Package 'LocKer'

October 12, 2022

Title Locally Sparse Estimator of Generalized Varying Coefficient Model for Asynchronous Longitudinal Data

Version 1.1

Description Locally sparse estimator of generalized varying coefficient model for asynchronous longitudinal data by kernel-weighted estimating equation.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.1.1

Imports fda, Matrix, psych, splines, stats

NeedsCompilation no

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

Maintainer Rou Zhong <zhong_rou@163.com>

Repository CRAN

Date/Publication 2022-03-27 23:10:12 UTC

R topics documented:

`LocKer` *Locally sparse estimator of generalized varying coefficient model for asynchronous longitudinal data.*

Description

Locally sparse estimator of generalized varying coefficient model for asynchronous longitudinal data by kernel-weighted estimating equation. The function is suitable for generalized varying coefficient model with one covariate.

Usage

```
LocKer(
  X,
  Y,
  family,
  X_obser_num,
  Y_obser_num,
  X_obser,
  Y_obser,
  timeint,
  L_list,
  roupen_para_list,
  lambda_list,
  absTol_list,
  nfold = 5,
  d = 3
)
```

Arguments

`X` A `list` of n vectors, where n is the sample size. Each entry contains the measurements of the covariate for each subject at the observation time correspond to `X_obser`.

`Y` A `list` of n vectors, where n is the sample size. Each entry contains the measurements of the response for each subject at the observation time correspond to `Y_obser`.

`family` A `character` string representing the distribution family of the response. The value can be "Gaussian", "binomial", "poisson".

`X_obser_num` A `vector` denoting the observation size of the covariate for each subject.

`Y_obser_num` A `vector` denoting the observation size of the response for each subject.

`X_obser` A `list` of n vectors, where n is the sample size. Each entry contains the observation times of the covariate for each subject.

`Y_obser` A `list` of n vectors, where n is the sample size. Each entry contains the observation times of the response for each subject.

`timeint` A `vector` of length two denoting the supporting interval.

`L_list` A `vector` denoting the candidates for the number of B-spline basis functions. The best `L` is chosen by cross-validation.

`roupen_para_list`
A `vector` denoting the candidates for the roughness parameters. The best roughness parameter is chosen by `EBIC` together with sparseness parameter.

`lambda_list` A `vector` denoting the candidates for the sparseness parameter. The best sparseness parameter is chosen by `EBIC` together with roughness parameter.

`absTol_list` A `vector` denoting the threshold of the norm for coefficient function on each sub-interval. The `vector` is related to `L_list`, with the same length as `L_list`.

nfold	An integer denoting the number of fold for the selection of L by cross-validation. (default: 5)
d	An integer denoting the degree of B-spline basis functions. (default: 3)

Value

A list containing the following components:

beta0fd_est	A functional data object denoting the estimated intercept function.
betafd_est	A functional data object denoting the estimated coefficient function.
time	A scalar denoting the computation time.
L	An integer denoting the selected number of B-spline basis function.
roupen_select	A scalar denoting the selected roughness parameter.
lambda_select	A scalar denoting the selected sparseness parameter.
EBIC	A matrix denoting the EBIC scores for various roughness parameters and sparseness parameters belongs to the candidates when using the selected L.

Examples

```
####Generate data
n <- 200
beta0 <- function(x){cos(2 * pi * x)}
beta <- function(x){sin(2 * pi * x)}
Y_rate <- 15
X_rate <- 15
Y_obser_num <- NULL
X_obser_num <- NULL
Y_obser <- list()
X_obser <- list()
for(i in 1:n){
Y_obser_num[i] <- stats::rpois(1, Y_rate) + 1
Y_obser[[i]] <- stats::runif(Y_obser_num[i], 0, 1)
X_obser_num[i] <- stats::rpois(1, X_rate) + 1
X_obser[[i]] <- stats::runif(X_obser_num[i], 0, 1)
}
## The covariate functions Xi(t)
X_basis <- fda::create.bspline.basis(c(0, 1), nbasis = 74, norder = 5,
breaks = seq(0, 1, length.out = 71))
a <- matrix(0, nrow = n, ncol = 74)
X <- list()
XY <- list() #X at the observation time of Y
muY <- list()
for(i in 1:n){
a[i,] <- stats::rnorm(74)
Xi_B <- splines::bs(X_obser[[i]], knots = seq(0, 1, length.out = 71)[-c(1, 71)],
degree = 4, intercept = TRUE)
X[[i]] <- Xi_B %*% a[i,]
Yi_B <- splines::bs(Y_obser[[i]], knots = seq(0, 1, length.out = 71)[-c(1, 71)],
degree = 4, intercept = TRUE)
XY[[i]] <- Yi_B %*% a[i,]
muY[[i]] <- beta0(Y_obser[[i]]) + XY[[i]] * beta(Y_obser[[i]])
}
Y <- list()
errY <- list()
for(i in 1:n){
errY[[i]] <- stats::rnorm(Y_obser_num[[i]], mean = 0, sd = 1)
Y[[i]] <- muY[[i]] + errY[[i]]
}
L_list <- 20
absTol_list <- 10^(-3)
roupen_para_list <- 1.5 * 10^(-3)
lambda_list <- c(0, 0.001, 0.002)
LocKer_list <- LocKer(X, Y, family = "Gaussian", X_obser_num, Y_obser_num, X_obser,
Y_obser, timeint = c(0, 1), L_list, roupen_para_list, lambda_list, absTol_list)
```

Index