

Package ‘ssofqrm’

March 22, 2025

Title Spatial Function-on-Function Quantile Regression Model

Version 1.0

Maintainer Muge Mutis <muge.mutis@yildiz.edu.tr>

Description

Functions for implementing methods for spatial scalar-on-function quantile regression model.

License GPL-3

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Depends R (>= 3.5)

Imports fda, fda.usc, quantreg, expm

LazyData true

LazyDataCompression xz

Contents

ssofqrm-package	2
air_data	2
chernozhukov_fun	3
dgp	4
fqrm	5
getPCA	6
getPCA_test	6
kim_fun	7
norm_wei	7
predict_fqrm	8
Index	9

ssofqrm-package

Spatial Scalar-on-Function Quantile Regression Model

Description

Functions for implementing spatial scalar-on-function quantile regression model.

Author(s)

Muge Mutis (<muge.mutis@yildiz.edu.tr>) [aut](#), [cre](#), [cph](#)

Ufuk Beyaztas [aut](#)

Filiz Karaman [aut](#)

Han Lin Shang [aut](#)

See Also

[fqrm](#), [dgp](#), [getPCA](#), etc.

air_data

Air Quality Data Example

Description

A dataset containing air quality metrics used in the spatial quantile regression models.

Usage

```
data(air_data)
```

Format

A list including three elements.

data_2023 Data for the year 2023.

data_2024 Data for the year 2024.

wie_mat Spatial weight matrix.

Source

<https://cran.r-project.org/web/packages/ARPALData/index.html>

Examples

```
## Not run:
data(air_data)
yy <- matrix(air_data$data_2023$PM2.5_mean, ncol = 365, byrow = T)
y <- apply(yy, 1, mean)
x <- matrix(air_data$data_2023$Ozone_max_8h, ncol = 365, byrow = T)
yy_test <- matrix(air_data$data_2024$PM2.5_mean, ncol = 366, byrow = T)[,1:365]
y_test <- apply(yy_test, 1, mean)
x_test <- matrix(air_data$data_2024$Ozone_max_8h, ncol = 366, byrow = T)[,1:365]
wei_mat <- air_data$wei_mat
fit_kim <- fqrm(y=y, x=x, w=wei_mat, tau=0.5, method = "KM")
fit_ch <- fqrm(y=y, x=x, w=wei_mat, tau=0.5, method = "Ch")
predict_kim <- predict_fqrm(object = fit_kim, xnew = x_test, wnew = wei_mat)
predict_ch <- predict_fqrm(object = fit_ch, xnew = x_test, wnew = wei_mat)

## End(Not run)
```

chernozhukov_fun

Chernozhukov Estimator for Spatial Quantile Regression

Description

Implements the Chernozhukov and Hansen (2006) estimation method for spatial quantile regression with instrumental variable correction and a grid search over possible spatial autocorrelation values.

Usage

```
chernozhukov_fun(y, x, w, tau)
```

Arguments

y	Numeric vector. Response variable of length n.
x	Numeric matrix of size n x p. Predictor variables.
w	Numeric matrix of size n x n. Spatial weight matrix.
tau	Numeric. Quantile level (between 0 and 1).

Value

A list with the following components:

- b0** Estimated intercept.
- rho** Estimated spatial autoregressive parameter.
- b** Estimated regression coefficients (excluding intercept).

dgp

*Data Generating Process for Spatial Scalar-on-Function Model***Description**

Generates synthetic data from a spatial scalar-on-function regression model.

Usage

```
dgp(n, rho, out.p = 0)
```

Arguments

<code>n</code>	Integer. Sample size.
<code>rho</code>	Numeric. Spatial autoregressive parameter.
<code>out.p</code>	Numeric. Proportion of outliers to introduce (default is 0).

Value

A list with the following elements:

y Response vector of length `n`.

x Matrix of functional predictor values.

w Spatial weight matrix of size `n` x `n`.

tcoefs True coefficient function used to generate the data.

out.index Indices of outlier observations, if any.

Examples

```
## Not run:
sim_data <- dgp(n = 250, rho=0.5)
y <- sim_data$y
x <- sim_data$x
w <- sim_data$w
tcoef <- sim_data$tcoefs

## End(Not run)
```

Description

Fits a spatial scalar-on-function quantile regression model using either the Kim and Muller (2004) or Chernozhukov and Hansen (2006) method.

Usage

```
fqrm(y, x, w, tau, argvals = NULL, nbasis = NULL, method = c("KM", "Ch"))
```

Arguments

y	Numeric vector of length n . Response variable.
x	Numeric matrix of size $n \times p$. Functional predictor observed on a common grid.
w	Numeric matrix of size $n \times n$. Spatial weight matrix.
tau	Numeric. Quantile level to estimate (between 0 and 1).
argvals	Numeric vector of length p . Grid of argument values for the functional predictor. If NULL, defaults to a regular grid on 0, 1 .
nbasis	Integer. Number of basis functions to use in FPCA. If NULL, defaults to $\min(10, p / 4)$.
method	Character string. Estimation method to use. One of "KM" (Kim et al.) or "Ch" (Chernozhukov et al.).

Value

A list containing:

- b** Estimated coefficient vector in FPCA space.
- b0** Estimated intercept.
- bhat** Estimated coefficient function in the original function space.
- rho** Estimated spatial autoregressive parameter.
- fitted.values** Fitted values from the model.
- residuals** Residuals from the model.
- tau** Quantile level used.
- fpca** Output from getPCA, including FPCA scores and basis functions.

Examples

```
## Not run:
sim_data <- dgp(n = 250, rho=0.5)
y <- sim_data$y
x <- sim_data$x
w <- sim_data$w
fit_kim <- fqrm(y=y, x=x, w=w, tau=0.5, method = "KM")
fit_ch <- fqrm(y=y, x=x, w=w, tau=0.5, method = "Ch")

## End(Not run)
```

getPCA

Functional Principal Component Analysis with B-spline Basis

Usage

```
getPCA(data, nbasis, gp)
```

Arguments

data Numeric matrix of size $n \times p$. Functional data observed over grid points in rows.

nbasis Integer. Number of B-spline basis functions.

gp Numeric vector of length p . Grid points corresponding to functional observations.

Value

A list with the following components:

PCAcoef An fd object representing estimated eigenfunctions.

PCAscore Matrix of principal component scores.

meanScore fd object for the mean function.

ncomp Number of components explaining more than 95% of the variance.

itembs_basis B-spline basis used.

itemgp Evaluation of basis functions at grid points.

itemgp The original grid points used.

Performs FPCA (Functional Principal Component Analysis) using a B-spline basis.

getPCA_test

Project Test Data onto FPCA Basis

Description

Computes principal component scores for new functional test data using an existing FPCA model.

Usage

```
getPCA_test(object, data)
```

Arguments

object A list object returned from [getPCA](#) containing FPCA model components (e.g., basis, eigenfunctions, mean function, grid points).

data Numeric matrix of size $n \times p$. Test data observed on the same grid used in training.

Value

A matrix of principal component scores for the test data.

kim_fun	<i>Kim and Muller (2004) Estimator for Spatial Quantile Regression</i>
---------	--

Description

Estimates spatial quantile regression parameters using the two-stage approach proposed by Kim et al.

Usage

```
kim_fun(y, x, w, tau)
```

Arguments

y	Numeric vector. Response variable of length n.
x	Numeric matrix of size n x p. Predictor variables.
w	Numeric matrix of size n x n. Spatial weight matrix.
tau	Numeric. Quantile level (between 0 and 1).

Value

A list with the following components:

- b0** Estimated intercept.
- rho** Estimated spatial autoregressive parameter.
- b** Estimated regression coefficients (excluding intercept and rho).

norm_wei	<i>Row-Normalize a Spatial Weight Matrix</i>
----------	--

Description

Checks whether a spatial weight matrix is row-normalized. If not, it normalizes the matrix by dividing each row by its sum.

Usage

```
norm_wei(W, tol = 1e-08)
```

Arguments

W	Numeric matrix. Spatial weight matrix of size n x n.
tol	Numeric. Tolerance for checking whether rows sum to 1. Default is 1e-8.

Value

A row-normalized version of the input matrix W.

predict_fqrm	<i>Predict Method for Functional Quantile Regression with Spatial Effects</i>
--------------	---

Description

Makes predictions for new observations using a fitted spatial functional quantile regression model.

Usage

```
predict_fqrm(object, xnew, wnew)
```

Arguments

object	A fitted model object returned by fqrm .
xnew	Numeric matrix of size $n \times p$. New functional predictor data (on the same grid used in training).
wnew	Numeric matrix of size $n \times n$. Spatial weight matrix for the new observations.

Value

A numeric vector of predicted values of length n .

Examples

```
## Not run:
sim_data <- dgp(n = 250, rho=0.5)
y <- sim_data$y
x <- sim_data$x
w <- sim_data$w
fit_kim <- fqrm(y=y, x=x, w=w, tau=0.5, method = "KM")
fit_ch <- fqrm(y=y, x=x, w=w, tau=0.5, method = "Ch")
sim_test <- dgp(n = 1000, rho=0.5)
y_test <- sim_test$y
x_test <- sim_test$x
w_test <- sim_test$w
predict_kim <- predict_fqrm(object = fit_kim, xnew = x_test, wnew = w_test)
predict_ch <- predict_fqrm(object = fit_ch, xnew = x_test, wnew = w_test)

## End(Not run)
```


Index

- * **datasets**
 - air_data, [2](#)
- 0, 1, [5](#)
- air_data, [2](#)
- aut, [2](#)
- aut, cre, cph, [2](#)
- chernozhukov_fun, [3](#)
- dgp, [2](#), [4](#)
- fqrm, [2](#), [5](#), [8](#)
- getPCA, [2](#), [6](#), [6](#)
- getPCA_test, [6](#)
- kim_fun, [7](#)
- norm_wei, [7](#)
- predict_fqrm, [8](#)
- ssofqrm (ssofqrm-package), [2](#)
- ssofqrm-package, [2](#)