Package ‘KFPLS’

January 10, 2023

Title Kernel Functional Partial Least Squares

Version 1.0

Description
Implementation for kernel functional partial least squares (KFPLS) method. KFPLS method is developed for functional nonlinear models, and the method does not require strict constraints for the nonlinear structures. The crucial function of this package is KFPLS().

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.1.1

Imports fda, splines, stats

NeedsCompilation no

Author Rou Zhong [aut, cre],
Jingxiao Zhang [aut]

Maintainer Rou Zhong <zhong_rou@163.com>

Repository CRAN

Date/Publication 2023-01-10 13:03:12 UTC

R topics documented:

`KFPLS` *Kernel functional partial least squares method*

Description

Kernel functional partial least squares (KFPLS) method for functional nonlinear models with scalar response and functional predictors. The Gaussian kernel is used.

Usage

```
KFPLS(X, Y, obser_time, nfold, n_comp, sigm_list, basis)
```

Arguments

X — An array with three indices. The (i, j, k)-th element of it corresponds to the measurment of the i-th subject for the k-th functional predictor at j-th observation grid.

Y — A vector with length n, where n is the sample size. The i-th element of it corresponds to the measurement of the scalar response for the i-th subject.

obser_time — A vector denoting the observation times of the functional predictors.

nfold — An integer denoting the number of folds for the selection of the tuning parameters by cross-validation.

n_comp — A vector denoting the candidates of the number of components.

sigm_list — A vector denoting the candidates of the tuning parameter for the Gaussian kernel.

basis — A basis object denoting the basis that used for the smoothing of the functional predictors. It is created by functions in fda package, such as create.bspline.basis.

Value

A list containing the following components:

n — A scalar denoting the sample size.

p — A scalar denoting the number of functional predictors.

nk — A scalar denoting the selected number of components.

T — A matrix denoting the value of T at convergence.

U — A matrix denoting the value of U at convergence.

K — A matrix denoting the Gram matrix.

K_c — A matrix denoting the centralized Gram matrix.

Xfd_list — A list of length p. The k-th entry corresponds to the functional data object of the k-th functional predictor.

XX_list — A list of length p. The k-th entry corresponds to the matrix that denotes the inner product of the k-th functional predictor for all subjects.

Y_c — A vector denoting the centralized scalar response.

meanY — A scalar denoting the sample mean of the scalar response.

Y_hat — A vector denoting the prediction of the scalar response.

obser_time — A vector denoting the observation times of the functional predictors.

basis — A basis object denoting the basis that used for the smoothing of the functional predictors.

sigm — A scalar denoting the selected tuning parameter for the Gaussian kernel.

CVscore — A matrix denoting the CV scores.

time — A scalar denoting the computation time.

Examples

```
# Generate data
n <- 200
t_range <- c(0, 1)
obser_time <- seq(0, 1, length.out = 51)
beta_fun <- function(t){2 * sin(2 * pi * t)}
basis <- fda::create.bspline.basis(t_range, nbasis = 13, norder = 4,
breaks = seq(0, 1, length.out = 11))
beta_fd <- fda::smooth.basis(obser_time, beta_fun(obser_time), basis)$fd
X_basis <- fda::create.bspline.basis(t_range, nbasis = 23, norder = 4,
breaks = seq(0, 1, length.out = 21))
Bbeta <- fda::inprod(X_basis, beta_fd)
Xi_B <- splines::bs(obser_time, knots = seq(0, 1, length.out = 21)[-c(1, 21)],
degree = 3, intercept = TRUE)
a <- array(0, dim = c(n, 23, 1))
X <- array(0, dim = c(n, 51, 1))
Y <- NULL
for(i in 1:n){
a[i, , 1] <- stats::rnorm(23)
X[i, , 1] <- Xi_B %*% a[i, , 1]
aBbeta <- as.numeric(t(a[i, , 1]) %*% Bbeta)
Y[i] <- aBbeta + stats::rnorm(1, mean = 0, sd = 0.05)
}
# KFPLS
KFPLS_list <- KFPLS(X, Y, obser_time, nfold = 5, n_comp = 5, sigm_list = 0.005, basis)
plot(KFPLS_list$Y_hat, Y)
lines(Y, Y)
```

`predict.KFPLS` *Prediction by KFPLS*

Description

Prediction of the scalar response by KFPLS.

Usage

```
## S3 method for class 'KFPLS'
predict(object, newdata, ...)
```

Arguments

`object`	A KFPLS object obtained from `KFPLS`.
`newdata`	An array with three indices denoting the new observations of the functional predictors. The (i, j, k)-th element of it corresponds to the measurment of the i-th subject for the k-th functional predictor at j-th observation grid.
`...`	Not used.

Value

A `vector` denoting the prediction of the scalar response.

Index