Package: cvmaPLFAM (via r-universe)

September 10, 2024

Type Package

Title Cross-Validation Model Averaging for Partial Linear Functional Additive Models

Version 0.1.0

RoxygenNote 7.1.0

Imports fda, quadprog, mgcv, MASS, stats, utils

NeedsCompilation no

Author Shishi Liu [aut, cre], Jingxiao Zhang [aut]

Maintainer Shishi Liu <`liushishi_644@163.com`>

Description Produce an averaging estimate/prediction by combining all candidate models for partial linear functional additive models, using multi-fold cross-validation criterion. More details can be referred to Shishi Liu and Jingxiao Zhang. (2021) <arXiv:2105.00966>.

License GPL (>= 3)

Encoding UTF-8

Repository CRAN

Date/Publication 2023-06-09 08:50:07 UTC

Contents

`cvfolds` *Generate cross-validation folds*

Description

Randomly split the data indexes into `nfolds` folds.

Usage

```
cvfolds(nfolds, datasize)
```

Arguments

`nfolds`	The number of folds used in cross-validation.
`datasize`	The sample size.

Value

A `list`. Each element contains the index vector of sample data included in this fold.

Examples

```
# Given sample size 20, generate 5 folds
set.seed(1212)
cvfolds(5, 20)
#[[1]]
# [1]  6 11 14 16
#[[2]]
# [1]  3  5 10 18
#[[3]]
# [1]  4  7  8 19
#[[4]]
# [1]  2  9 12 15
#[[5]]
# [1]  1 13 17 20
```

`cvmaPLFAM` *Cross-Validation Model Averaging (CVMA) for Partial Linear Functional Additive Models (PLFAMs)*

Description

Summarize the estimate of weights for averaging across all candidate models for PLFAMs, using multi-fold cross-validation criterion, and the corresponding mean squared prediction error risk. Additionally, the results of AIC, BIC, SAIC and SBIC are delivered simultaneously.

Usage

```
cvmaPLFAM(
  Y,
  scalars,
  functional,
  Y.test = NULL,
  scalars.test = NULL,
  functional.test = NULL,
  tt,
  nump,
  numfpcs,
  nbasis,
  nfolds,
  ratio.train = NULL
)
```

Arguments

`Y` The vector of the scalar response variable.

`scalars` The design matrix of scalar predictors.

`functional` The matrix including records/measurements of the functional predictor.

`Y.test` Test data: The vector of the scalar response variable.

`scalars.test` Test data: The design matrix of scalar predictors.

`functional.test` Test data: The matrix including records/measurements of the functional predictor.

`tt` The vector of recording/measurement points for the functional predictor.

`nump` The number of scalar predictors in candidate models.

`numfpcs` The number of functional principal components (FPCs) for the functional predictor in candidate models.

`nbasis` The number of basis functions used for spline approximation.

`nfolds` The number of folds used in cross-validation.

`ratio.train` The ratio of data for training, if test data are `NULL`.

Value

A `list` of

`aic` Mean squared error risk in training data set, produced by AIC model selection method.

`bic` Mean squared error risk in training data set, produced by BIC model selection method.

`saic` Mean squared error risk in training data set, produced by SAIC model averaging method.

`sbic` Mean squared error risk in training data set, produced by SBIC model averaging method.

`cv` Mean squared error risk in training data set, produced by CVMA method.

`waic` The selected candidate model by AIC model selection method.

`wbic` The selected candidate model by BIC model selection method.

`wsaic` The weights for each candidate model by SAIC model averaging method.

`wsbic` The weights for each candidate model by SBIC model averaging method.

`wcv` The weights for each candidate model by CVMA method.

`predaic` Mean squared prediction error risk in test data set, produced by AIC model selection method.

`predbic` Mean squared prediction error risk in test data set, produced by BIC model selection method.

`predsaic` Mean squared prediction error risk in test data set, produced by SAIC model averaging method.

`predsbic` Mean squared prediction error risk in test data set, produced by SBIC model averaging method.

`predcv` Mean squared prediction error risk in test data set, produced by CVMA method.

Examples

```
# Generate simulated data
simdata = data_gen(R = 0.7, K = 1, n = 50, M0 = 20, typ = 1, design = 3)
dat1 = simdata[[1]]
scalars = dat1[,1:20]
fd = dat1[,21:120]
Y = dat1[,122]
tps = seq(0, 1, length.out = 100)

# Estimation
est_res = cvmaPLFAM(Y=Y, scalars = scalars, functional = fd, tt = tps,
       nump = 2, numfpcs = 3, nbasis = 50, nfolds = 5, ratio.train = 0.8)
# Weights estimated by CVMA method
est_res$wcv
# Prediction error risk on test data set
est_res$predcv
```

`data_gen` *Simulated data*

Description

Simulate sample data for illustration, including a `M0`-column design matrix of scalar predictors, a `100`-column matrix of the functional predictor, a one-column vector of `mu`, a one-column vector of `Y`, and a one-column vector of `testY`.

Usage

```
data_gen(R, K, n, M0 = 50, typ, design)
```

Arguments

`R` A scalar of value ranging from `0.1` to `0.9`. The ratio of `var(mu)/var(Y)`.

`K` A scalar. The number of replications.

`n` A scalar. The sample size of simulated data.

`M0` A scalar. True dimension of scalar predictors.

`typ` A scalar of value `1` or `2`. Type of the additive function for the functional predictor.

`design` A scalar of value `1`, `2`, or `3`. Designs 1, 2, 3 corresponding to simulation studies.

Value

A `list` of `K` simulated data sets. Each data set is of `matrix` type, whose first `M0` columns corresponds to the design matrix of scalar predictors, followed by the recording/measurement matrix of the functional predictor, and vectors `mu`, `Y`, `testY`.

Examples

```
library(MASS)
# Example: Design 1 in simulation study
data_gen(R = 0.6, K = 2, n = 10, typ = 1, design = 1)

# Example: Design 2 in simulation study
data_gen(R = 0.3, K = 3, n = 10, typ = 2, design = 2)

# Example: Design 3 in simulation study
data_gen(R = 0.9, K = 5, n = 20, typ = 1, design = 3)
```

`fpcscore` *Calculate functional principal component (fpc) scores*

Description

Conduct functional principal component analysis (FPCA) on the observation matrix of the functional predictor.

Usage

```
fpcscore(Z, nbasis, tt)
```

Arguments

`Z` An `n` by `nT` matrix. The recording/measurement matrix of the functional predictor.

`nbasis` The number of basis functions used for spline approximation.

`tt` The vector of recording/measurement points for the functional predictor.

Value

A `list` of

`score` An `n` by `nbasis` matrix. The estimated functional principal component scores.

`eigv` A vector of estimated eigen-values related to FPCA.

`varp` A vector of percents of variance explained related to FPCA.

Examples

```
# Generate a recording/measurement matrix of the functional predictor
fddata = matrix(rnorm(1000), nrow = 10, ncol = 100)
tpoints = seq(0, 1, length.out = 100)

library(fda)
# Using 20 basis functions for spline approximation
fpcscore(fddata, nbasis = 20, tt = tpoints)

# Generate simulated data
simdata = data_gen(R = 0.7, K = 1, n = 20, M0 = 20, typ = 1, design = 1)
# Extract functional data from 'simdata', columns (M0+1):(M0+100)
simfd = simdata[[1]][,21:120]
# Calculate fpc scores
fpcres = fpcscore(simfd, nbasis = 50, tt = seq(0, 1, length.out = 100))
fpcres$score
fpcres$eigv
cumsum(fpcres$varp)
```

`modelspec` *Generate non-nested candidate models*

Description

Specify non-nested candidate models, according to the prescribed number of scalar predictors and the number of functional principal components (FPCs). Each candidate model comprises at least one scalar predictor and one FPC, leading to a total number of candidate models (2^nump-1)*(2^numq-1).

Usage

```
modelspec(nump, numq)
```

Arguments

`nump` The number of scalar predictors used in candidate models.

`numq` The number of functional principal components (FPCs) used in candidate models.

Value

A `list` of

`a1` The number of scalar predictors in each candidate model.

`a2` The number of FPCs in each candidate model.

`a3` The index for each component in each candidate model.

Examples

```
# Given nump = 2 and numq = 2, resulting in 9 candidate models
modelspec(2, 2)
#$a1
#[1] 2 2 2 1 1 1 1 1 1
#$a2
#[1] 2 1 1 2 1 1 2 1 1
#$a3
#      [,1] [,2] [,3] [,4]
# [1,]    1    2    3    4
# [2,]    1    2    3    0
# [3,]    1    2    0    4
# [4,]    1    0    3    4
# [5,]    1    0    3    0
# [6,]    1    0    0    4
# [7,]    0    2    3    4
# [8,]    0    2    3    0
# [9,]    0    2    0    4
```

`predRisk` *Output the prediction risks of each method for partial linear functional additive models (PLFAMs)*

Description

Calculate the estimated weights for averaging across all candidate models and the corresponding mean squared prediction error risk. The methods include AIC, BIC, SAIC, SBIC, and CVMA for PLFAMs.

Usage

```
predRisk(M, nump, numq, a2, a3, nfolds, XX.train, Y.train, XX.pred, Y.pred)
```

Arguments

`M` The number of candidate models.

`nump` The number of scalar predictors in candidate models.

`numq` The number of funtional principal components (FPCs) in candidate models.

`a2` The number of FPCs in each candidate model. See `modelspec`.

`a3` The index for each component in each candidate model. See `modelspec`.

`nfolds` The number of folds used in cross-validation.

`XX.train` The training data of predictors processed.

`Y.train` The training data of response variable.

`XX.pred` The test data of predictors processed.

`Y.pred` The test data of response variable.

Value

A `list` of

`aic` Mean squared error risk in training data set, produced by AIC model selection method.

`bic` Mean squared error risk in training data set, produced by BIC model selection method.

`saic` Mean squared error risk in training data set, produced by SAIC model averaging method.

`sbic` Mean squared error risk in training data set, produced by SBIC model averaging method.

`cv` Mean squared error risk in training data set, produced by CVMA method.

`ws` A `list` of weights estimator for five methods.

`predaic` Mean squared prediction error risk in test data set, produced by AIC model selection method.

`predbic` Mean squared prediction error risk in test data set, produced by BIC model selection method.

`predsaic` Mean squared prediction error risk in test data set, produced by SAIC model averaging method.

`predsbic` Mean squared prediction error risk in test data set, produced by SBIC model averaging method.

`predcv` Mean squared prediction error risk in test data set, produced by CVMA method.

Index