

Package ‘spconform’

September 12, 2026

Type Package

Title Conformal Prediction for Spatially and Spatio-Temporally
Dependent Data

Version 0.1.0

Description Provides distribution-free, model-agnostic prediction intervals for spatially and spatio-temporally dependent data using localized conformal calibration. Implements locally weighted split conformal prediction for geostatistical (point-referenced) data based on spatial-distance kernels, and a neighbourhood-weighted conformal procedure for areal (lattice) data based on graph adjacency structures. Relaxes the standard exchangeability assumption using spatial proximity, following the localized conformal framework of Mao, Martin and Reich (2024) <[doi:10.1080/01621459.2022.2147531](https://doi.org/10.1080/01621459.2022.2147531)>. Includes comprehensive spatial diagnostic tools to audit empirical coverage, conditional spatial strata, and boundary proximity effects.

License GPL (>= 3)

Encoding UTF-8

Depends R (>= 3.6.0)

Imports graphics, grDevices, stats

Suggests bmstdr, knitr, mgcv, ranger, rmarkdown, sp, testthat (>= 3.0.0)

VignetteBuilder knitr

URL <https://github.com/amjed-droid/spconform>,
<https://amjed-droid.github.io/spconform/>,
<https://doi.org/10.5281/zenodo.21862024>

BugReports <https://github.com/amjed-droid/spconform/issues>

Config/testthat/edition 3

Language en-US

Config/roxygen2/version 8.1.0

NeedsCompilation no

Author Ahmed Sattar Jabbar [aut, cre] (ORCID:
<<https://orcid.org/0009-0006-4614-735X>>)
Maintainer Ahmed Sattar Jabbar <ahmed.state.me@gmail.com>
Repository CRAN
Date/Publication 2026-09-12 14:40:02 UTC

Contents

areal_neighbor_weights	2
coverage_report	3
diagnose	4
plot.spconform	5
print.spconform	6
scp_areal	6
scp_geostatistical	7
spatial_kernel_weights	9
summary.spconform	11
Index	12

areal_neighbor_weights
<i>Compute neighbourhood-based weights for areal (lattice) data</i>

Description

Compute neighbourhood-based weights for areal (lattice) data

Usage

areal_neighbor_weights(i0, adjacency, decay = 1)

Arguments

- | | |
|-----------|---|
| i0 | Integer index of the target (unobserved / held-out) areal unit. |
| adjacency | A square 0/1 (or weighted) adjacency matrix describing the neighbourhood structure of the areal units (e.g. a spatial contiguity matrix). Row/column i0 corresponds to the target unit. |
| decay | Numeric decay rate applied to graph distance (number of hops) from i0; larger values down-weight distant neighbours more aggressively. Defaults to 1. |

Examples

```
adj <- matrix(c(0, 1, 0,
               1, 0, 1,
               0, 1, 0), nrow = 3, byrow = TRUE)
w <- areal_neighbor_weights(1, adj, decay = 0.5)
print(w)
```

coverage_report	<i>Empirical Coverage and Average Interval Width for an spconform Object</i>
-----------------	--

Description

Computes the empirical coverage rate and mean interval width by comparing conformal prediction intervals against held-out ground truth responses.

Usage

```
coverage_report(object, y_true)
```

Arguments

object	An object of class "spconform".
y_true	Numeric vector of true observed values at the prediction locations. Must have the same length as object\$pred.

Value

A named list with:

coverage Proportion of true values falling within the prediction intervals.

mean_width Mean width of the prediction intervals.

Examples

```
set.seed(1)
n <- 20
adj <- matrix(0, n, n)
for (i in 1:(n - 1)) { adj[i, i + 1] <- 1; adj[i + 1, i] <- 1 }
y <- cumsum(rnorm(n))
out <- scp_areal(y, adjacency = adj, alpha = 0.2)
coverage_report(out, y)
```

diagnose

*Comprehensive Diagnostic Report for spconform Objects***Description**

Produces a multi-panel diagnostic report assessing marginal coverage, conditional coverage by spatial strata, boundary effects, and the distribution of nonconformity scores.

Usage

```
diagnose(object, y_true, s_test = NULL, n_bins = 4, plot = TRUE, ...)
```

Arguments

<code>object</code>	An object of class "spconform", typically the output of <code>scp_geostatistical()</code> or <code>scp_areal()</code> .
<code>y_true</code>	Numeric vector of true response values at the prediction locations. Must have the same length as <code>object\$pred</code> .
<code>s_test</code>	Optional numeric matrix of prediction coordinates (one row per location). Required for spatial diagnostics.
<code>n_bins</code>	Integer; number of spatial bins for conditional coverage (default 4).
<code>plot</code>	Logical; if TRUE (default), generates a multi-panel diagnostic plot.
<code>...</code>	Additional arguments passed to plotting functions.

Details

The conditional coverage analysis partitions the prediction locations into `n_bins` equal-area spatial quadrants and reports coverage within each. The boundary analysis classifies points by their distance to the convex hull of the training data (for geostatistical output).

If empirical coverage exceeds the nominal target by more than 5 percentage points, a message is issued suggesting bandwidth reduction for tighter intervals.

Value

A list (invisibly) containing:

marginal Marginal coverage and mean width.

conditional Coverage and width by spatial bin.

boundary Coverage by distance from convex hull boundary.

scores Summary of nonconformity score distribution.

Examples

```
# Minimal reproducible example (< 0.1s execution time)
set.seed(123)
s_tr <- matrix(runif(40), ncol = 2)
y_tr <- rnorm(20)
s_te <- matrix(runif(20), ncol = 2)
y_te <- rnorm(10)

pfun <- function(s_train, y_train, s_new) rep(mean(y_train), nrow(s_new))

out <- scp_geostatistical(s_tr, y_tr, s_te, pfun, alpha = 0.1)
diag_res <- diagnose(out, y_te, s_te, plot = FALSE)
print(diag_res)
```

plot.spconform	<i>Plot Prediction Intervals for spconform Objects</i>
----------------	--

Description

Visualizes the conformal prediction intervals and point predictions across target locations or areal units. Optionally overlays true observations if provided.

Usage

```
## S3 method for class 'spconform'
plot(x, y_true = NULL, ...)
```

Arguments

<code>x</code>	An object of class "spconform".
<code>y_true</code>	Optional numeric vector of true observed responses matching <code>x\$pred</code> .
<code>...</code>	Additional arguments passed to plot .

Value

Invisibly returns the input object `x`.

<code>print.spconform</code>	<i>Print Method for spconform Objects</i>
------------------------------	---

Description

Prints a concise summary of an `spconform` prediction object, including the data type, target coverage level, total number of prediction points, and a preview of the first few prediction intervals.

Usage

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

Arguments

<code>x</code>	An object of class " <code>spconform</code> ".
<code>...</code>	Further arguments passed to or from other methods.

Value

Invisibly returns the input object `x`.

<code>scp_areal</code>	<i>Neighbourhood-Weighted Conformal Prediction for Areal (Lattice) Data</i>
------------------------	---

Description

Constructs distribution-free prediction intervals for areal units (e.g. counties, census tracts, grid cells) using a leave-one-unit-out conformal procedure in which nonconformity scores are weighted by graph (neighbourhood) distance to the held-out unit, via its adjacency structure.

Usage

```
scp_areal(y, X = NULL, adjacency, pred_fun = NULL, alpha = 0.1, decay = 1)
```

Arguments

<code>y</code>	Numeric vector of observed responses, one per areal unit.
<code>X</code>	Optional numeric design matrix of covariates (one row per areal unit); passed to <code>pred_fun</code> if supplied. May be <code>NULL</code> for purely spatial (neighbourhood-based) prediction.
<code>adjacency</code>	Square adjacency (contiguity) matrix describing the neighbourhood structure among the <code>length(y)</code> areal units. Non-zero entries are treated as neighbours (coerced to binary).

pred_fun	A function with signature <code>function(y_train, X_train, idx_train, idx_target, adjacency)</code> returning a single numeric point prediction for the areal unit indexed by <code>idx_target</code> , fitted using the training units <code>idx_train</code> . If <code>NULL</code> (default), a simple neighbourhood-mean predictor is used (the mean of <code>y_train</code> over the target's graph neighbours, falling back to the global training mean if the target has no observed neighbours).
alpha	Miscoverage level; intervals target $1 - \alpha$ coverage. Default 0.1.
decay	Decay rate for neighbourhood weights; see areal_neighbor_weights .

Details

Uses a full leave-one-out (jackknife-style) conformal scheme: for each areal unit i , a model is fit on all other units and used to predict unit i ; the resulting nonconformity scores across all units are combined into a graph-distance-weighted quantile specific to each target unit, so that the calibration set is dominated by spatially/graph-proximate units rather than treating all units as exchangeable.

If the effective neighbourhood weight for a target unit is vanishingly small (e.g., an isolated unit with no neighbours), the procedure falls back to an unweighted quantile over all other units and issues a warning.

Value

An object of class "spconform" (areal variant) with the same structure as [scp_geostatistical](#), using unit indices in place of coordinates.

Examples

```
set.seed(1)
n <- 25
adj <- matrix(0, n, n)
for (i in 1:(n - 1)) { adj[i, i + 1] <- 1; adj[i + 1, i] <- 1 }
y <- cumsum(rnorm(n)) + rnorm(n, sd = 0.2)

out <- scp_areal(y, adjacency = adj, alpha = 0.2)
print(out)
```

scp_geostatistical	<i>Locally Weighted Split Conformal Prediction for Geostatistical (Point-Referenced) Data</i>
--------------------	---

Description

Constructs distribution-free prediction intervals at new spatial (or spatio-temporal) locations by combining split conformal prediction with spatial-distance kernel weights, relaxing the exchangeability assumption that standard conformal prediction relies on.

Usage

```

scp_geostatistical(
  s_train,
  y_train,
  s0,
  pred_fun,
  alpha = 0.1,
  split = 0.5,
  bandwidth = NULL,
  t_train = NULL,
  t0 = NULL,
  temporal_bandwidth = NULL,
  seed = NULL
)

```

Arguments

<code>s_train</code>	Numeric matrix of training coordinates, one row per observation (e.g. longitude/latitude, or projected easting/northing).
<code>y_train</code>	Numeric vector of training responses, same length as <code>nrow(s_train)</code> .
<code>s0</code>	Numeric matrix of coordinates at which prediction intervals are desired, one row per target location.
<code>pred_fun</code>	A function with signature <code>function(s_train, y_train, s_new)</code> that returns a numeric vector of point predictions for the locations in <code>s_new</code> , fitted using <code>s_train/y_train</code> . This lets the user plug in any spatial predictor (kriging, random forest, GAM, etc.); <code>spconform</code> handles only the conformal calibration layer.
<code>alpha</code>	Miscoverage level; prediction intervals target $1 - \alpha$ coverage. Default 0.1 (90 percent intervals).
<code>split</code>	Proportion of the training data used for model fitting; the remainder is used as the calibration set for conformal scores. Default 0.5.
<code>bandwidth</code>	Optional spatial kernel bandwidth passed to spatial_kernel_weights ; if <code>NULL</code> , chosen automatically per target location.
<code>t_train, t0, temporal_bandwidth</code>	Optional time indices (and temporal bandwidth) for spatio-temporal weighting; see spatial_kernel_weights .
<code>seed</code>	Optional integer seed for the train/calibration split, for reproducibility.

Details

This implements a localized split-conformal procedure in the spirit of Mao, Martin and Reich (2020, "Valid model-free spatial prediction") and subsequent spatio-temporal extensions: nonconformity scores from a held-out calibration set are combined into a weighted empirical quantile where weights decay with distance (in space, and optionally time) from the target location, so nearby, more relevant observations dominate the calibration of the interval while validity is retained under a local exchangeability argument.

Value

An object of class "spconform", a list with components:

pred Point predictions at `s0`.

lower, upper Lower/upper prediction interval bounds at `s0`.

alpha The requested miscoverage level.

s0 The target coordinates.

call The matched call.

References

Mao, H., Martin, R., and Reich, B. J. (2020). Valid model-free spatial prediction. *arXiv:2006.15640*.

Examples

```
set.seed(42)
n <- 200
s <- matrix(runif(2 * n), ncol = 2)
f <- function(s) sin(4 * s[, 1]) + cos(3 * s[, 2])
y <- f(s) + rnorm(n, sd = 0.3)

idx <- sample(n, 150)
s_train <- s[idx, ]; y_train <- y[idx]
s0 <- s[-idx, ][1:10, ]

pred_fun <- function(s_train, y_train, s_new) {
  fit <- lm(y_train ~ s_train[, 1] + s_train[, 2] +
            I(s_train[, 1]^2) + I(s_train[, 2]^2))
  nd <- data.frame(s_train = I(s_new))
  as.numeric(cbind(1, s_new[, 1], s_new[, 2], s_new[, 1]^2, s_new[, 2]^2) %*%
             coef(fit))
}

out <- scp_geostatistical(s_train, y_train, s0, pred_fun, alpha = 0.1, seed = 1)
print(out)
```

spatial_kernel_weights

*Compute Gaussian-kernel spatial weights between a target location
and a set of reference locations*

Description

Compute Gaussian-kernel spatial weights between a target location and a set of reference locations

Usage

```
spatial_kernel_weights(
  s0,
  s,
  bandwidth = NULL,
  t0 = NULL,
  t = NULL,
  temporal_bandwidth = NULL
)
```

Arguments

<code>s0</code>	A numeric vector (or 1-row matrix) giving the coordinates of the target (prediction) location.
<code>s</code>	A numeric matrix of coordinates (one row per observation) for the reference (calibration) set.
<code>bandwidth</code>	Positive numeric bandwidth of the Gaussian kernel. If <code>NULL</code> (default), the bandwidth is chosen automatically as the median pairwise distance among <code>s</code> (Silverman-type rule).
<code>t0</code>	Optional numeric scalar: time index of the target observation, for spatio-temporal weighting. If supplied, <code>t</code> must be supplied too.
<code>t</code>	Optional numeric vector: time indices for the reference set, same length as <code>nrow(s)</code> .
<code>temporal_bandwidth</code>	Positive numeric bandwidth for the temporal kernel; only used if <code>t0/t</code> are supplied. Defaults to the spatial bandwidth logic applied to <code>t</code> .

Details

Weights are computed as

$$w_i = \exp(-\|s_0 - s_i\|^2 / (2h^2))$$

for the spatial-only case, and multiplied by an analogous temporal kernel term when `t0/t` are provided. This is the core device used to relax the exchangeability assumption required by standard conformal prediction: observations located near the point to be predicted contribute more to the calibration of the prediction interval than distant ones (Mao, Martin and Reich, 2020).

Value

A numeric vector of weights (not necessarily summing to one), one per row of `s`, giving higher weight to reference points that are spatially (and, if requested, temporally) closer to `s0`.

Examples

```
set.seed(1)
s <- matrix(runif(20), ncol = 2)
s0 <- c(0.5, 0.5)
w <- spatial_kernel_weights(s0, s)
```

```
round(w, 3)
```

summary.spconform	<i>Summary Method for spconform Objects</i>
-------------------	---

Description

Computes and displays summary statistics of the prediction interval widths generated by an spconform object.

Usage

```
## S3 method for class 'spconform'  
summary(object, ...)
```

Arguments

object	An object of class "spconform".
...	Further arguments passed to or from other methods.

Value

Invisibly returns the input object object.

Index

areal_neighbor_weights, [2](#), [7](#)

coverage_report, [3](#)

diagnose, [4](#)

plot, [5](#)

plot.spconform, [5](#)

print.spconform, [6](#)

scp_areal, [6](#)

scp_geostatistical, [7](#), [7](#)

spatial_kernel_weights, [8](#), [9](#)

summary.spconform, [11](#)