Package 'spconform'

September 24, 2026

Type Package

Title Conformal Prediction for Spatially and Spatio-Temporally Dependent Data

Version 0.1.1

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>. 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 stats

Suggests testthat (>= 3.0.0), knitr, rmarkdown, sp

VignetteBuilder knitr

Config/testthat/edition 3

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 2026-09-24 12:20:02 UTC

Contents

`areal_neighbor_weights` *Areal (Lattice) Graph Distance Proximity Weights*

Description

Computes neighbourhood-based weights for areal (lattice) data based on shortest-path graph distances.

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.

Value

A numeric vector of length `nrow(adjacency)` with weights based on graph distance from `i0` (self-weight is 0, i.e. the target unit is excluded from its own calibration set).

as.data.frame.spconform *Coerce an spconform Object to a Data Frame*

Description

Converts a conformal prediction output object of class "spconform" into a tidy data.frame containing coordinates (if present), point predictions, lower/upper bounds, and interval widths.

Usage

```
## S3 method for class 'spconform'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

x	An object of class "spconform".
row.names	NULL or a character vector giving row names.
optional	Logical; passed to as.data.frame.
...	Additional arguments (currently unused).

Value

A data.frame with prediction columns pred, lower, upper, width, and any coordinate columns.

coverage_report *Empirical Coverage and Average Interval Width for an spconform Object*

Description

Empirical Coverage and Average Interval Width for an spconform Object

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, same length/order as object$pred.

Value

A named list with `coverage` (proportion of `y_true` falling within `[lower, upper]`) and `mean_width` (average interval width).

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 Suite for Spatial Conformal Prediction Objects*

Description

Produces a comprehensive multi-panel diagnostic report evaluating marginal coverage validity, Winkler Interval Score (WIS) sharpness, conditional coverage across spatial strata, boundary proximity effects, spatial residual autocorrelation (Moran's I), and nonconformity score distributions.

Usage

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

Arguments

`object`	An object of class `"spconform"`, typically output from `scp_geostatistical` or `scp_areal`.
`y_true`	Numeric vector of true observed responses at target prediction locations. Must have the same length as `object$pred`.
`s_test`	Optional numeric matrix of target prediction coordinates ($m \times p$). Required for spatial stratification, boundary effects, and spatial autocorrelation diagnostics.
`n_bins`	Integer; number of spatial bins per dimension for conditional coverage (default 4).
`plot`	Logical; if `TRUE` (default), generates a multi-panel diagnostic plot.
`...`	Additional graphical parameters passed to internal plotting methods.

Details

The diagnostic suite performs five rigorous audits on the prediction intervals:

1. **Marginal Validity & Sharpness**: Evaluates empirical coverage $\frac{1}{m}\sum \mathbf{1}(Y_i \in C(s_i))$, mean/median interval width, and the strictly proper **Winkler Interval Score (WIS)**:

$$\text{WIS}_\alpha(l, u, y) = (u - l) + \frac{2}{\alpha}(l - y)\mathbf{1}(y < l) + \frac{2}{\alpha}(y - u)\mathbf{1}(y > u)$$

2. **Conditional Spatial Stratification**: Partitions the 2D spatial domain into n_bins $\times$ n_bins equal-area quadrants and assesses local coverage.
3. **Convex Hull Boundary Proximity**: Assesses whether edge-effect extrapolations degrade coverage near the boundary of the spatial domain.
4. **Spatial Autocorrelation (Moran's I)**: Evaluates whether prediction miscoverages or nonconformity scores exhibit residual spatial clustering via Moran's I.
5. **Score Distribution**: Analyzes empirical quantiles and normality/QQ structure.

Value

An S3 object of class `"spconform_diagnose"` containing:

marginal List with coverage, mean width, median width, sd width, Winkler score, and sample size.

conditional Data frame of coverage and widths partitioned by spatial quadrant.

boundary List comparing empirical metrics between interior and boundary units.

spatial_autocorr List containing Moran's I statistic, expected value, z-score, and p-value.

scores Summary of nonconformity score moments and quantiles.

alpha Miscoverage level of the evaluated object.

References

Winkler, R. L. (1972). "A Decision-Theoretic Approach to Interval Estimation." *Journal of the American Statistical Association*, 67(337), 187-191.

Mao, R., Martin, R., and Reich, B. J. (2023). "Valid Conformal Prediction for Dependent Data." *Journal of the American Statistical Association*, doi:10.1080/01621459.2022.2147531.

See Also

`scp_geostatistical`, `scp_areal`, `coverage_report`

Examples

```
set.seed(42)
n <- 100
s_tr <- matrix(runif(n * 2), n, 2)
y_tr <- sin(3 * s_tr[, 1]) + rnorm(n, sd = 0.2)
s_te <- matrix(runif(30 * 2), 30, 2)
y_te <- sin(3 * s_te[, 1]) + rnorm(30, sd = 0.2)
```

```
pfun <- function(s_tr, y_tr, s_new) rep(mean(y_tr), nrow(s_new))
out <- scp_geostatistical(s_tr, y_tr, s_te, pfun, alpha = 0.1)

diag_res <- diagnose(out, y_true = y_te, s_test = s_te, plot = FALSE)
print(diag_res)
```

`plot.spconform` *Plot Prediction Intervals for spconform Objects*

Description

Plots point predictions and conformal prediction intervals across observation indices, optionally overlaying true target values for visual evaluation.

Usage

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

Arguments

`x`	An object of class `"spconform"`.
`y_true`	Optional numeric vector of true observed responses at prediction locations.
`...`	Further graphical parameters passed to `plot`.

Value

Invisibly returns the input object `x`.

`plot.spconform_diagnose`
Plot Diagnostic Audit for an spconform_diagnose Object

Description

Generates a multi-panel visual audit of marginal coverage, nonconformity score QQ-distribution, conditional spatial strata coverage, and boundary proximity effects.

Usage

```
## S3 method for class 'spconform_diagnose'
plot(x, ...)
```

Arguments

x	An object of class "spconform_diagnose".
...	Additional arguments passed to plotting functions.

Value

Invisibly returns the input object x.

---

predict.spconform *Extract Predictions and Intervals from an spconform Object*

Description

Extracts point predictions or prediction interval bounds from a fitted spconform object.

Usage

```
## S3 method for class 'spconform'
predict(object, interval = c("none", "prediction"), ...)
```

Arguments

object	An object of class "spconform".
interval	Character string specifying the type of prediction intervals to extract: "none" (default, point predictions only) or "prediction" (matrix with fit, lwr, upr).
...	Further arguments passed to or from other methods.

Value

If interval = "none", a numeric vector of point predictions. If interval = "prediction", a numeric matrix with columns fit, lwr, and upr.

---

print.spconform *Print Method for spconform Objects*

Description

Displays a concise summary of the conformal prediction object including model type, nominal target coverage, and a preview table of prediction intervals.

Usage

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

Arguments

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

Value

Invisibly returns the input object x.

---

residuals.spconform *Extract Residuals from an spconform Object*

Description

Calculates prediction residuals (observed minus fitted values) or nonconformity scores for an spconform object given true responses.

Usage

```
## S3 method for class 'spconform'
residuals(object, y_true, type = c("response", "abs"), ...)
```

Arguments

object	An object of class "spconform".
y_true	Numeric vector of true observed responses at prediction locations.
type	Character string indicating residual type: "response" (raw residuals $y - \hat{y}$) or "abs" (absolute nonconformity scores $\|y - \hat{y}\|$).
...	Further arguments passed to or from other methods.

Value

A numeric vector of residuals.

scp_areal *Neighbourhood-Weighted Conformal Prediction for Areal (Lattice) Data*

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

y	Numeric vector of observed responses, one per areal unit.
X	Optional numeric design matrix of covariates (one row per areal unit); passed to pred_fun if supplied. May be NULL for purely spatial (neighbourhood-based) prediction.
adjacency	Square adjacency (contiguity) matrix describing the neighbourhood structure among the length(y) areal units. Non-zero entries are treated as neighbours (coerced to binary).
pred_fun	A function with signature function(y_train, X_train,idx_train, idx_target, adjacency) returning a single numeric point prediction for the areal unit indexed by idx_target, fitted using the training units idx_train. If NULL (default), a simple neighbourhood-mean predictor is used (the mean of y_train 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 *Locally Weighted Split Conformal Prediction for Geostatistical (Point-Referenced) Data*

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

s_train	Numeric matrix of training coordinates, one row per observation (e.g. longitude/latitude, or projected easting/northing).
y_train	Numeric vector of training responses, same length as nrow(s_train).
s0	Numeric matrix of coordinates at which prediction intervals are desired, one row per target location.

`pred_fun` A function with signature `function(s_train, y_train,s_new)` that returns a numeric vector of point predictions for the locations in `s_new`, fitted using `s_train`/`y_train`. This lets the user plug in any spatial predictor (kriging, random forest, GAM, etc.); `spconform` handles only the conformal calibration layer.

`alpha` Miscoverage level; prediction intervals target $1 - \alpha$ coverage. Default 0.1 (90 percent intervals).

`split` Proportion of the training data used for model fitting; the remainder is used as the calibration set for conformal scores. Default 0.5.

`bandwidth` Optional spatial kernel bandwidth passed to `spatial_kernel_weights`; if `NULL`, chosen automatically per target location.

`t_train`, `t0`, `temporal_bandwidth` Optional time indices (and temporal bandwidth) for spatio-temporal weighting; see `spatial_kernel_weights`.

`seed` 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`

Spatial Gaussian Kernel Proximity Weights

Description

Computes 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

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

temporal_bandwidth
: Positive numeric bandwidth for the temporal kernel; only used if t0/t are supplied. Defaults to the spatial bandwidth logic applied to t.

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 *Summary Method for spconform Objects*

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