Package 'gplsim'

February 11, 2023

Type Package

Title Spline Estimation for GPLSIM

Version 1.0.0

Date 2023-02-11

Description We provides functions that employ splines to estimate generalized partially linear single index models (GPLSIM), which extend the generalized linear models to include nonlinear effect for some predictors. Please see Y. (2017) at <doi:10.1007/s11222-016-9639-0> and Y., and R. (2002) at <doi:10.1198/016214502388618861> for more details.

License GPL-2

Encoding UTF-8

Depends R (>= 3.6.0)

Imports mgcv, stats, minpack.lm

LazyData true

RoxygenNote 7.2.1

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Tianhai Zu [aut, trl, cre], Yan Yu [aut]

Maintainer Tianhai Zu <zuti@mail.uc.edu>

Repository CRAN

Date/Publication 2023-02-11 16:50:02 UTC

R topics documented:

`add_sim_bound` *function dedicated to add simulation standard error bound, in development draw the bound to current plot*

Description

function dedicated to add simulation standard error bound, in development draw the bound to current plot

Usage

```
add_sim_bound(
  data,
  family = gaussian(),
  M = 200,
  n = 1000,
  true.theta = c(1, 1, 1)/sqrt(3)
)
```

Arguments

`data`	a list of simulated data
`family`	default is gaussian()
`M`	number of simulations
`n`	sample size
`true.theta`	the true coefficients

air *dataset from an environmental study.*

Description

This dataset contains four variables: The concentration of the air pollutant ozone, wind speed, temperature and radiation. All of them are daily measurements for 111 days. Usually the concentration of the air pollutant ozone serves as the response variable while the other three are predictors.

Usage

```
data("air")
```

Format

A data frame with 111 observations on the following 4 variables.

`ozone` a numeric vector in cube root ppb

`radiation` a numeric vector in langley

`temperature` a numeric vector in degrees F

`wind_speed` a numeric vector in mph

Examples

```
data(air)
y=air$ozone               # response
X=as.matrix(air[,3:4])    # single index term ;
Z=air[,2]                 # partially linear term ;

result <- gplsim(y,X,Z=Z,family = gaussian,k=10)
result$theta
result$coefficients
summary(result)

# Or you can try different spline basis
result <- gplsim(y,X,Z=Z,family = gaussian,bs="tp",k=10)
result$theta
result$coefficients
summary(result)
```

`generate_data` *Data generation function for simulation and demonstration A sine-bump setting has been employed.*

Description

Data generation function for simulation and demonstration A sine-bump setting has been employed.

Usage

```
generate_data(
  n,
  true.theta = c(1, 1, 1)/sqrt(3),
  family = "gaussian",
  ncopy = 1
)
```

Arguments

`n`	sample size
`true.theta`	true single-index coefficients, default is c(1,1,1)/sqrt(3) for setting 1 and c(1,2)/sqrt(5) for other settings
`family`	chose from "gaussian", "binomial" or "poisson".
`ncopy`	generates multiple copies of data for Monte Carlo simulations

Value

X single index predictors

Y response variables, a list

Z partial linear predictor(s)

single_index_values single index term

`gplsim` *Function to fit generalized partially linear single-index models via penalized splines*

Description

This function employs penalized spline (P-spline) to estimate generalized partially linear single index models, which extend the generalized linear models to include nonlinear effect for some predictors.

This function add formula interface to gplsim function

Usage

```
gplsim(...)

## Default S3 method:
gplsim(
  Y = Y,
  X = X,
  Z = Z,
  family = gaussian(),
  penalty = TRUE,
  profile = TRUE,
  user.init = NULL,
  bs = "ps",
  ...
)

## S3 method for class 'formula'
gplsim(
  formula,
  data,
  family = gaussian(),
  penalty = TRUE,
  profile = TRUE,
  user.init = NULL,
  bs = "ps",
  ...
)
```

Arguments

`...`	includes optional arguments user can pass to `mgcv::gam` or `glm`, such as `k`, which is the dimension of the basis of the smooth term and `m`, which is the order of the penalty for the smooth term. Others include: `scale` The optional argument scale is a numeric indicator with a default value set to -1. Any negative value including -1 indicates that the scale of response distribution is unknown, thus need to be estimated. Another option is 0 signaling scale of 1 for Poisson and binomial distribution and unknown for others. Any positive value will be taken as the known scale parameter. `smooth_selection` The optional argument smooth_selection is another character variable that specifies the criterion used in the selection of a smoothing parameter. The supported criteria include "GCV.Cp","GACV.Cp", "ML","P-ML", "P-REML" and "REML", while the default criterion is "GCV.Cp".
`Y`	Response variable, should be a vector.
`X`	Single index covariates.
`Z`	Partially linear covariates.
`family`	A `family` object: a list of functions and expressions for defining `link` and `variance` functions. Families supported are `binomial`, `gaussian`. The default

	family is gaussian.
penalty	Whether use penalized splines or un-penalized splines to fit the model. The default is TRUE.
profile	profile is a logical variable that indicates whether the algorithm with profile likelihood or algorithm with NLS procedure should be used. The default algorithm is set to algorithm with profile likelihood.
user.init	The user.init is a numeric vector of the same length as the dimensionality of single index predictors. The users can use this argument to pass in any appropriate user-defined initial single-index coefficients based on prior information or domain knowledge. The default value is NULL.
bs	bs is a character variable that specifies the spline basis in the estimation of unknown univariate function of single index. Default is P-splines.
formula	A model formula;
data	A data matrix containing the variables in the formula.

Details

For formula method, see ?gplsim.formula

Value

theta Estimation of Theta

coefficients the coefficients of the fitted model. Parametric coefficients are first, followed by coefficients for each spline term in turn.

... See GAM object

theta Estimation of Theta

coefficients the coefficients of the fitted model. Parametric coefficients are first, followed by coefficients for each spline term in turn.

... See GAM object

Examples

```
# parameter settings
n=200
true.theta = c(1, 1, 1)/sqrt(3)
# Gaussian case
# This function generate a plain sin bump model with gaussian response.
data <- generate_data(n,true.theta=true.theta,family="gaussian")
y=data$Y       # continous response
X=data$X       # single index term ;
Z=data$Z       # partially linear term ;

result <- gplsim(y,X,Z,family = gaussian)
result$theta
result$coefficients
summary(result)
```

```
#plot the estimated single index function curve
plot_si(result)
```

`plot_si` *Function that plot fitted curve for the unknown univariate function for single index term*

Description

Function that plot fitted curve for the unknown univariate function for single index term

Usage

```
plot_si(
  x,
  family = gaussian(),
  ylab = "mean",
  yscale = NULL,
  plot_data = FALSE
)
```

Arguments

`x`	the gam/gplism fitted object
`family`	default is gaussian()
`ylab`	y label
`yscale`	scale of y
`plot_data`	controls whether to plot the data as points

Value

NULL single-index plot

`Predict.matrix.tr.smooth` *prediction method function for the tr smooth class*

Description

prediction method function for the tr smooth class

Usage

```
Predict.matrix.tr.smooth(object, data)
```

Arguments

object	smooth object for gam class
data	the new data to predict on ’

Value

X the prediction matrix

---

print.summary.gplsim *Print Summary function of gplsim object*

Description

Print Summary function of gplsim object

Usage

```
## S3 method for class 'summary.gplsim'
print(
  x,
  digits = max(5, getOption("digits") - 3),
  signif.stars = getOption("show.signif.stars"),
  ...
)
```

Arguments

x	the gam/gplism fitted object
digits	controls number of digits printed in output.
signif.stars	should significance stars be printed alongside output.
...	optional arguments

Value

summarized object with nice format

si *An internal function to optimization and fitting. Don't use it solely.*

Description

An internal function to optimization and fitting. Don't use it solely.

Usage

```
si(
  alpha,
  y,
  x,
  z,
  opt = TRUE,
  smooth_selection,
  fam,
  bs = "ps",
  fx = FALSE,
  scale = scale,
  ...
)
```

Arguments

alpha	single-index coefficients
y	Response variable, should be a vector.
x	Single index covariates.
z	Partially linear covariates.
opt	see ?gplsim
smooth_selection	see ?gplsim
fam	see ?gplsim
bs	see ?gplsim
fx	see ?gplsim
scale	see ?gplsim
...	includes optional arguments user can pass to mgcv::gam or glm, such as k, which is the dimension of the basis of the smooth term and m, which is the order of the penalty for the smooth term

Value

b fitted gam object

smooth.construct.tr.smooth.spec *supporting function to make tr smooth*

Description

supporting function to make tr smooth

Usage

```
smooth.construct.tr.smooth.spec(object, data, knots)
```

Arguments

object	smooth object for gam class
data	the new data to predict on
knots	knots

Value

tr smooth object

summary.gplsim *Summary function of gplsim object*

Description

Summary function of gplsim object

Usage

```
## S3 method for class 'gplsim'
summary(object, ...)
```

Arguments

object	the gam/gplism fitted object
...	optional arguments

Value

gplsim_obj a list of summary information for a fitted gplsim object, which extends on gam object.

Index