Package ‘psfofr’

August 6, 2025

Type Package

Title Penalized Spatial Function-on-Function Regression Model

Version 1.0

Depends R (>= 3.5.0), fda, MASS, Matrix, matrixStats, methods

LazyLoad yes

ByteCompile TRUE

Encoding UTF-8

Maintainer Ufuk Beyaztas <ufukbeyaztas@gmail.com>

Description

Functions for implementing methods for penalized spatial function-on-function regression model.

Contents

predict_sffr2SLS 1
sffr_pen2SLS 3
sff_dgp 5

Index **8**

`predict_sffr2SLS` *Out–of–sample prediction for Penalised Spatial FoFR models*

Description

Given a fitted model returned by `sffr_pen2SLS`, this function produces predicted functional responses at new spatial units whose functional covariates and spatial weight matrix are supplied by the user. A fixed–point solver enforces the spatial autoregressive feedback implicit in the SFoFR model.

Usage

```
predict_sffr2SLS(object, xnew, Wnew)
```

Arguments

- `object` An object of class "sffr2SLS": the list produced by sffr_pen2SLS. At minimum it must contain gpx, gpy, K0, Ky, Kx, and the B–spline coefficient matrices b0_mat, b_mat, r_mat.
- `xnew` Numeric matrix of dimension $n_{\text{new}} \times |\text{gpx}|$, holding the functional covariate for the *new* spatial units, evaluated on the same predictor grid used during model fitting.
- `Wnew` Row–normalised $n_{\text{new}} \times n_{\text{new}}$ spatial weight matrix that captures proximity among the *new* units. Its definition should mirror that of the training matrix (e.g.\ inverse distance, k–nearest neighbours, etc.).

Details

Let $\widehat{\beta}_0(t)$, $\widehat{\beta}(t, s)$, and $\widehat{\rho}(t, u)$ be the estimated surfaces stored in object. For each new unit i the algorithm first forms the non–spatial regression prediction

$$\widehat{G}_i(t) \;=\; \widehat{\beta}_0(t) + \int_0^1 X_i(s)\, \widehat{\beta}(t, s)\, ds,$$

computed efficiently by pre-evaluated B–spline bases. Spatial feedback is then introduced by iterating

$$Y_i^{(\ell+1)}(t) \;=\; \widehat{G}_i(t) + \sum_{j=1}^{n_{\text{new}}} w_{ij} \int_0^1 Y_j^{(\ell)}(u)\, \widehat{\rho}(t, u)\, du,$$

until the sup–norm difference between successive curves falls below 1e-3 or 1,000 iterations are reached. Convergence is guaranteed when $\|\widehat{\rho}\|_\infty < 1/\|Wnew\|_\infty$, a condition typically satisfied by the fitted model if the training weight matrix met it during estimation.

Value

A numeric matrix of dimension $n_{\text{new}} \times |\text{gpy}|$ containing the predicted functional responses evaluated on gpy. Row i corresponds to the i-th row of xnew.

Note

If the new weight matrix induces very strong dependence, the fixed–point iterations may converge slowly. Consider scaling Wnew to have $\|Wnew\|_\infty \leq 1$ or relaxing the tolerance.

Author(s)

Ufuk Beyaztas, Han Lin Shang, and Gizel Bakicierler Sezer

References

Beyaztas, U., Shang, H. L., and Sezer, G. B. (2025). *Penalised Spatial Function–on–Function Regression*. Journal of Agricultural, Biological, and Environmental Statistics (in-press).

See Also

sff_dgp for simulated data generation; sffr_pen2SLS for model fitting.

Examples

```
## Not run: -----------------------------------------------------------
## 1. Fit a model on small simulated data
# train <- sff_dgp(n = 500, rf = 0.5)
# lam   <- list(lb = c(10^{-3}, 10^{-2}, 10^{-1}),
#                       lrho = c(10^{-3}, 10^{-2}, 10^{-1}))

# fit <- sffr_pen2SLS(train$Y, train$X, train$W,
#                     gpy = seq(0, 1, length = 101),
#                     gpx = seq(0, 1, length = 101),
#                     K0 = 10, Ky = 10, Kx = 10,
#                     lam_cands = lam)

## 2. Simulate NEW covariates and a compatible weight matrix
# test <- sff_dgp(n = 1000, rf = 0.5)  ## we keep only X and W
# pred <- predict_sffr2SLS(fit, xnew = test$X, Wnew = test$W)
## End(Not run)
```

`sffr_pen2SLS` *Penalised Spatial Two–Stage Least Squares for SFoFR*

Description

Fits the penalised spatial function-on-function regression (SFoFR) model via the two–stage least–squares (Pen2SLS) estimator introduced by Beyaztas, Shang and Sezer (2025). It selects optimal smoothing parameters, estimates regression and spatial autocorrelation surfaces, and (optionally) builds percentile bootstrap confidence bands.

Usage

```
sffr_pen2SLS(
  y, x, W, gpy, gpx,
  K0, Ky, Kx,
  lam_cands,
  boot       = FALSE,
  nboot      = NULL,
  percentile = NULL
)
```

Arguments

`y`	`n` x `length(gpy)` matrix of functional responses evaluated on grid `gpy`.
`x`	`n` x `length(gpx)` matrix of functional predictors evaluated on grid `gpx`.
`W`	`n` x `n` row–normalised spatial weight matrix, typically inverse–distance.
`gpy`	Numeric vector of response evaluation points $t \in [0, 1]$.
`gpx`	Numeric vector of predictor evaluation points $s \in [0, 1]$.
`K0`	Integer; number of basis functions for the intercept $\beta_0(t)$.
`Ky`	Integer; number of basis functions in the *response* direction of the bivariate surfaces $\rho(t, u)$ and $\beta(t, s)$.

`Kx`	Integer; number of basis functions in the *predictor* direction of the regression surface $\beta(t, s)$.
`lam_cands`	Two–column matrix or data frame whose rows contain candidate smoothing pairs $(\lambda_\rho,\ \lambda_\beta)$ to be ranked by BIC.
`boot`	Logical; if `TRUE` percentile bootstrap confidence intervals are produced.
`nboot`	Number of bootstrap resamples. Required when `boot = TRUE`.
`percentile`	Desired CI nominal width in percent (e.g., `95`). Required when `boot = TRUE`.

Details

The estimator minimises the penalised objective

$$\|Z^*\{\mathrm{vec}(Y) - \Pi\theta\}\|^2 \ + \ \frac{1}{2}\lambda_\rho P(\rho) \ + \ \frac{1}{2}\lambda_\beta P(\beta),$$

where $\theta = (\mathrm{vec}\,\rho, \mathrm{vec}\,\beta)$ are tensor-product B-spline coefficients, Z^* is the projection onto instrumental variables, and $P(\cdot)$ are Kronecker-sum quadratic roughness penalties in both surface directions. Candidate smoothing pairs are scored by the Bayesian Information Criterion

$$\mathrm{BIC} = -2\log\mathcal{L} + \omega\log n,$$

with log-likelihood based on squared residuals and ω equal to the effective degrees of freedom.

If `boot = TRUE`, residuals are centred, resampled, and the entire estimation procedure is repeated `nboot` times. Lower and upper percentile bounds are then extracted for $\beta(t, s)$, $\rho(t, u)$, and $\widehat{Y}_i(t)$.

Value

A named `list`:

b0hat Estimated intercept curve $\widehat{\beta}_0(t)$.

bhat Matrix of $\widehat{\beta}(t, s)$ values.

rhohat Matrix of $\widehat{\rho}(t, u)$ values.

b0_mat, b_mat, r_mat Raw coefficient matrices of B-spline basis weights for β_0, β, and ρ.

fitted.values $\widehat{Y}_i(t)$ matrix.

residuals $\widehat{\varepsilon}_i(t)$ matrix.

CI_bhat Two-element list with lower/upper percentile surfaces (NULL unless `boot = TRUE`).

CI_rhohat Analogous list for ρ.

CIy Percentile bands for the fitted responses.

gpy, gpx, K0, Ky, Kx Returned for convenience.

Author(s)

Ufuk Beyaztas, Han Lin Shang, and Gizel Bakicierler Sezer

References

Beyaztas, U., Shang, H. L., and Sezer, G. B. (2025). *Penalised Spatial Function–on–Function Regression*. Journal of Agricultural, Biological, and Environmental Statistics (in-press).

Examples

```
## Not run: ------------------------------------------------------------
## 1. simulate data
# sim <- sff_dgp(n = 100, rf = 0.7)
## 2. candidate smoothing grid (four pairs)
# lam <- list(lb = c(10^{-3}, 10^{-2}, 10^{-1}),
#                        lrho = c(10^{-3}, 10^{-2}, 10^{-1}))
## 3. fit model without bootstrap
# fit <- sffr_pen2SLS(
#   y         = sim$Y,
#   x         = sim$X,
#   W         = sim$W,
#   gpy       = seq(0, 1, length.out = 101),
#   gpx       = seq(0, 1, length.out = 101),
#   K0        = 10,
#   Ky        = 10,
#   Kx        = 10,
#  lam_cands  = lam,
#  boot       = FALSE
# )
## End(Not run)
```

---

sff_dgp *Simulate data from a Spatial Function–on–Function Regression model*

Description

Generates synthetic functional predictors and responses from the spatial function-on-function regression (SFoFR) data-generating process described in Beyaztas, Shang and Sezer (2025). The model embeds spatial autoregression on the functional response, Fourier–type basis structure for the covariate, and user-controlled Gaussian noise.

Usage

```
sff_dgp(
  n,
  nphi     = 10,
  gpy      = NULL,
  gpx      = NULL,
  rf       = 0.9,
  sd.error = 0.01,
  tol      = 0.001,
  max_iter = 1000
)
```

Arguments

n	Number of spatial units (curves) to generate.
nphi	Number of sine *and* cosine basis functions used to build each functional predictor. Total latent scores generated are therefore `2 * nphi`.
gpy	Numeric vector of evaluation points for the response domain $t \in [0, 1]$. Defaults to an equally–spaced grid of 101 points.

gpx Numeric vector of evaluation points for the predictor domain $s \in [0, 1]$. Defaults to an equally–spaced grid of 101 points.

rf Scalar in $(0, 1)$ controlling the strength of spatial autocorrelation through the surface $\rho(t, u)$. Values closer to 1 yield stronger dependence.

sd.error Standard deviation of the i.i.d. Gaussian noise $\varepsilon_i(t)$ added to the latent regression part.

tol Absolute tolerance used in the fixed–point iteration that solves the spatial autoregressive operator equation (stopping rule on the sup–norm of successive iterates).

max_iter Maximum number of fixed–point iterations. Prevents infinite looping when strong spatial feedback and small tol interact.

Details

The generator mimics the penalised SFoFR set-up:

$$Y_i(t) \;=\; \sum_{j=1}^{n} w_{ij} \int_0^1 Y_j(u)\,\rho(t,u)\,du \;+\; \int_0^1 X_i(s)\,\beta(t,s)\,ds \;+\; \varepsilon_i(t),$$

where

- w_{ij} are row–normalised inverse–distance weights,
- $X_i(s)$ is built from Fourier scores $\xi_{ijk} \sim \mathcal{N}(0, 1)$ and damped basis functions $\phi_k^{\cos}(s) = (k^{-3/2})\sqrt{2}\cos(k\pi s)$ and $\phi_k^{\sin}(s) = (k^{-3/2})\sqrt{2}\sin(k\pi s)$,
- the regression surface is $\beta(t, s) = 2 + s + t + 0.5\sin(2\pi st)$,
- the spatial autocorrelation surface is $\rho(t, u) = rf\,(1 + ut)/(1 + |u - t|)$,
- $\varepsilon_i(t) \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2)$, with $\sigma =$ sd.error.

Given the contraction condition $\|\rho\|_\infty < 1/\|W\|_\infty$, the Neumann series defining $(\mathbb{I} - \mathcal{T})^{-1}$ converges and the solution is obtained by simple fixed-point iterations until the change is below tol. Full details are in Beyaztas, Shang and Sezer (2025).

Value

A named list with components

Y n x length(gpy) matrix of observed functional responses on the grid gpy.

Y_true Same dimension as Y; noise–free latent responses before adding $\varepsilon_i(t)$.

X n x length(gpx) matrix of functional predictors.

W n x n row-normalised spatial weight matrix based on inverse distances.

rho length(gpy) x length(gpy) matrix containing $\rho(t, u)$ evaluated on the response grid.

beta length(gpx) x length(gpy) matrix containing $\beta(t, s)$ evaluated on the Cartesian product of the predictor and response grids.

Author(s)

Ufuk Beyaztas, Han Lin Shang, and Gizel Bakicierler Sezer

References

Beyaztas, U., Shang, H. L., and Sezer, G. B. (2025). *Penalised Spatial Function–on–Function Regression*. Journal of Agricultural, Biological, and Environmental Statistics (in-press).

Examples

```
## Not run: ----------------------------------------------------------------
## generate a toy data set
# dat <- sff_dgp(n = 250, rf = 0.5)
## End(Not run)
```

Index

predict_sffr2SLS, 1

sff_dgp, *2*, 5
sffr_pen2SLS, *1*, *2*, 3