Package 'cvmaPLFAM'

April 28, 2025

Type Package

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

Version 0.1.1

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 arXiv e-Prints via <doi:10.48550/arXiv.2105.00966>.

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

Repository CRAN

Date/Publication 2025-04-28 02:40:01 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.

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

cv	Mean squared error risk in training data set, produced by CVMA method.
wcv	The weights for each candidate model by CVMA 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, ntest = 10, M0 = 4, typ = 1, design = 1)
train_dat = simdata[[1]]
scalars.train = train_dat[,1:4]
fd.train = train_dat[,5:104]
Y.train = train_dat[,106]


test_dat = simdata[[2]]
scalars.test = test_dat[,1:4]
fd.test = test_dat[,5:104]
Y.test = test_dat[,106]

tps = seq(0, 1, length.out = 100)

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

---

cvpredRisk *Output the prediction risks of the cross-validation model averaging (CVMA) 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.

Usage

```
cvpredRisk(
  M,
  nump,
  numq,
  a2,
  a3,
  nfolds,
  X.train,
  ZZ.train,
  Y.train,
```

```
  X.pred,
  ZZ.pred,
  Y.pred,
  nbasis,
  tt
)
```

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.
X.train	The training data of scalar predictors.
ZZ.train	The training data of the functional predictor.
Y.train	The training data of response variable.
X.pred	The test data of scalar predictors.
ZZ.pred	The test data of the functional predictor.
Y.pred	The test data of response variable.
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

cv	Mean squared error risk in training data set, produced by CVMA method.
ws	A vector of weights estimator.
predcv	Mean squared prediction error risk in test data set, produced by CVMA method.

---

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, ntest, M0, 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 training data.
ntest	A scalar. The sample size of test data.
M0	A scalar. True dimension of scalar predictors.
typ	A scalar of value 1 - 2. Type of the effect for the functional predictor.
design	A scalar of value 1 - 3. Correspond to simulation studies.

Value

A list of K simulated training data sets and K simulated test 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.

Examples

```
library(MASS)
# Example: Design 1 in simulation study
set.seed(22)
data1 <- data_gen(R = 0.6, K = 2, n = 10, ntest = 5, M0 = 4, typ = 1, design = 1)
str(data1)
# List of 4
#$ : num [1:10, 1:106] -0.501 -1.266 -0.564 -0.563 -0.395 ...
#$ : num [1:10, 1:106] -1.207 -0.089 -0.782 0.123 0.66 ...
#$ : num [1:5, 1:106] 0.816 0.679 0.816 -0.563 -1.367 ...
#$ : num [1:5, 1:106] -0.089 -0.785 0.899 -0.785 -0.445 ...


# Example: Design 2 in simulation study
data_gen(R = 0.3, K = 3, n = 10, ntest = 5, M0 = 20, typ = 1, design = 2)

# Example: Design 3 in simulation study
data_gen(R = 0.9, K = 5, n = 20, ntest = 10, M0 = 4, typ = 2, 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)
```

---

modelspec *Generate candidate models*

Description

Specify non-nested or 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.

Usage

```
modelspec(nump, numq, method = NULL)
```

Arguments

nump	The number of scalar predictors used in candidate models.
numq	The number of functional principal components (FPCs) used in candidate models.
method	A character string or NULL. If NULL, candidate models are generated under a non-nested structure. If "nested", candidate models are generated under a nested structure. Otherwise, an error will be raised.

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

```
# Example 1: non-nested models
# 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

# Example 2: nested models
# Given nump = 2 and numq = 3, resulting in 6 candidate models
modelspec(2, 3, method = "nested")
#$a1
# [1] 2 2 2 1 1 1
#$a2
# [1] 3 2 1 3 2 1
#$a3
#      [,1] [,2] [,3] [,4] [,5]
# [1,]    1    2    3    4    5
# [2,]    1    2    3    4    0
# [3,]    1    2    3    0    0
# [4,]    1    0    3    4    5
```

```
# [5,]    1    0    3    4    0
# [6,]    1    0    3    0    0
```

`plam.fit` *Fitting partial linear functional additive model*

Description

Calculate the prediction values and prediction errors across all candidate models.

Usage

```
plam.fit(
  M,
  nump,
  numq,
  a3,
  X.train,
  ZZ.train,
  y.train,
  X.pred,
  ZZ.pred,
  y.pred,
  nbasis,
  tt
)
```

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.
`a3`	The index for each component in each candidate model. See `modelspec`.
`X.train`	The training data of scalar predictors.
`ZZ.train`	The training data of the functional predictor.
`y.train`	The training data of response variable.
`X.pred`	The test data of scalar predictors.
`ZZ.pred`	The test data of the functional predictor.
`y.pred`	The test data of response variable.
`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

`muhat.train`	A `matrix` of prediction values on training data set for `M` candidate models.
`ehat.train`	A `matrix` of prediction errors on training data set for `M` candidate models.
`muhat.pred`	A `matrix` of prediction values on test data set for `M` candidate models.
`prederr`	A `matrix` of prediction errors on test data set for `M` candidate models.
`edf`	A `vector` of effective degree of freedom for `M` candidate models.

Index