

Value

Invisibly returns a table with the number of regions in each cluster for the selected sample. The function also prints a summary that includes:

- the within-cluster model formula,
- the total number of MCMC clustering samples,
- the cluster membership counts for the specified sample (optionally sorted),
- and the log marginal likelihood of the selected clustering sample.

update.sfclust *Update MCMC Clustering Procedure*

Description

This function continues the MCMC sampling of a `sfclust` object based on previous results or update the model fitting for a specified sample clustering if the argument `sample` is provided.

Usage

```
## S3 method for class 'sfclust'
update(
  object,
  niter = 100,
  burnin = 0,
  thin = 1,
  nmessage = 10,
  sample = NULL,
  path_save = NULL,
  nsave = nmessage,
  ...
)
```

Arguments

object	A sfclust object.
niter	An integer specifying the number of additional MCMC iterations to perform.
burnin	An integer specifying the number of burn-in iterations to discard.
thin	An integer specifying the thinning interval for recording results.
nmessage	An integer specifying the number of messages to display during the process.
sample	An integer specifying the clustering sample number to be executed. The default is the last sample (i.e., nrow(x$samples$membership)).
path_save	A character string specifying the file path to save the results. If NULL, results are not saved.
nsave	An integer specifying how often to save results. Defaults to nmessage.
...	Additional arguments (currently not used).

Details

This function takes the last state of the Markov chain from a previous sfclust execution and uses it as the starting point for additional MCMC iterations. If sample is provided, it simply udpates the within-cluster models for the specified clustering sample.

Value

An updated sfclust object with (i) new clustering samples if sample is not specified, or (ii) updated within-cluster model results if sample is given.

Index