

# GPLSIM: An R Package for Penalized Spline Estimation for Generalized Partially Linear Single-index Models

by Tianhai Zu and Yan Yu

**Abstract** Generalized partially linear single-index models (GPLSIM) are important tools in non-parametric regression. They extend popular generalized linear models to allow flexible nonlinear dependence on some predictors while overcoming the “curse of dimensionality”. We develop an R package **GPLSIM** that implements efficient penalized spline (P-spline) estimation of GPLSIM, proposed by Yu and Ruppert (2002) and Yu et al. (2017), for a response variable from a general exponential family. The package provides functions that allow users to fit GPLSIMs via penalized splines with various link functions, select smoothing tuning parameter  $\lambda$  against generalized cross validation or alternative choices, and visualize the estimated unknown univariate function of single-index term. In this paper, we discuss the implementation of **GPLSIM** in detail, and illustrate the use case through a sine-bump simulation study with various links and a real-data application to an air pollution data.

## Introduction

A popular approach to analyzing the relationship between a response variable and a set of predictors is generalized linear models or GLM (McCullagh and Nelder, 1989), where the conditional mean of the response variable is linked to a linear combination of predictors via a link function. Although GLM is simple and easy to interpret, in many complex real data applications the underlying linear assumption is often violated. Generalized partially linear single-index models (GPLSIM) (e.g. Carroll et al. (1997), Yu and Ruppert (2002), Yu et al. (2017)) are flexible semiparametric models that allow for a non-linear relationship while retaining ease of interpretation. In particular, GPLSIM include a partial linear component  $\mathbf{z}\gamma$ , and importantly a nonparametric single-index component, effectively reducing the dimensionality of  $p$ -dimensional predictors  $\mathbf{x}$  to a univariate single index  $\mathbf{x}\theta$  with a flexible univariate function  $\phi(\mathbf{x}\theta)$ , avoiding the “curse of dimensionality” in multivariate nonparametric regression. GPLSIM reduce to popular single-index models (Ichimura, 1993; Härdle et al., 1993; Xia and Härdle, 2006) when there are no partial linear terms. Another popular special case is partially linear models (Härdle et al., 2012) when there is only one predictor in the nonparametric component.

GPLSIM and the reduced models have been studied extensively in the literature. Applications lie in various fields, for example, discrete choice analysis, dose-response models, credit scoring, Framingham heart study etc. (Yu et al. (2017) and references there in). Yu and Ruppert (2002), Xia and Härdle (2006), and Liang et al. (2010) studied partially linear single-index models for continuous responses. For responses from a general exponential family, Carroll et al. (1997) proposed local linear approach via quasi-likelihood for GPLSIM estimation. However, as noted in Yu and Ruppert (2002), the algorithm using local methods in Carroll et al. (1997) may suffer from some computational issues and become unstable. Yu and Ruppert (2002) proposed a stable and computationally expedient approach using penalized splines (P-splines) with non-linear least square minimization. Yu et al. (2017) further proposed an efficient profile likelihood algorithm for the P-splines approach to GPLSIM.

We develop a package **GPLSIM** in R using penalized splines for efficient estimation of the unknown univariate function in GPLSIM following Yu and Ruppert (2002) and Yu et al. (2017). The **GPLSIM** R package mainly implements the P-spline profile likelihood estimation method in Yu et al. (2017) utilizing the function **gam** in the state-of-the-art R package **mgcv**. A similar object structure has been used for straightforward computation and implementation. A side benefit is that our **GPLSIM** package will enjoy improvements and features as those made to the **mgcv** package. For example, **mgcv** 1.5 added smoothness selection method “REML” and “ML” in addition to “GCV” to its core function “gam()”, and **GPLSIM** can enjoy those new features naturally. In addition, our **GPLSIM** package also implements the simultaneous non-linear least square minimization methods for continuous responses in Yu and Ruppert (2002) as an alternative option.

The rest of the paper is organized as follows. In the next section, we review the GPLSIM model and the penalized spline estimation for GPLSIM. Next, we discuss the estimation algorithm implemented in this package. The following section describes the main features of the functions provided. The section “real data and simulation examples” illustrates the use of **GPLSIM** in R via an air pollution example and a sine-bump simulation study. The last section concludes the paper.

## An overview of generalized partially linear single-index models

### The GPLSIM model

For given predictor vectors of  $p$ -dimensional  $X = \mathbf{x}$  and  $q$ -dimensional  $Z = \mathbf{z}$ , and under the assumption of a general exponential family of the conditional density of a response variable  $Y$ , the conditional mean  $E(Y|\mathbf{x}, \mathbf{z})$  can be modeled by

$$E(Y|\mathbf{x}, \mathbf{z}) := \mu(\mathbf{x}, \mathbf{z}) = g^{-1}\{\phi(\mathbf{x}\boldsymbol{\theta}) + \mathbf{z}\gamma\}, \quad (1)$$

where the single-index parameter  $\boldsymbol{\theta}$  maps the  $p$ -dimensional predictors  $\mathbf{x}$  to a univariate single index  $\mathbf{x}\boldsymbol{\theta}$  by a linear projection, and  $\phi(\cdot)$  is a univariate unknown function, while  $g\{\cdot\}$  is a known link function.  $\|\boldsymbol{\theta}\| = 1$  with first element  $\theta_1$  positive for identifiability (Yu and Ruppert, 2002).

One of the main challenges to estimate model (1) is that the  $p$ -dimensional single-index parameter  $\boldsymbol{\theta}$  is nested within the unknown univariate function  $\phi(\cdot)$ , and hence a highly nonlinear problem.

### Review of penalized spline estimation for GPLSIM

When the single-index parameter  $\boldsymbol{\theta}$  or the single-index  $u = \mathbf{x}\boldsymbol{\theta}$  is given, we can estimate the unknown univariate function  $\phi(\cdot)$  with penalized splines (Ruppert et al., 2003) such that  $\phi(u) \approx \mathbf{H}(u)\boldsymbol{\beta}$ . The systematic component of GPLSIM can then be approximated by

$$g\{\mu(\mathbf{x}, \mathbf{z})\} = \mathbf{H}(\mathbf{x}\boldsymbol{\theta})\boldsymbol{\beta} + \mathbf{z}\gamma, \quad (2)$$

where  $\boldsymbol{\omega} = (\boldsymbol{\theta}, \boldsymbol{\beta}, \gamma)$  is the column parameter vector,  $\mathbf{H}(\cdot)$  is the spline basis, and  $\boldsymbol{\beta}$  is the spline coefficient vector.

There are many choices of the spline basis functions  $\mathbf{H}(\cdot)$ , such as B-spline, truncated power basis, thin-plate spline, and their variations. For simplicity, we first illustrate using a truncated power basis of degree  $d$ :

$$\mathbf{H}(u)\boldsymbol{\beta} = \beta_0 + \beta_1 u + \cdots + \beta_d u^d + \sum_{k=1}^K \beta_{d+k} (u - v_k)_+^d,$$

where  $\mathbf{H}(u) = \{1, u, \dots, u^d, (u - v_1)_+^d, \dots, (u - v_K)_+^d\}$  are spline bases with  $K$  interior knots placed at  $(v_1, \dots, v_K)$ . Quadratic or cubic splines are commonly used. The interior knots are usually placed equally spaced or at equally-spaced quantiles within the domain.

Another popular choice of spline basis is the B-spline basis. Any B-spline basis functions  $\mathbf{H}(\cdot)$  of degree higher than 0, can be defined by the following Cox-de Boor recursion formula (Boor, 2001):

$$H_{k,d}(u) = \frac{u - u_k}{u_{k+d-1} - u_k} H_{k,d-1}(u) + \frac{u_{k+d} - u}{u_{k+d} - u_{k+1}} H_{k+1,d-1}(u),$$

where

$$H_{k,0}(u) = \begin{cases} 1, & u_k \leq u \leq u_{k+1} \\ 0, & \text{otherwise.} \end{cases}$$

One of the appealing features of the B-spline is that, unlike truncated power basis, B-spline basis functions have local supports that can result in high numerical stability.

To avoid overfitting, a roughness penalty controlled by a smoothing parameter  $\lambda$  is applied to the log-likelihood. Specifically, we can obtain the penalized log-likelihood estimator of  $\boldsymbol{\omega}$  by maximizing the following penalized log-likelihood function:

$$\begin{aligned} Q_{n,\lambda}(\boldsymbol{\omega}) &= \frac{1}{n} L_n(\boldsymbol{\omega}) - \frac{1}{2} \lambda \boldsymbol{\beta}^\top \mathbf{D} \boldsymbol{\beta} \\ &= \frac{1}{n} \sum_{i=1}^n [y_i \zeta(\mathbf{x}_i, \mathbf{z}_i; \boldsymbol{\omega}) - b\{\zeta(\mathbf{x}_i, \mathbf{z}_i; \boldsymbol{\omega})\}] - \frac{1}{2} \lambda \boldsymbol{\beta}^\top \mathbf{D} \boldsymbol{\beta}, \end{aligned} \quad (3)$$

where  $\zeta$  is the natural parameter in generalized linear models,  $\mu(\mathbf{x}_i, \mathbf{z}_i) = b'\{\zeta(\mathbf{x}_i, \mathbf{z}_i; \boldsymbol{\omega})\}$ , for observations  $i = 1, \dots, n$ , and  $\mathbf{D}$  is a positive semidefinite symmetric matrix. Common penalty matrix includes usual quadratic integral penalty on second derivatives of  $\phi(\cdot)$  or alternatively a diagonal penalty matrix with its last  $K$  diagonal elements equal to one and the rest equal to zero (see e.g. Ruppert and Carroll (2000); Yu and Ruppert (2002)), which in effect penalizes the coefficients of the truncated power basis at the jump of  $d$ -th derivatives.

Maximizing the penalized log-likelihood function (3) can be achieved in several ways. We mainly focus on implementing an efficient profile log-likelihood method in Yu et al. (2017). We also present an

option to implement a simultaneous nonlinear least square method in [Yu and Ruppert \(2002\)](#).

The selection of smoothing parameter  $\lambda$  is important as it controls the tradeoff between over-smoothing (possible underfitting) and under-smoothing (possible overfitting). We use an outer iteration to select  $\lambda$  against some selection criterion, as recommended by [Wood \(2011\)](#). For the default choice, we adopt generalized cross validation (GCV) to select the smoothing parameter  $\lambda$ . Alternatively, we can consider maximum likelihood (ML) ([Anderssen and Bloomfield, 1974](#)) or restricted maximum likelihood (REML) ([Wahba, 1985](#)) based approaches. A nice feature is that we can directly adopt criteria that have been provided by the “gam()” function arguments from R package **mgcv**, which is one of the main components in the implementation of our GPLSIM estimation algorithm.

## Algorithm

We present the main algorithm for fitting the generalized partially linear single-index models (GPLSIM) with penalized splines estimation with profile likelihood in detail as following:

---

**Input:** Non-linear predictor vector of  $p$ -dimensional  $X = \mathbf{x}$ , partially linear predictor vector of  $q$ -dimensional  $Z = \mathbf{z}$ , and a response vector  $Y = \mathbf{y}$  of family=family.

**Output:** The estimated single-index parameter  $\hat{\theta}$ , spline coefficient  $\hat{\beta}$ , partially linear coefficient  $\hat{\gamma}$ , and fitted response  $\hat{\mathbf{y}}$ .

- 1 Obtain an initial estimate  $\hat{\theta}^{(0)}$  of the single-index parameter  $\theta$  from a generalized linear model (default), or a user-provided initial list.
- 2 With an estimate of  $\theta$  (equivalently, the single index  $\{u_i = \mathbf{x}_i\theta : i = 1, \dots, n\}$ ), the spline coefficient  $\beta$  and partially linear coefficient  $\gamma$  can be written as implicit functions of  $\theta$  to maximize penalized log-likelihood:

$$\begin{aligned} Q(\beta, \gamma, \lambda; u_1, \dots, u_n) \\ = \frac{1}{n} \sum_{i=1}^n [y_i \xi(u_i, \mathbf{z}_i; \beta, \gamma) - b\{\xi(u_i, \mathbf{z}_i; \beta, \gamma)\}] \\ - \frac{1}{2} \lambda \beta^\top \mathbf{D} \beta. \end{aligned}$$

The roughness penalty parameter  $\lambda$  is selected using generalized cross-validation score (default) or alternative options.

- 3 Given the spline coefficient vector  $\hat{\beta}_\lambda(\theta)$  and partially linear coefficient vector  $\hat{\gamma}_\lambda(\theta)$  as implicit functions of  $\theta$ , obtain the profile log-likelihood estimator of the single-index parameter  $\theta$  by maximizing:

$$\begin{aligned} Q(\theta) = \frac{1}{n} \sum_{i=1}^n \left[ y_i \xi(\mathbf{x}_i\theta, \mathbf{z}_i; \hat{\beta}_\lambda(\theta), \hat{\gamma}_\lambda(\theta)) \right. \\ \left. - b\left\{ \xi(\mathbf{x}_i\theta, \mathbf{z}_i; \hat{\beta}_\lambda(\theta), \hat{\gamma}_\lambda(\theta)) \right\} \right]. \end{aligned}$$

- 4 With the estimated profile log-likelihood estimator  $\hat{\theta}$  of the single-index parameter, obtain the final estimator  $\hat{\beta}$  of spline coefficient and  $\hat{\gamma}$  of partially linear coefficient via step 2.
  - 5 Obtain the final fitted response vector  $\hat{\mathbf{y}}$  from model (1).
- 

Alternatively, for continuous responses under the default assumption of *family = gaussian*, maximizing the penalized log-likelihood estimator equation (3) is equivalent to minimizing the penalized sum of squared errors:

$$\frac{1}{n} \sum_{i=1}^n \{y_i - \mathbf{H}(\mathbf{x}_i\theta)\beta - \mathbf{z}_i\gamma\}^2 + \frac{1}{2} \lambda \beta^\top \mathbf{D} \beta.$$

For the simultaneous non-linear least square minimization methods in [Yu and Ruppert \(2002\)](#), we can directly apply a standard nonlinear least square (NLS) optimization algorithm on minimization of the above penalized sum of squared errors with respect to the full parameter  $\omega = (\theta, \beta, \gamma)$ . This is useful to facilitate joint inferences as described in [Yu and Ruppert \(2002\)](#). This algorithm as presented in [Yu and Ruppert \(2002\)](#) is also implemented in our GPLSIM package.

## The GPLSIM package

The R package **GPLSIM** consists of one core estimation function **GPLSIM** and some supporting functions such as visualization of the estimated curve for the unknown univariate function. The R package **GPLSIM** depends on the R package **mgcv** ([Wood, 2001](#)) and package **minpack.lm**.

## Main fitting function

The main estimation function **GPLSIM** implements the profile likelihood algorithm of Yu et al. (2017) as well as the non-linear least square method of Yu and Ruppert (2002) described in the previous section, while the profile likelihood being the default method for a response variable from a general exponential family.

The usage and input arguments of the main fitting function **GPLSIM** are summarized as follows:

```
gplsim(Y, X, Z, family = gaussian, penalty = TRUE, penalty_type = "L2", scale = -1,
       smoothing_select = "GCV.Cp", profile = TRUE, bs="ps", user.init=NULL)
```

This function takes three required arguments: the response variable  $Y$  in vector format, the single-index nonlinear predictors  $X$  in matrix or vector format and the linear predictors  $Z$  in matrix or vector format. Please note that all the input covariates are required to be numeric variables.

This function also takes several optional arguments for finer controls. The optional argument **family** is a family objects for models from the built-in R package **stats**. This object is essentially a list of functions and expressions for defining link and variance functions. Supported link functions include identity; logit, probit, cloglog; log; and inverse for the family distributions of gaussian, binomial, Poisson, and gamma respectively. The optional argument **penalty** is a logical variable to specify whether to use penalized splines or un-penalized splines to fit the model. The default value is TRUE to implement penalized splines. The optional argument **penalty\_type** is a character variable, which specifies the type of penalty used in the penalized splines estimation. The default penalty type is  $L_2$  penalty, while  $L_1$  is also supported. 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 and thus needs 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. The optional argument **smoothing\_select** is another character variable that specifies the criterion used in the selection of a smoothing parameter  $\lambda$ . The supported criteria include "GCV.Cp", "GACV.Cp", "ML", "P-ML", "P-REML" and "REML", while the default criterion is "GCV.Cp". The optional argument **profile** is a logical variable that indicates whether the algorithm with profile likelihood or algorithm with NLS procedure is used. The default algorithm is set to the profile likelihood algorithm. The optional argument **bs** is a character variable that specifies the spline basis in the estimation of unknown univariate function of single index. The default has been set to "ps" (P-splines with B-spline basis) while other choices are "tr" (truncated power basis), "tp" (thin plate regression splines) and others. The last optional argument **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, which instructs the function to estimate initial single-index coefficients by a generalized linear model. The function **GPLSIM** returns an object class of **GPLSIM**, which extends the **gam** object and **glm** object.

The documentation of **gam** object from the package **mgcv** can also be used as an external reference for some of the arguments as they are connected to the function **gam**. More specifically, **smoothing\_select** and **scale** in **gplsim** are passed into **gam** as the **method** argument and **scale** respectively.

## Other functions

```
plot.si(gplsim.object, reference = NULL)
```

This function plots the estimated curve for the unknown univariate function  $\phi$  from a **gplsim**-fitted model object. If the reference object is provided, this function will add a reference line accordingly.

```
summary.gplsim(gplsim.object)
print.summary.gplsim(gplsim.object)
```

The functions **summary.gplsim** and **print.summary.gplsim** provide detailed information related to the fitted model and summarize the results as illustrated in the next section. These two functions can be called directly by applying functions **print** and **summary** to **gplsim.object**.

```
simulation_data <- generate_data(n, true.beta=c(1, 1, 1)/sqrt(3), family="gaussian")
```

The function **generate\_data** generates data from a sine-bump model with user-defined single index coefficients  $\theta$  via the argument **true.beta**. If single-index coefficients  $\theta$  are not provided, this function will generate data against the default coefficients  $\theta = (1, 1, 1) / \sqrt{3}$ . The default response is Gaussian distributed, while binomial and Poisson distributions are also supported.

## Real data and simulation examples

In this section, we demonstrate the use of the R package **GPLSIM** via a real data analysis and a sine-bump simulation study.

### Air Pollution Data

We consider an environmental study on how meteorological variables  $X$  affect the concentration of the air pollutant ozone  $y$ . Meteorological variables  $X$  contains: wind speed, temperature, and radiation with  $n = 111$  daily measurements. As the response variable  $y$  is a continuous variable, we adopt an identity link for Gaussian distribution. Note that we use the same sequence of predictor variables to keep the results directly comparable to [Yu and Ruppert \(2002\)](#).

```
library(gplsim)
data(air)
y=air$ozone           # response
X=as.matrix(air[,c(3,4,2)]) # single-index term
colnames(X)=colnames(air[,c(3,4,2)])
Z=NULL
```

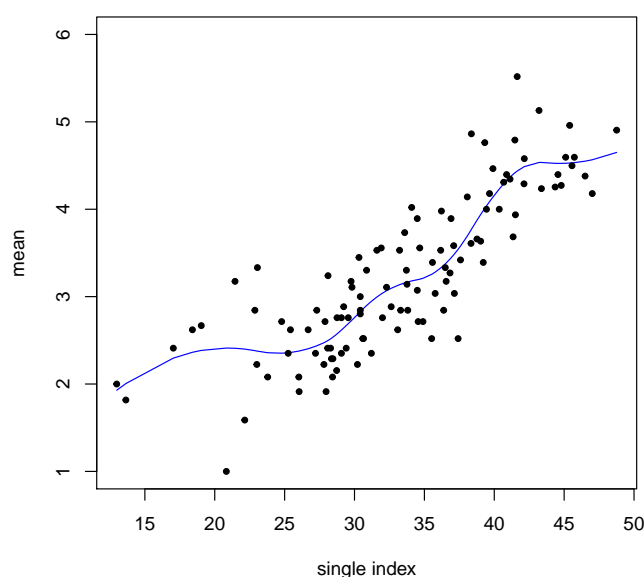
We allow all three predictor variables temperature, wind speed, and radiation entering the single-index term to capture the non-linear dependency as in [Yu and Ruppert \(2002\)](#). This model collapses to the single-index model as there is no partially linear term in the model.

```
air.fit <- gplsim(y,X,Z=NULL,family = gaussian,bs="ps")
summary(air.fit)

#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> y ~ s(a, bs = bs, fx = fx, m = 2, k = k)
#>
#> partial linear coefficients:
#>           Estimate Std. Error t value Pr(>|t|)
#> Intercept 3.247784    0.043024  75.488 < 2.2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> single index coefficients:
#>           Estimate
#> temperature    0.5442
#> wind_speed    -0.8386
#> radiation      0.0223
#>
#> Approximate significance of smooth terms:
#>           edf Ref.df      F    p-value
#> s(a) 8.1431  9.173 34.867 < 2.2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.741   Deviance explained =   76%
#> GCV = 0.22391   Scale est. = 0.20546    n = 111
```

The estimated normalized single-index coefficients with the profile likelihood algorithm are comparable to the results in [Yu and Ruppert \(2002\)](#). As shown in the figure below, the estimated unknown function is quite monotonic and exhibits clear curvature. The estimated coefficient is positive for temperature, negative for wind speed, positive for radiation but in a smaller magnitude per the reported summary.

```
plot.si(air.fit,yscale=c(1,6),plot_data = TRUE)
```



The above figure gives the single-index curve estimate of the air pollution data. The presence of curvature with multiple turning points is observed. This non-linear dependency is unlikely to capture by a linear model. The single index that contains information from temperature, wind speed, and radiation contributes to the ozone concentration differently on different segments.

We also implemented the simultaneous non-linear least square minimization algorithm in [Yu and Ruppert \(2002\)](#), where the original code was written in Matlab.

```
air.fit <- gplsim(y,X,Z=Z,family = gaussian,profile = FALSE,bs="ps")
summary(air.fit)

#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> y ~ s(a, bs = bs, fx = !penalty, m = 2, k = 13)
#>
#> partial linear coefficients:
#>           Estimate Std. Error t value Pr(>|t|)
#> Intercept 3.247784   0.043056  75.431 < 2.2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#>
#> single index coefficients:
#>           Estimate
#> temperature  0.5340
#> wind_speed  -0.8451
#> radiation    0.0235
#>
#> Approximate significance of smooth terms:
#>           edf Ref.df    F  p-value
#> s(a) 8.0012 9.0533 35.22 < 2.2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) =  0.74   Deviance explained = 75.9%
#> GCV = 0.22394   Scale est. = 0.20578    n = 111
```

## Simulations

We present a popular sine-bump simulation study that adopts the design as in (Carroll et al., 1997; Yu et al., 2017; Yu and Ruppert, 2002). The package can accommodate responses from a general exponential family, where the conditional mean is generated from the following model

$$g^{-1}\{\sin\{\pi(\mathbf{x}\boldsymbol{\theta} - c_1)/(c_2 - c_1)\} + z\gamma\},$$

where  $g\{\cdot\}$  is a link function;  $\boldsymbol{\theta} = (1, 1, 1)/\sqrt{3}$  with each predictor  $x$  from independent uniform in  $[0, 1]$ ;  $\gamma = 0.3$ ;  $z$  is a binary predictor with 1 for even observations and 0 otherwise;  $c_1 = \sqrt{3}/2 - 1.645/\sqrt{12}$  and  $c_2 = \sqrt{3}/2 + 1.645/\sqrt{12}$  are two constants.

For demonstration, we first show simulation codes and outputs on one random replication. We use the supporting function `generate_data` to generate simulation data.

```
set.seed(2020)
# Gaussian family
# parameter settings
n=1000
M=200
true.theta = c(1, 1, 1)/sqrt(3)
# This function generates a sine-bump simulation data
data <- generate_data(n,true.theta=true.theta,family="gaussian",ncopy=M)
y=(data$Y)[[1]]      # Gaussian error with standard deviation 0.1
X=data$X              # single-index predictors
Z=data$Z              # partially linear predictors
```

We use default settings of the main estimation function **GPLSIM** on the simulated data, assuming no prior information. The codes and summary results are provided as follows.

```
result <- gplsim(y,X,Z,user.init=NULL,family = gaussian)
summary(result)

#>
#> Family: gaussian
#> Link function: identity
#>
#> Formula:
#> y ~ s(a, bs = bs, fx = fx, m = 2, k = k) + z
#>
#> partial linear coefficients:
#>           Estimate Std. Error t value Pr(>|t|)
#> Intercept 0.6439549  0.0046579 138.251 < 2.2e-16 ***
#> Z.1        0.3057685  0.0065963  46.354 < 2.2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#>
#> single index coefficients:
#>           Estimate
#> X.1    0.5786
#> X.2    0.5798
#> X.3    0.5736
#>
#> Approximate significance of smooth terms:
#>           edf Ref.df      F    p-value
#> s(a) 6.3561 7.4656 2129.9 < 2.2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> R-sq.(adj) = 0.947   Deviance explained = 94.7%
#> GCV = 0.010909   Scale est. = 0.010818   n = 1000
```

From the above summary results of the fitted model, we see that the estimated single-index coefficients  $\hat{\boldsymbol{\theta}}$  and partial-linear coefficients  $\hat{\gamma}$  are quite close to the true parameters. Next, we plot the average estimated curve for the unknown univariate function over 200 replications. The dashed



lines are the corresponding 2.5 and 97.5 percentile bound. We observe that the average curve estimate virtually overlays the true curve.

```
#plot the estimated univariate function curve
plot.si(result,plot_data = FALSE)
par(new=T)
sort_index = order(X%%true.theta)
lines((X%%true.theta)[sort_index],data$single_index_values[sort_index],lty=1,
xaxt="n", yaxt="n",col="red")
legend("topright",legend=c("GPLSIM fit", "True"),lty=c(1,1),col = c("black","red"))
add_sim_bound(data)
```

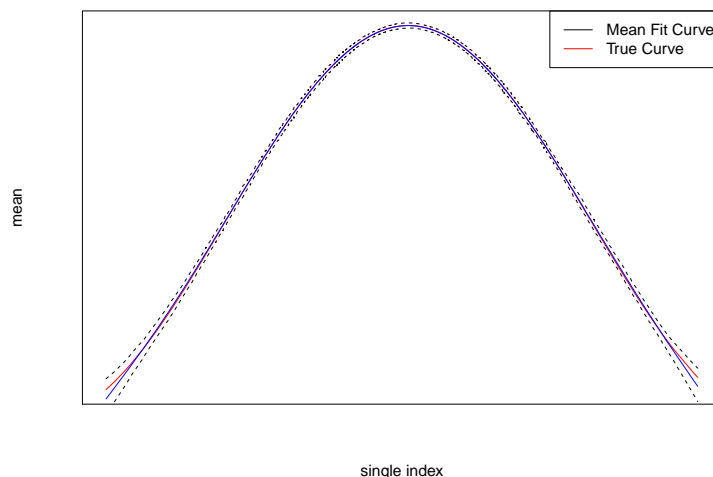


Table 1 reports the mean, standard error (se) and bias for each parameter estimate with sample size  $n = 1000$  over  $M = 200$  replications. We use canonical link functions, that is, identity link for Gaussian family, logit link for Binomial family, and log link for Poisson family. One can see that the algorithm for our R package **GPLSIM** is effective.

**Table 1:** Summary of parameter estimates for various responses of sample size  $n = 1000$ . True  $\theta = (1, 1, 1)/\sqrt{3}$ ,  $\gamma = 0.3$ . The sample mean (mean), standard error (se, in parenthesis), and bias of the parameter estimates from generalized partially linear single-index models (GPLSIM) by penalized splines from 200 replications.

	Gaussian		Binomial		Poisson	
	Mean(se)	Bias	Mean(se)	Bias	Mean(se)	Bias
$\hat{\theta}_1$	0.5771(0.0048)	-0.0002	0.5545(0.1040)	-0.0228	0.5808(0.0305)	0.0035
$\hat{\theta}_2$	0.5774(0.0048)	0.0005	0.5744(0.1111)	-0.0029	0.5738(0.0349)	-0.0035
$\hat{\theta}_3$	0.5765(0.0047)	-0.0004	0.5717(0.1126)	-0.0056	0.5745(0.0340)	-0.0028
$\hat{\gamma}$	0.2995(0.0065)	-0.0004	0.3094(0.1312)	0.0094	0.2978(0.0430)	-0.0022

## Summary

In this paper, we present an R package **GPLSIM** that implements penalized splines estimation of generalized partial linear single-index models in Yu et al. (2017) and Yu and Ruppert (2002). The approaches are able to accurately estimate the single-index coefficients, partial-linear coefficients, as well as the unknown univariate function with expedient computation. We believe this package is useful to practitioners in diverse fields such as finance, econometrics, and medicine, where a flexible and interpretable model is desirable.



## Bibliography

- R. S. Anderssen and P. Bloomfield. A Time Series Approach to Numerical Differentiation. *Technometrics*, 16(1):69–75, 1974. ISSN 0040-1706. doi: 10.2307/1267494. URL <https://www.jstor.org/stable/1267494>. Publisher: [Taylor & Francis, Ltd., American Statistical Association, American Society for Quality]. [p3]
- C. d. Boor. *A Practical Guide to Splines*. Springer, New York, Nov. 2001. ISBN 978-0-387-95366-3. [p2]
- R. J. Carroll, J. Fan, I. Gijbels, and M. P. Wand. Generalized Partially Linear Single-Index Models. *Journal of the American Statistical Association*, 92(438):477–489, 1997. ISSN 0162-1459. doi: 10.2307/2965697. URL <https://www.jstor.org/stable/2965697>. Publisher: [American Statistical Association, Taylor & Francis, Ltd.]. [p1, 7]
- W. Härdle, P. Hall, and H. Ichimura. Optimal Smoothing in Single-Index Models. *The Annals of Statistics*, 21(1):157–178, 1993. ISSN 0090-5364. URL <https://www.jstor.org/stable/3035585>. Publisher: Institute of Mathematical Statistics. [p1]
- W. Härdle, H. Liang, and J. Gao. *Partially Linear Models*. Springer Science & Business Media, Dec. 2012. ISBN 978-3-642-57700-0. Google-Books-ID: OpvyCAAQBAJ. [p1]
- H. Ichimura. Semiparametric least squares (SLS) and weighted SLS estimation of single-index models. *Journal of Econometrics*, 58(1):71–120, July 1993. ISSN 0304-4076. doi: 10.1016/0304-4076(93)90114-K. URL <http://www.sciencedirect.com/science/article/pii/030440769390114K>. [p1]
- H. Liang, X. Liu, R. Li, and C.-L. Tsai. ESTIMATION AND TESTING FOR PARTIALLY LINEAR SINGLE-INDEX MODELS. *The Annals of Statistics*, 38(6):3811–3836, 2010. ISSN 0090-5364. URL <https://www.jstor.org/stable/29765281>. Publisher: Institute of Mathematical Statistics. [p1]
- P. McCullagh and J. A. Nelder. *Generalized Linear Models, Second Edition*. CRC Press, Aug. 1989. ISBN 978-0-412-31760-6. Google-Books-ID: h9kFH2\_FfBkC. [p1]
- D. Ruppert and R. J. Carroll. Theory & Methods: Spatially-adaptive Penalties for Spline Fitting. *Australian & New Zealand Journal of Statistics*, 42(2):205–223, 2000. ISSN 1467-842X. doi: 10.1111/1467-842X.00119. URL <https://onlinelibrary.wiley.com/doi/abs/10.1111/1467-842X.00119>. \_eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/1467-842X.00119>. [p2]
- D. Ruppert, M. P. Wand, and R. J. Carroll. *Semiparametric regression*. Number 12. Cambridge university press, 2003. [p2]
- G. Wahba. A Comparison of GCV and GML for Choosing the Smoothing Parameter in the Generalized Spline Smoothing Problem. *Annals of Statistics*, 13(4):1378–1402, Dec. 1985. ISSN 0090-5364, 2168-8966. doi: 10.1214/aos/1176349743. URL <https://projecteuclid.org/euclid.aos/1176349743>. [p3]
- S. N. Wood. mgcv: GAMs and Generalized Ridge Regression for R. *R news*, Vol.1(2):20, June 2001. [p3]
- S. N. Wood. Fast stable restricted maximum likelihood and marginal likelihood estimation of semi-parametric generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(1):3–36, 2011. ISSN 1467-9868. doi: 10.1111/j.1467-9868.2010.00749.x. URL <https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2010.00749.x>. \_eprint: <https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-9868.2010.00749.x>. [p3]
- Y. Xia and W. Härdle. Semi-parametric estimation of partially linear single-index models. *Journal of Multivariate Analysis*, 97(5):1162–1184, May 2006. ISSN 0047-259X. doi: 10.1016/j.jmva.2005.11.005. URL <http://www.sciencedirect.com/science/article/pii/S0047259X05001995>. [p1]
- Y. Yu and D. Ruppert. Penalized Spline Estimation for Partially Linear Single-Index Models. *Journal of the American Statistical Association*, 97(460):1042–1054, 2002. ISSN 0162-1459. URL <https://www.jstor.org/stable/3085829>. Publisher: [American Statistical Association, Taylor & Francis, Ltd.]. [p1, 2, 3, 4, 5, 6, 7, 8]
- Y. Yu, C. Wu, and Y. Zhang. Penalised spline estimation for generalised partially linear single-index models. *Statistics and Computing*, 27(2):571–582, Mar. 2017. ISSN 0960-3174. doi: 10.1007/s11222-016-9639-0. URL <https://doi.org/10.1007/s11222-016-9639-0>. [p1, 2, 4, 7, 8]

*Tianhai Zu*  
*University of Cincinnati*  
*2906 Woodside Drive*  
*Cincinnati, OH 45221*  
*<https://orcid.org/0000-0002-4634-7937>*  
*[zuti@mail.uc.edu](mailto:zuti@mail.uc.edu)*

*Yan Yu*  
*University of Cincinnati*  
*2906 Woodside Drive*  
*Cincinnati, OH 45221*  
*<https://orcid.org/0000-0002-2859-3093>*  
*[Yan.YU@uc.edu](mailto:Yan.YU@uc.edu)*