Package: VDPO (via r-universe)

October 22, 2024

Title Working with and Analyzing Functional Data of Varying Lengths

Version 0.1.0

Description Comprehensive set of tools for analyzing and manipulating
functional data with non-uniform lengths. This package
addresses two common scenarios in functional data analysis:
Variable Domain Data, where the observation domain differs
across samples, and Partially Observed Data, where observations
are incomplete over the domain of interest. 'VDPO' enhances the
flexibility and applicability of functional data analysis in
'R'. See Amaro et al. (2024) <doi:10.48550/arXiv.2401.05839>.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 3.5.0)

Imports utils, Matrix, SOP, splines, stats

Suggests ggplot2, knitr, RColorBrewer, rmarkdown, testthat (>= 3.0.0)

RoxygenNote 7.3.2

Config/testthat/edition 3

VignetteBuilder knitr

URL https://pavel-hernadez-amaro.github.io/VDPO/

NeedsCompilation no

Author Pavel Hernandez [aut, cre], Jose Ignacio Diez [ctr], Maria
Durban [ctb], Maria del Carmen Aguilera-Morillo [ctb]

Maintainer Pavel Hernandez <pavel.amaro96@gmail.com>

Repository CRAN

Date/Publication 2024-10-21 11:30:26 UTC

Contents

---

data_generator_vd   *Data generator function for the variable domain case*

Description

Generates a variable domain functional regression model

Usage

```
data_generator_vd(
  N = 100,
  J = 100,
  nsims = 1,
  Rsq = 0.95,
  aligned = TRUE,
  multivariate = FALSE,
  beta_index = 1,
  use_x = FALSE,
  use_f = FALSE
)
```

Arguments

N	Number of subjects.
J	Number of maximum observations per subject.
nsims	Number of simulations per the simulation study.
Rsq	Variance of the model.
aligned	If the data that will be generated is aligned or not.
multivariate	If TRUE, the data is generated with 2 functional variables.
beta_index	Index for the beta.
use_x	If the data is generated with x.
use_f	If the data is generated with f.

Value

A list containing the following components:

- y: vector of length N containing the response variable.
- X_s: matrix of non-noisy functional data for the first functional covariate.
- X_se: matrix of noisy functional data for the first functional covariate
- Y_s: matrix of non-noisy functional data for the second functional covariate (if multivariate).
- Y_se: matrix of noisy functional data for the second covariate (if multivariate).

- x1: vector of length N containing the non-functional covariate (if use_x is TRUE).
- x2: vector of length N containing the observed values of the smooth term (if use_f is TRUE).
- smooth_term: vector of length N containing a smooth term (if use_f is TRUE).
- Beta: array containing the true functional coefficients.

Examples

```
# Basic usage with default parameters
sim_data <- data_generator_vd()

# Generate data with non-aligned domains
non_aligned_data <- data_generator_vd(N = 150, J = 120, aligned = FALSE)

# Generate multivariate functional data
multivariate_data <- data_generator_vd(N = 200, J = 100, multivariate = TRUE)

# Generate data with non-functional covariates and smooth term
complex_data <- data_generator_vd(
  N = 100,
  J = 150,
  use_x = TRUE,
  use_f = TRUE
)

# Generate data with a different beta function and R-squared value
custom_beta_data <- data_generator_vd(
  N = 80,
  J = 80,
  beta_index = 2,
  Rsq = 0.8
)

# Access components of the generated data
y <- sim_data$y # Response variable
X_s <- sim_data$X_s # Noise-free functional covariate
X_se <- sim_data$X_se # Noisy functional covariate
```

---

ffvd   *Defining variable domain functional data terms in vd_fit formulae*

Description

Auxiliary function used to define ffvd terms within vd_fit model formulae. This term represents a functional predictor where each function is observed over a domain of varying length. The formulation is $\frac{1}{T_i} \int_1^{T_i} X_i(t)\beta(t, T_i)dt$, where $X_i(t)$ is a functional covariate of length T_i, and $\beta(t, T_i)$ is an unknown bivariate functional coefficient. The functional basis used to model this term is the B-spline basis.

Usage

```
ffvd(X, grid, nbasis = c(30, 50, 30), bdeg = c(3, 3, 3))
```

Arguments

X	variable domain functional covariate `matrix`.
grid	observation points of the variable domain functional covariate. If not provided, it will be `1:ncol(X)`.
nbasis	number of bspline basis to be used.
bdeg	degree of the bspline basis used.

Value

the function is interpreted in the formula of a VDPO model. `list` containing the following elements:

- An item named B design matrix.

- An item named X_hat smoothed functional covariate.

- An item named L_Phi and B_T 1-dimensional marginal B-spline basis used for the functional coefficient.

- An item named M matrix object indicating the observed domain of the data.

- An item named nbasis number of basis used.

Examples

```
# Generate sample data
set.seed(123)
data <- data_generator_vd(beta_index = 1, use_x = FALSE, use_f = FALSE)
X <- data$X_se

# Specifying a custom grid
custom_grid <- seq(0, 1, length.out = ncol(X))
ffvd_term_custom_grid <- ffvd(X, grid = custom_grid, nbasis = c(10, 10, 10))

# Customizing the number of basis functions
ffvd_term_custom_basis <- ffvd(X, nbasis = c(10, 10, 10))

# Customizing both basis functions and degrees
ffvd_term_custom <- ffvd(X, nbasis = c(10, 10, 10), bdeg = c(3, 3, 3))
```

---

vd_fit	*Estimation of the generalized additive functional regression models for variable domain functional data*

Description

The `vd_fit` function fits generalized additive functional regression models for variable domain functional data.

Usage

```
vd_fit(formula, data, family = stats::gaussian(), offset = NULL)
```

Arguments

formula	a formula object with at least one `ffvd` term.
data	a `list` object containing the response variable and the covariates as the components of the list.
family	a `family` object specifying the distribution from which the data originates. The default distribution is [gaussian]().
offset	An offset vector. The default value is NULL.

Value

An object of class `vd_fit`. It is a `list` containing the following items:

- An item named `fit` of class sop. See [sop.fit]().
- An item named `Beta` which is the estimated functional coefficient.
- An item named `theta` which is the basis coefficient of `Beta`.
- An item named `covar_theta` which is the covariance matrix of `theta`.
- An item named `M` which is the number of observations points for each curve.
- An item named `ffvd_evals` which is the result of the evaluations of the `ffvd` terms in the formula.

See Also

[ffvd]()

Examples

```
# VARIABLE DOMAIN FUNCTIONAL DATA EXAMPLE

# set seed for reproducibility
set.seed(42)

# generate example data
```

```
data <- data_generator_vd(
  N = 100,
  J = 100,
  beta_index = 1,
  use_x = TRUE,
  use_f = TRUE,
)

# Define a formula object that specifies the model behavior.
# The formula includes a functional form of the variable 'X_se' using 'ffvd'
# with a non-default number of basis functions ('nbasis' is set to c(10, 10, 10)).
# Additionally, it includes a smooth function 'f' applied to 'x2' with 10 segments ('nseg = 10'),
# a second-order penalty ('pord = 2'), and cubic splines ('degree = 3').
# The model also contains the linear term 'x1'.
formula <- y ~ ffvd(X_se, nbasis = c(10, 10, 10)) + f(x2, nseg = 10, pord = 2, degree = 3) + x1

# We can fit the model using the data and the formula
res <- vd_fit(formula = formula, data = data)

# Some important parameters of the model can be accesed as follows
res$Beta # variable domain functional coefficient
res$fit$fitted.values # estimated response variable

# Also, a summary of the fit can be accesed using the summary function
summary(res)

# And a heatmap for an specific beta can be obtained using the plot function
plot(res, beta_index = 1)
```

Index