Package 'rfsac'

June 24, 2024

Type Package

Title Robust Spatial Scalar-on-Function Regression

Version 1.0

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

Imports expm, fda.usc, pcaPP, methods

LazyLoad yes

ByteCompile TRUE

Encoding UTF-8

Maintainer Ufuk Beyaztas <ufukbeyaztas@gmail.com>

Description
Functions for implementing robust methods for spatial scalar-on-function linear regression.

License GPL-3

RoxygenNote 7.1.2

R topics documented:

`data_generation` *Generate a dataset for the spatial scalar-on-function regression model*

Description

This function can be used to generate a dataset for the spatial scalar-on-function regression model

$$Y = \rho WY + \int X(t)\beta(t)dt + \epsilon,$$

where Y denotes the scalar response, $X(t)$ denotes the functional predictor, W is the spatial weight matrix, $\beta(t)$ denotes the regression coefficient function, ρ is the spatial autocorrelation parameter, and ϵ is the error process.

Usage

```
data_generation(n, j, rho, sig.e, out.p = 0)
```

Arguments

`n`	An integer, specifying the number of observations for each variable to be generated.
`j`	An integer, denoting the number of grid points, i.e., a fine grid on the interval [0, 1].
`rho`	A numeric value, denoting the spatial autocorrelation parameter.
`sig.e`	A numeric value, denoting the standard deviation of the error process.
`out.p`	A numeric value between 0 and 1 specifying the outlier percentage.

Details

In the data generation process, first, the functional predictor is generated based on the following process:

$$X(t) = \sum_{j=1}^{5} \kappa_j v_j(t),$$

where κ_j is a vector generated from a Normal distribution with mean zero and variance $4j^{-3/2}$ and

$$v_j(t) = \sin(j\pi t) - \cos(j\pi t).$$

The regression coefficient function is generated from $\sin(2\pi t)$. The error process is generated from the standard normal distribution. The row-normalized spatial weight matrix is constructed based on a one-dimensional regular grid, with $w_{ii'} = \frac{1}{d_{ii'}}$ where $d_{ii'} = |i - i'|$, for $i \neq i'$.

Value

A list object with the following components:

y	An $n \times 1$-dimensional matrix containing the observations of simulated scalar response variable.
x	A matrix with dimension $n \times j$ containing the observations of simulated functional predictor variable.
w	A matrix with dimension $n \times n$, i.e., the generated spatial weight matrix.
tcoefs	A vector with length n containing the generated regression coefficient function.
out.index	A vector consisting of the index set of outlying observations.

Author(s)

Ufuk Beyaztas, Abhijit Mandal and Han Lin Shang

Examples

```
# Generate data
sim.dat <- data_generation(n = 100, j = 101, rho = 0.5, sig.e = 1, out.p = 0.05)

y <- sim.dat$y
x <- sim.dat$x
wei_mat <- sim.dat$w

# true coefficient
btrue <- c(sim.dat$tcoefs$data)

library(fda.usc)
fx <- fdata(x, argvals = seq(0, 1, length.out = 101))
par(mfrow = c(1,2))
plot(y, type = "p", pch = 16, xlab = "Index", ylab = "", main = "Response")
plot(fx, lty = 1, ylab = "", xlab = "Grid point",
main = expression(X(t)), mgp = c(2, 0.5, 0))
dev.off()
```

---

flm_pca *Scalar-on-function linear regression based on functional principal component analysis*

Description

This function can be used to perform scalar-on-function linear regression model

$$Y = \int X(t)\beta(t)dt + \epsilon,$$

based on the functional principal component decomposition of the functional predictor.

Usage

```
flm_pca(y, x, nbasis = NULL, gpx = NULL)
```

Arguments

y	An $n \times 1$-dimensional matrix containing the observations of scalar response Y, where n denotes the sample size.
x	A matrix with dimension $n \times j$ containing the observations of functional predictor $X(t)$, where n is the sample size and j denotes the number of grid points for $X(t)$.
nbasis	A numeric value denoting the number of B-spline basis expansion functions to be used to approximate the functional principal components for the functional predictor $X(t)$.
gpx	A vector containing the grid points of the functional predictor $X(t)$.

Value

A list object with the following components:

b	A vector containing the estimate of parameters of the regression model coducted between the scalar response and principal component scores of the functional predictor.
b0	A vector containing the estimate of intercept parameter of the regression model coducted between the scalar response and principal component scores of the functional predictor.
bhat	A vector containing the estimate of regression coefficient function.
fitted.values	An $n \times 1$-dimensional matrix containing the fitted values of the scalar response.
residuals	An $n \times 1$-dimensional matrix containing the residuals.
sig	Estimated standar deviation for the model.
fpca	A list object containing model details, such as number of basis functions, number of principal components, and grid points used for the functional predictor variable.

Author(s)

Ufuk Beyaztas, Abhijit Mandal and Han Lin Shang

Examples

```
# Generate data
sim.dat <- data_generation(n = 100, j = 101, rho = 0.5, sig.e = 1, out.p = 0.05)
y <- sim.dat$y
x <- sim.dat$x
wei_mat <- sim.dat$w

flmPCA <- flm_pca(y=y, x=x)
```

`fsac_pca` *Scalar-on-function linear quantile regression based on functional principal component analysis*

Description

This function can be used to perform spatial scalar-on-function linear regression model

$$Y = \rho YW + \int X(t)\beta(t)dt + \epsilon$$

based on both the usual and robust functional principal component decomposition of the functional predictor.

Usage

```
fsac_pca(y, x, nbasis = NULL, gpx = NULL, wei_mat,
         method.type = c("classical", "robust"),
         c1=1.4, c2=2.4, c3=1.65, tol = 1e-04)
```

Arguments

`y`	An $n \times 1$-dimensional matrix containing the observations of scalar response Y, where n denotes the sample size.
`x`	A matrix with dimension $n \times j$ containing the observations of functional predictor $X(t)$, where n is the sample size and j denotes the number of grid points for $X(t)$.
`nbasis`	A numeric value denoting the number of B-spline basis expansion functions to be used to approximate the functional principal components.
`gpx`	A vector containing the grid points of the functional predictor $X(t)$.
`wei_mat`	An $n \times n$ dimensional spatial weight matrix.
`method.type`	Method type used to estimate the spatial scalar-on-function linear regression model. Possibilities are "classical" and "robust".
`c1`	Tuning parameter.
`c2`	Tuning parameter.
`c3`	Tuning parameter.
`tol`	A numeric value used for the precision of the coefficient estimate.

Value

A list object with the following components:

`b`	A vector containing the estimate of parameters of the regression model coducted between the scalar response and principal component scores of the functional predictor.

b0	A vector containing the estimate of intercept parameter of the regression model coducted between the scalar response and principal component scores of the functional predictor.
bhat	A vector containing the estimate of regression coefficient function.
rho	Estimated spatial autocorrelation parameter.
sig	Estimated standar deviation for the model.
fitted.values	An $n \times 1$-dimensional matrix containing the fitted values of the scalar response.
residuals	An $n \times 1$-dimensional matrix containing the residuals.
fpca	A list object containing model details, such as number of basis functions, number of principal components, and grid points used for the functional predictor variable.
ncomp	An integer denoting the number of principal components used in the analysis.

Author(s)

Ufuk Beyaztas, Abhijit Mandal and Han Lin Shang

Examples

```
# Generate data
sim.dat <- data_generation(n = 100, j = 101, rho = 0.5, sig.e = 1, out.p = 0.05)
y <- sim.dat$y
x <- sim.dat$x
wei_mat <- sim.dat$w

# Spatial functional regression model based on FPCA decomposition
fsacPCA <- fsac_pca(y=y, x=x, wei_mat = wei_mat, method.type = "classical")
# Robust spatial functional regression model based on FPCA decomposition
RfsacPCA <- fsac_pca(y=y, x=x, wei_mat = wei_mat, method.type = "robust")
```

---

fsac_pls *Scalar-on-function linear quantile regression based on functional partial least squares*

Description

This function can be used to perform spatial scalar-on-function linear regression model

$$Y = \rho YW + \int X(t)\beta(t)dt + \epsilon$$

based on both the usual and robust functional partial least squares decomposition of the functional predictor.

Usage

```
fsac_pls(y, x, h, nbasis = NULL, gpx = NULL, wei_mat,
         method.type = c("classical","robust"),
         probp1 = 0.95, hampelp2 = 0.975, hampelp3 = 0.999,
         maxit = 1000, conv = 0.01,
         c1=1.4, c2=2.4, c3=1.65, tol = 1e-04)
```

Arguments

y	An $n \times 1$-dimensional matrix containing the observations of scalar response Y, where n denotes the sample size.
x	A matrix with dimension $n \times j$ containing the observations of functional predictor $X(t)$, where n is the sample size and j denotes the number of grid points for $X(t)$.
h	A numeric value denoting the number of functional partial least regression components to be computed.
nbasis	A numeric value denoting the number of B-spline basis expansion functions to be used to approximate the functional principal components.
gpx	A vector containing the grid points of the functional predictor $X(t)$.
wei_mat	An $n \times n$ dimensional spatial weight matrix.
method.type	Method type used to estimate the spatial scalar-on-function linear regression model. Possibilities are "classical" and "robust".
probp1	A numeric value used to determine the first outlier cutoff point for the weights.
hampelp2	A numeric value used to determine the first outlier cutoff point for the weights.
hampelp3	A numeric value used to determine the third outlier cutoff point for the weights.
maxit	An integer value defining the maximum iteration used to achieve convergence.
conv	A numeric value used for the precision of the coefficient estimate.
c1	Tuning parameter.
c2	Tuning parameter.
c3	Tuning parameter.
tol	Tolerance for the approximation of M-estimation.

Value

A list object with the following components:

b	A vector containing the estimate of parameters of the regression model coducted between the scalar response and partial least squares scores of the functional predictor.
b0	A vector containing the estimate of intercept parameter of the regression model coducted between the scalar response and partial least squares scores of the functional predictor.
bhat	A vector containing the estimate of regression coefficient function.

rho	Estimated spatial autocorrelation parameter.
sig	Estimated standar deviation for the model.
fitted.values	An $n \times 1$-dimensional matrix containing the fitted values of the scalar response.
residuals	An $n \times 1$-dimensional matrix containing the residuals.
fdd	A list object containing model details, such as number of basis functions, number of partial least squares components, and grid points used for the functional predictor variable.

Author(s)

Ufuk Beyaztas, Abhijit Mandal and Han Lin Shang

Examples

```
# Generate data
sim.dat <- data_generation(n = 100, j = 101, rho = 0.5, sig.e = 1, out.p = 0.05)
y <- sim.dat$y
x <- sim.dat$x
wei_mat <- sim.dat$w

fsacPCA <- fsac_pca(y=y, x=x, wei_mat = wei_mat, method.type = "classical")

# Spatial functional regression model based on FPLS decomposition
fsacPLS <- fsac_pls(y=y, x=x, h = fsacPCA$ncomp, wei_mat = wei_mat, method.type = "classical")
# Robust spatial functional regression model based on FPLS decomposition
RfsacPLS <- fsac_pls(y=y, x=x, h = fsacPCA$ncomp, wei_mat = wei_mat, method.type = "robust")
```

---

predict_flm_pca *Prediction for a scalar-on-function linear regression model based on functional principal component analysis*

Description

This function is used to make prediction for a new set of functional predictors based upon a fitted scalar-on-function linear regression model in the output of flm_pca.

Usage

```
predict_flm_pca(object, xnew)
```

Arguments

object	An output object obtained from flm_pca.
xnew	A matrix consisting of the new observations of functional predictor. The argument xnew must have the same length and the same structure as the input x of flm_pca.

Value

An $n_{test} \times 1$-dimensional matrix of predicted values of the scalar response variable for the given set of new functional predictor xnew. Here, n_{test}, the number of rows of the matrix of predicted values, equals to the number of rows of xnew.

Author(s)

Ufuk Beyaztas, Abhijit Mandal and Han Lin Shang

Examples

```
# Generate data
sim.dat <- data_generation(n = 100, j = 101, rho = 0.5, sig.e = 1, out.p = 0.05)
y <- sim.dat$y
x <- sim.dat$x
wei_mat <- sim.dat$w

# Generate test data
test.dat <- data_generation(n = 100, j = 101, rho = 0.5, sig.e = 1)
x.test <- test.dat$x
y.test <- test.dat$y
w.test <- test.dat$w

flmPCA <- flm_pca(y=y, x=x)
yhat_flm <- predict_flm_pca(object = flmPCA, xnew = x.test)
```

---

predict_fsac_pca *Prediction for a spatial scalar-on-function linear regression model based on functional principal component analysis*

Description

This function is used to make prediction for a new set of spatial weight matrix and functional predictors based upon a fitted spatial scalar-on-function linear regression model in the output of fsac_pca.

Usage

```
predict_fsac_pca(object, xnew, wnew)
```

Arguments

object	An output object obtained from fsac_pca.
xnew	A matrix consisting of the new observations of functional predictor. The argument xnew must have the same length and the same structure as the input x of fsac_pca.
wnew	A matrix consisting of spatial weight matrix for the test set.

Value

An $n_{test} \times 1$-dimensional matrix of predicted values of the scalar response variable for the given spatial weight matrix and a set of new functional predictor xnew. Here, n_{test}, the number of rows of the matrix of predicted values, equals to the number of rows of xnew.

Author(s)

Ufuk Beyaztas, Abhijit Mandal and Han Lin Shang

Examples

```
# Generate data
sim.dat <- data_generation(n = 100, j = 101, rho = 0.5, sig.e = 1, out.p = 0.05)
y <- sim.dat$y
x <- sim.dat$x
wei_mat <- sim.dat$w

# Generate test data
test.dat <- data_generation(n = 100, j = 101, rho = 0.5, sig.e = 1)
x.test <- test.dat$x
y.test <- test.dat$y
w.test <- test.dat$w

# Spatial functional regression model based on FPCA decomposition
fsacPCA <- fsac_pca(y=y, x=x, wei_mat = wei_mat, method.type = "classical")
yhat_PCA <- predict_fsac_pca(object = fsacPCA, xnew = x.test, wnew = w.test)
# Robust spatial functional regression model based on FPCA decomposition
RfsacPCA <- fsac_pca(y=y, x=x, wei_mat = wei_mat, method.type = "robust")
yhat_RPCA <- predict_fsac_pca(object = RfsacPCA, xnew = x.test, wnew = w.test)
```

---

predict_fsac_pls *Prediction for a spatial scalar-on-function linear regression model based on functional partial least squares*

Description

This function is used to make prediction for a new set of spatial weight matrix and functional predictors based upon a fitted spatial scalar-on-function linear regression model in the output of fsac_pls.

Usage

```
predict_fsac_pls(object, xnew, wnew)
```

Arguments

object	An output object obtained from fsac_pls.
xnew	A matrix consisting of the new observations of functional predictor. The argument xnew must have the same length and the same structure as the input x of fsac_pls.
wnew	A matrix consisting of spatial weight matrix for the test set.

Value

An $n_{test} \times 1$-dimensional matrix of predicted values of the scalar response variable for the given spatial weight matrix and a set of new functional predictor xnew. Here, n_{test}, the number of rows of the matrix of predicted values, equals to the number of rows of xnew.

Author(s)

Ufuk Beyaztas, Abhijit Mandal and Han Lin Shang

Examples

```
# Generate data
sim.dat <- data_generation(n = 100, j = 101, rho = 0.5, sig.e = 1, out.p = 0.05)
y <- sim.dat$y
x <- sim.dat$x
wei_mat <- sim.dat$w

# Generate test data
test.dat <- data_generation(n = 100, j = 101, rho = 0.5, sig.e = 1)
x.test <- test.dat$x
y.test <- test.dat$y
w.test <- test.dat$w

fsacPCA <- fsac_pca(y=y, x=x, wei_mat = wei_mat, method.type = "classical")

# Spatial functional regression model based on FPCA decomposition
fsacPLS <- fsac_pls(y=y, x=x, h = fsacPCA$ncomp, wei_mat = wei_mat, method.type = "classical")
yhat_PLS <- predict_fsac_pls(object = fsacPLS, xnew = x.test, wnew = w.test)
# Robust spatial functional regression model based on FPCA decomposition
RfsacPLS <- fsac_pls(y=y, x=x, h = fsacPCA$ncomp, wei_mat = wei_mat, method.type = "robust")
yhat_RPLS <- predict_fsac_pls(object = RfsacPLS, xnew = x.test, wnew = w.test)
```

Index