




---

# *Journal of Statistical Software*

MMMMMM YYYY, Volume VV, Issue II.

<http://www.jstatsoft.org/>

---

## **cgam: An R Package for the Constrained Generalized Additive Model**

**Xiyue Liao**  
Colorado State University

**Mary C. Meyer**  
Colorado State University

---

### **Abstract**

The **cgam** package contains routines to fit the generalized additive model where the components may be modeled with shape and smoothness assumptions. The main routine is **cgam** and nineteen symbolic routines are provided to indicate the relationship between the response and each predictor, which satisfies constraints such as monotonicity, convexity, their combinations, tree, and umbrella orderings. The user may specify constrained splines to fit the components for continuous predictors, and various types of orderings for the ordinal predictors. In addition, the user may specify parametrically modeled covariates. The set over which the likelihood is maximized is a polyhedral convex cone, and a least-squares solution is obtained by projecting the data vector onto the cone. For generalized models, the fit is obtained through iteratively re-weighted cone projections. The cone information criterion is provided and may be used to compare fits for combinations of variables and shapes. In addition, the routine **wps** implements monotone regression in two dimensions using warped-plane splines, without an additivity assumption. The graphical routine **plotpersp** will plot an estimated mean surface for a selected pair of predictors, given an object of either **cgam** or **wps**. This package is now available from the Comprehensive R Archive Network at <http://CRAN.R-project.org/package=cgam>.

*Keywords:* constrained generalized additive model, isotonic regression, spline regression, partial linear, iteratively re-weighted cone projection, R, graphical routine.

---

## **1. Introduction and overview**

### **1.1. Generalized additive model with shape or order constraints**

We introduce a comprehensive framework for the generalized additive model with shape or order constraints. We consider models with independent observations from an exponential

family with density of the form

$$p(y_i; \theta, \tau) = \exp[\{y_i \theta_i - b(\theta_i)\} \tau - c(y_i, \tau)], \quad i = 1, \dots, n, \quad (1)$$

where the specifications of the functions  $b$  and  $c$  determine the sub-family of models. The mean vector  $\boldsymbol{\mu} = E(\mathbf{y})$  has values  $\mu_i = b'(\theta_i)$ , and is related to a design matrix of predictor variables through a link function  $g(\mu_i) = \eta_i$ ,  $i = 1, \dots, n$ . The link function specifies the relationship with the predictors; for example, suppose  $x_1, \dots, x_L$  are continuous or ordinal predictors and  $\mathbf{z} \in \mathbb{R}^p$  is a vector of covariates to be parametrically modeled. We specify an additive model

$$\eta_i = f_1(x_{1i}) + \dots + f_L(x_{Li}) + \mathbf{z}_i^\top \boldsymbol{\alpha}, \quad (2)$$

where the parameter vector  $\boldsymbol{\alpha} \in \mathbb{R}^p$  and the functions  $f_\ell$ ,  $\ell = 1, \dots, L$ , are to be estimated simultaneously. The  $\eta$  function has been called the “systematic component” or the “linear predictor” (McCullagh and Nelder (1989), Hastie and Tibshirani (1990)); here we will use “predictor function.” We consider the Gaussian, Poisson and binomial families in this package; the default is Gaussian.

For modeling smooth constrained  $f_\ell$ , there are eight shape choices, i.e., increasing, decreasing, concave, convex, and combinations of monotonicity and convexity. For increasing and decreasing constraints, we use quadratic *I-spline basis functions*, and for constraints involving convexity, cubic *C-spline basis functions* are used. Example sets of basis functions for seven equally spaced knots are shown in Figure 1; see Meyer (2008) for details about these spline bases.

The *I-spline basis functions*, together with constant function, span the space of piece-wise quadratic splines for the given knots. The spline function is increasing if and only if the coefficients of the basis functions are positive, and decreasing if and only if the coefficients of the basis functions are negative. The *C-spline basis functions*, together with the constant function and the identity function, span the space of piece-wise cubic splines for the given knots. The spline function is convex if and only if the coefficients of the basis functions are positive, and concave if and only if the coefficients of the basis functions are negative. If we also restrict the sign of the coefficient on the identity function, all four combinations of monotonicity and convexity can be modeled with constrained *C-splines*.

Define  $\boldsymbol{\phi}_\ell \in \mathbb{R}^n$  as  $\phi_{\ell,i} = f_\ell(x_{\ell,i})$ ,  $i = 1, \dots, n$ , for a continuous predictor  $x_\ell$ , and define  $\mathbf{s}_{\ell,j}$ ,  $j = 1, \dots, m_\ell$  to be the spline basis vectors appropriate for the constraints associated with  $x_\ell$ . The constraints are satisfied if  $\boldsymbol{\phi}_\ell \in \mathcal{C}_\ell$  where for *increasing or decreasing constraints*,

$$\mathcal{C}_\ell = \left\{ \boldsymbol{\phi} \in \mathbb{R}^n : \boldsymbol{\phi} = a_0 \mathbf{1} + \sum_{j=1}^{m_\ell} b_j \mathbf{s}_{\ell,j}, \quad b_j \geq 0, \quad j = 1, \dots, m_\ell \right\},$$

and for convex or concave constraints,

$$\mathcal{C}_\ell = \left\{ \boldsymbol{\phi} \in \mathbb{R}^n : \boldsymbol{\phi} = a_0 \mathbf{1} + a_1 \mathbf{x} + \sum_{j=1}^{m_\ell} b_j \mathbf{s}_{\ell,j}, \quad b_j \geq 0, \quad j = 1, \dots, m_\ell \right\}.$$

For ordinal predictors, constraint cones are defined according to Meyer (2013a). Similar to the previous case, there are eight shape constraints involving monotonicity and convexity; in

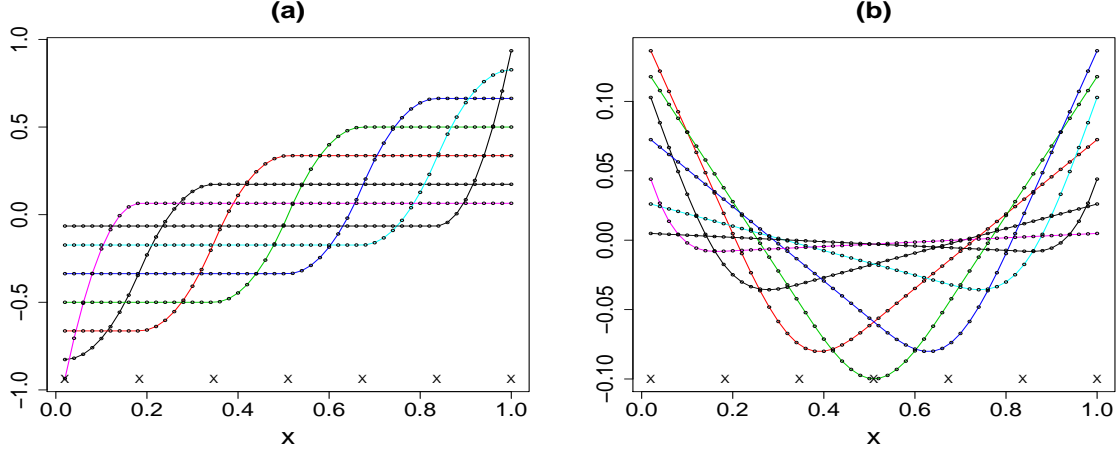


Figure 1: Smooth spline basis functions using seven equally spaced knots marked as “X” for a data set with  $n = 100$  observations with values marked as dots. (a) smooth and increasing with each basis function centered (b) smooth and convex with each basis function scaled to be orthogonal to  $\mathbf{1}$  and  $\mathbf{x}$ .

addition, tree and umbrella orderings are options. For these orderings, the estimate of  $\phi_\ell$  is in  $\mathcal{C}_\ell$ , where

$$\mathcal{C}_\ell = \{\phi_\ell \in \mathbb{R}^n : \mathbf{A}_\ell \phi_\ell \geq \mathbf{0} \text{ and } \mathbf{B}_\ell \phi_\ell = \mathbf{0}\},$$

for constraint matrices  $\mathbf{A}_\ell$  and  $\mathbf{B}_\ell$  that are  $r_{l1} \times n$  and  $r_{l2} \times n$ , respectively. The equality constraints handle duplicate values of the predictor, and the rows of  $\mathbf{A}_\ell$  and  $\mathbf{B}_\ell$  together form a linearly independent set.

The tree-ordering describes a scenario in which the user assumes that the effect of a categorical variable on the response, for all but one of the levels, is larger (or smaller) than the effect at that level. This ordering is useful in the case of several treatment levels and a placebo, where it is assumed that the treatment effect is at least that of the placebo, but there are no imposed orderings among the treatments. For implementation in `cgam`, the level zero is assumed to be the placebo level.

An umbrella ordering is a unimodal assumption on a categorical variable, where the level at which the maximum effect is given. For implementation in `cgam`, the level zero is used as this mode; other levels are indicated by positive or negative integers. The effects are ordered on either side of the mode.

See Meyer (2013a) for details about construction of basis vectors  $\mathbf{w}_1, \dots, \mathbf{w}_{m_\ell}$ , given  $\mathbf{A}_\ell$  and  $\mathbf{B}_\ell$ , so that

$$\mathcal{C}_\ell = \{\phi_\ell \in \mathbb{R}^n : \phi_\ell = \mathbf{v} + \sum_{j=1}^{m_\ell} b_j \mathbf{w}_j, \quad b_j \geq 0, \quad j = 1, \dots, m_\ell\}. \quad (3)$$

The vector  $\mathbf{v}$  is in a linear space  $\mathcal{V}_\ell$  defined by the shape assumptions. If monotonicity constraints are imposed,  $\mathcal{V}_\ell$  is the one-dimensional space of constant vectors; for other types of order constraints, see Meyer (2013a) for the construction and composition of  $\mathcal{V}_\ell$ .

Then  $\boldsymbol{\eta} = \phi_1 + \dots + \phi_L + \mathbf{Z}\boldsymbol{\beta}$ , where  $\phi_\ell \in \mathcal{C}_\ell$  for  $\ell = 1, \dots, L$ , and the rows of the matrix  $\mathbf{Z}$  are  $\mathbf{z}_i$ ,  $i = 1, \dots, n$ . Each set  $\mathcal{C}_\ell$  is a polyhedral convex cone, and let  $\mathcal{V}_z$  be the column space of  $\mathbf{Z}$ . Meyer (2013a) showed that  $\mathcal{C} = \mathcal{C}_1 + \dots + \mathcal{C}_L + \mathcal{V}_z$  is also a polyhedral convex cone,

where  $\boldsymbol{\eta} \in \mathcal{C}$  if  $\boldsymbol{\eta} = \boldsymbol{\phi}_1 + \cdots + \boldsymbol{\phi}_L + \mathbf{v}$  with  $\mathbf{v} \in \mathcal{V}_z$  and  $\boldsymbol{\phi}_\ell \in \mathcal{C}_\ell$ ,  $\ell = 1, \dots, L$ . That paper also showed how to find a linear space  $\mathcal{L}$  containing the linear spaces  $\mathcal{V}_1, \dots, \mathcal{V}_L$  and the column space of  $\mathbf{Z}$ , together with “edge” vectors  $\mathbf{e}_1, \dots, \mathbf{e}_m$  that are orthogonal to  $\mathcal{L}$ , so that we can write

$$\mathcal{C} = \left\{ \boldsymbol{\eta} \in \mathbb{R}^n : \boldsymbol{\eta} = \mathbf{v} + \sum_{j=1}^m \alpha_j \mathbf{e}_j + \mathbf{Z}\boldsymbol{\beta}, \text{ for } \mathbf{v} \in \mathcal{L}, \alpha_j \geq 0, j = 1, \dots, m \right\}.$$

## 1.2. Iteratively re-weighted cone projection

For the Gaussian family, fitting the additive model (2) involves a projection of the data vector  $\mathbf{y}$  onto  $\mathcal{C} \subseteq \mathbb{R}^n$ . This is accomplished using the `coneB` routine of the R package `coneproj` [Liao and Meyer \(2014\)](#). For binomial and Poisson families, an iteratively re-weighted cone projection algorithm is used. The negative log-likelihood

$$L(\theta, \tau; \mathbf{y}) = \sum_{i=1}^n \left\{ c(y_i, \tau) - \frac{y_i \theta_i - b(\theta_i)}{\tau} \right\}$$

is written in terms of the systematic component and minimized over  $\mathcal{C}$ . Let  $\ell(\boldsymbol{\eta})$  be the negative log likelihood written as a function of  $\boldsymbol{\eta} = (\eta_1, \dots, \eta_n)^\top$ . For  $\boldsymbol{\eta}_k$  in  $\mathcal{C}$ , let

$$\psi_k(\boldsymbol{\eta}) = \ell(\boldsymbol{\eta}_k) + \nabla \ell(\boldsymbol{\eta}_k)^\top (\boldsymbol{\eta} - \boldsymbol{\eta}_k) + \frac{1}{2} (\boldsymbol{\eta} - \boldsymbol{\eta}_k)^\top \mathbf{Q}_k (\boldsymbol{\eta} - \boldsymbol{\eta}_k), \quad (4)$$

where  $\nabla \ell(\boldsymbol{\eta}_k)$  is the gradient vector and  $\mathbf{Q}_k$  is the Hessian matrix for  $\ell(\boldsymbol{\eta})$ , both evaluated at  $\boldsymbol{\eta}_k$ . The iteratively re-weighted algorithm is:

1. Choose a valid starting  $\boldsymbol{\eta}_0$ , and set  $k = 0$ .
2. Given  $\boldsymbol{\eta}_k$ , minimize  $\psi_k(\boldsymbol{\eta})$  over  $\mathcal{C}$  defined by the model. Then  $\boldsymbol{\eta}_{k+1}$  minimizes  $\ell(\boldsymbol{\eta})$  over the line segment connecting the minimizer of  $\psi_k(\boldsymbol{\eta})$  and  $\boldsymbol{\eta}_k$ .
3. Set  $k = k + 1$  and repeat step 2, stopping when a convergence criterion is met.

At step 2, `coneB` is used. At each iteration of the algorithm, the vector  $\boldsymbol{\mu}_k$  is computed where  $\mu_{ki} = g^{-1}(\eta_{ki})$ . If the Hessian matrix is positive definite for all  $\boldsymbol{\eta}$  then the negative log-likelihood function is strictly convex and  $\boldsymbol{\mu}_k$  is guaranteed to converge to the MLE  $\hat{\boldsymbol{\mu}}_k$  (the proof is similar to that in [Meyer and Woodroffe \(2004\)](#), Theorem 1).

## 1.3. Two-dimensional monotone regression

For two-dimensional isotonic regression without additivity assumptions, the “warped-plane spline” (WPS) of [Meyer \(2016b\)](#) is implemented in `cgam` using the function `wps`. The least-squares model has the form

$$y_i = f(x_{1i}, x_{2i}) + \mathbf{z}_i^\top \boldsymbol{\alpha} + \sigma \varepsilon_i, \text{ for } i = 1, \dots, n, \quad (5)$$

where  $\boldsymbol{\alpha} \in \mathbb{R}^p$ ,  $\mathbf{z}_1, \dots, \mathbf{z}_n \in \mathbb{R}^p$  contain values of parametrically modeled covariates, and the  $\varepsilon_i$ ’s are mean-zero random errors. We know *a priori* that  $f$  is continuous and monotone in

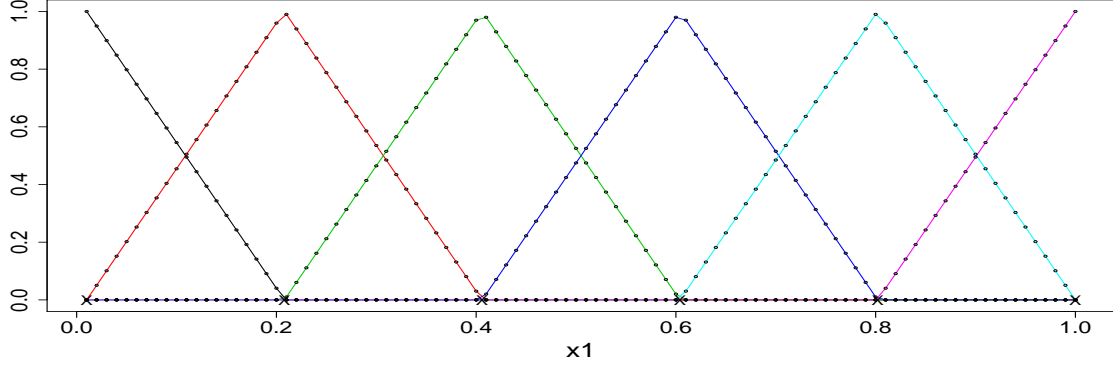


Figure 2: A set of linear basis functions for a predictor in two-dimensional isotonic regression with  $n = 100$  observations with values marked as dots and knots marked as “X”.

both dimensions; that is, for fixed  $x_1$ , if  $x_{2a} \leq x_{2b}$ , then  $f(x_1, x_{2a}) \leq f(x_1, x_{2b})$ , and similarly for fixed  $x_2$ ,  $f$  is non-decreasing in  $x_1$ . For linear spline basis functions defined in  $x_1$  and  $x_2$ , the basis functions for the necessary and sufficient constraints are straight-forward and the fitted surface can be described as a continuous piece-wise warped plane.

Given predictor values  $x_{1i}$ ,  $i = 1, \dots, n$ , we define knots  $t_{1,1} < \dots < t_{1,k_1}$ , where  $t_{1,1} \leq \min(\mathbf{x}_1)$  and  $t_{1,k_1} \geq \max(\mathbf{x}_1)$  are defined by evaluating the basis functions at the design points, that is,  $\delta_{1,l_1,i} = \delta_{1,l_1}(x_{1,i})$ ,  $l_1 = 1, \dots, k_1$ . These basis functions span the space of continuous piece-wise linear functions with given knots, and if we replace  $\delta_{1,1}$  with the constant function  $\delta_0(\mathbf{x}) = \mathbf{1}$ , then  $\delta_0, \delta_{1,2}, \dots, \delta_{1,k_1}$  span the same space. Similarly, spline basis functions  $\delta_{2,1}, \dots, \delta_{2,k_2}$  can be defined with knots  $t_{2,1} < \dots < t_{2,k_2}$ , where  $t_{2,1} \leq \min(\mathbf{x}_2)$  and  $t_{2,k_2} \geq \max(\mathbf{x}_2)$ . An example of a set of basis functions is in Figure 2. Let the  $n \times (k_1 - 1)$  matrix  $\mathbf{B}_1$  have as columns  $\delta_{1,2}, \dots, \delta_{1,k_1}$ , and let the  $n \times (k_2 - 1)$  matrix  $\mathbf{B}_2$  have as columns  $\delta_{2,2}, \dots, \delta_{2,k_2}$ . Finally let the  $n \times (k_1 - 1)(k_2 - 1)$  matrix  $\mathbf{B}_{12}$  contain the products of basis vectors, so that column  $(l_1 - 2)(k_1 - 1) + l_2 - 1$  of  $\mathbf{B}_{12}$  is the element-wise product of  $\delta_{1,l_1}$  and  $\delta_{2,l_2}$ , for  $l_1 = 2, \dots, k_1$  and  $l_2 = 2, \dots, k_2$ . The columns of  $\mathbf{B}_1$ ,  $\mathbf{B}_2$ , and  $\mathbf{B}_{12}$ , together with the vector  $\mathbf{1}$ , form a linearly independent set if  $n \geq k_1 k_2$  and there are no “empty cells”.

Let  $\theta_{ij} = f(x_{1i}, x_{2j})$  be the values of the regression function evaluated at the design points. This is approximated by  $\beta_0 \mathbf{1} + \mathbf{B}_1 \beta_1 + \mathbf{B}_2 \beta_2 + \mathbf{B}_{12} \beta_3 = \mathbf{B} \beta$ , where  $\mathbf{B} = [\mathbf{1} | \mathbf{B}_1 | \mathbf{B}_2 | \mathbf{B}_{12}]$  and  $\beta^\top = [\beta_0 | \beta_1^\top | \beta_2^\top | \beta_3^\top]$ . A constraint matrix  $\mathbf{A}$  will give the necessary and sufficient conditions, which is stated in Theorem 1 of Meyer (2016b), for monotonicity of the spline basis functions in both predictors if and only if  $\mathbf{A} \beta \geq \mathbf{0}$ . Here,  $\mathbf{A}$  is a  $k \times (k_1 k_2)$  matrix where  $k = 2k_1 k_2 - k_1 - k_2$ . The constrained least-squares solution is a projection of  $\mathbf{y}$  onto the cone

$$\mathcal{C} = \{\mu \in \mathbb{R}^n : \mu = \mathbf{B} \beta + \mathbf{Z} \alpha; \mathbf{A} \beta \geq \mathbf{0}\}; \quad (6)$$

the routine `coneA` in the R package `coneproj` Liao and Meyer (2014) is used.

Penalized warped-plane regression is also included in this package. To obtain smoother fits and to side-step the problem of knot choices, we can use “large”  $k_1$  and  $k_2$ , and penalize the changes in slopes of the regression surface, which is a warped plane over each knot rectangle whose slopes can change abruptly from one rectangle to the next. An additional advantage of penalization is that empty cells are allowed. The sum of the squared differences in slopes is used as the penalty term, where  $\lambda > 0$  is a penalty parameter and it will control the

constrained “effective degrees of freedom” ( $\text{edf}_{\lambda}$ ) of the fit. The standard generalized cross validation (GCV) defined in Chapter 5 of [Ruppert, Wand, and Carroll \(2003\)](#) can be used to select a penalty parameter:

$$\text{GCV}(\lambda) = \frac{\sum_{i=1}^n [y_i - \hat{\mu}_{\lambda,i}]^2}{(1 - \text{edf}_{\lambda}/n)^2}. \quad (7)$$

A range of values of  $\lambda$  can be tried and the GCV choice of  $\lambda$  minimizes the criterion. (See [Meyer \(2016b\)](#) for more details.)

Inference methods are discussed in Section 2, main routines in this package are discussed in Section 3, and the usage of this package is exemplified with simulated and real data sets in Section 4. In Section 5, we compare this package with some competitor packages in terms of utility and speed.

## 2. Inference methods

For inference regarding  $\boldsymbol{\alpha}$ , an approximately normal distribution can be derived for  $\sqrt{n}(\hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha})$ . Proposition 4 of [Meyer \(1999\)](#) says that there is a subset of edges, indexed by  $J \subseteq \{1, \dots, m\}$  such that the projection of  $\mathbf{y}$  onto  $\mathcal{C}$  coincides with projection of  $\mathbf{y}$  onto the linear space spanned by  $\mathbf{e}_j$ ,  $j \in J$ , and the basis vectors for  $\mathcal{L}$ . Suppose  $\mathbf{X}_0$  is a matrix so that the columns of  $\mathbf{X}_0$  and  $\mathbf{Z}$  span the linear space  $\mathcal{L}$ . Then if  $\mathbf{P}_J$  is the projection matrix for the space spanned by  $\mathbf{e}_j$ ,  $j \in J$  and the columns of  $\mathbf{X}_0$ , we can write

$$\hat{\boldsymbol{\alpha}} = [\mathbf{Z}^\top (\mathbf{I} - \mathbf{P}_J) \mathbf{Z}]^{-1} \mathbf{Z}^\top (\mathbf{I} - \mathbf{P}_J) \mathbf{y}.$$

Under mild regularity conditions,  $\hat{\boldsymbol{\alpha}}$  is approximately normal with mean zero and covariance  $[\mathbf{Z}^\top (\mathbf{I} - \mathbf{P}_J) \mathbf{Z}]^{-1} \sigma^2$ . We estimate  $\sigma^2$  as

$$\hat{\sigma}^2 = \frac{SSR}{n - d_0 - cEDF},$$

where  $SSR$  is the sum of squared residuals,  $d_0$  is the dimension of the linear space  $\mathcal{L}$ , and  $EDF$  is the effective degrees of freedom, that is, the cardinality of  $J$  plus the dimension of  $\mathcal{L}$ . The constant  $c$  is between 1 and 2; [Meyer and Woodroffe \(2000\)](#) showed this multiplier is appropriate for cone regression. That paper gave evidence that  $c = 1.5$  is appropriate for unsmoothed isotonic regression; simulations suggest that for constrained splines, a smaller value gives better estimates of  $\sigma^2$ . In the `cgam` routine and the `wps` routine, the default is  $c = 1.2$ , but the user can specify  $c \in [1, 2]$ .

These results are used to construct approximate  $t$  and  $F$  tests for  $\boldsymbol{\alpha}$ ; specifically,  $\hat{\boldsymbol{\alpha}}$  is taken to be approximately normal with mean  $\boldsymbol{\alpha}$  and covariance  $[\mathbf{Z}^\top (\mathbf{I} - \mathbf{P}_J) \mathbf{Z}]^{-1} \hat{\sigma}^2$ . See [Meyer \(2016a\)](#) and [Meyer \(2016b\)](#) for detailed conditions under which this is a good approximation. The cone information criterion (CIC) proposed in [Meyer \(2013b\)](#) for the least-squares model is generalized to

$$CIC = -\frac{2}{n} \log(L) + \log \left\{ \frac{2[E_0(EDF) + d_0]}{n - d_0 - 1.5E_0(EDF)} + 1 \right\},$$

where  $L$  is the likelihood maximized over the cone  $\mathcal{C}$ ,  $d_0$  is the dimension of the linear space  $\mathcal{L}$ , and  $E_0(EDF)$  is the null expected dimension of the face of  $\mathcal{C}$  on which the projection lands.

To compute  $E_0(EDF)$ , we simulate from (1) and (2) with  $f_\ell \equiv 0$  for  $\ell = 1, \dots, L$ . In this way we get the expected degrees of freedom for the constrained model, in the case where the  $f_\ell$  do not contribute to the expected response. This is appropriate for model selection, as the observed  $EDF$  tends to be larger when the predictors are related to the response. See Meyer (2013b) for more details about using the CIC for model selection.

This criterion is the estimated predictive squared error, similar to the AIC, and is specially derived for cone projection problems. If the constraints are not known *a priori*, the CIC model selection procedure may be used to select not only the variables in a model of the form (2), but also the nature of their relationships with the response.

### 3. Main routines in this package

The function `cgam` is the main routine which implements the constrained generalized additive regression. For a non-parametrically modeled effect, a shape restriction can be imposed on the predictor function component with optional smoothing, or a partial ordering can be imposed. An arbitrary number of parametrically modeled covariates may be included. The user can also choose an unconstrained smooth fit for one or more of the  $f_\ell$ , which is simply the least-squares estimator using the set of cubic spline basis functions created for convex constraints. The specification of the model in `cgam` uses one or more of nineteen symbol functions to specify the shape, ordering, and smoothness of each  $f_\ell$ .

#### 3.1. The symbolic functions that specify the form of $f_\ell$

To specify an effect that is increasing with a predictor  $x$  without smoothing, the function `incr(x)` is used in the statement of the `cgam` routine. Other functions for unsmoothed effects are `decr`, `conv`, `conc`, `incr.conv`, `incr.conc`, `decr.conv`, `decr.conc`, `tree`, and `umbrella`.

For smooth estimates of  $f_\ell$ , the following functions may be used: `s.incr`, `s.decr`, `s.conv`, `s.conc`, `s.incr.conv`, `s.incr.conc`, `s.decr.conv`, and `s.decr.conc`. For fitting an unconstrained smooth effect, `s(x)` may be used. Each of these nineteen functions implements a routine to create the appropriate set of basis functions. The smoothed versions have options for number and spacing of knots. For example, `s.decr(x, numknots = 10, space = "Q")` will create I-spline basis functions with ten knots at equal quantiles of the observed  $x$  values. The default is `space = "E"` which provides equal spacing. For a data set of  $n$  observations, the number of knots has a default of order  $n^{1/9}$  ( $n^{1/7}$ ) when C-spline (I-spline) basis functions are used.

#### 3.2. Basic usage of the main routine: `cgam`

In the `cgam` routine, the specification of the model can be one of the three exponential families: Gaussian, Poisson and binomial. The symbolic functions are used to specify how the predictors are related to the response. For example,

```
R> fit <- cgam(y ~ s.incr.conv(x1) + s(x2, numknots = 5), family = gaussian())
```

specifies that the response  $y$  is from the Gaussian distribution, and  $E(y)$  is smoothly increasing and concave in  $x_1$ , while component function for  $x_2$  is smooth but unconstrained, with five equally spaced knots.



The user can also specify the parameter `nsim` to simulate the CIC value of the model. Such simulation can be time-consuming. The default is `nsim = 100`. For example, we can write

```
R> fit <- cgam(y ~ s.incr.conv(x1, numknots = 10, space = "Q")
+ s(x2, numknots = 10, space = "Q"), family = gaussian(), nsim = 1000)
```

For a `cgam` fit, the main values returned are the estimated systematic component  $\hat{\eta}$  and the estimated mean value  $\hat{\mu}$ , obtained by transforming  $\hat{\eta}$  by the inverse of the link function. The CIC value will also be returned if the user chooses to simulate it.

The routine `summary` provides the estimates, the standard errors, and approximate  $t$  values and  $p$  values for the linear terms. A `summary` table also includes the deviance for the null model of a `cgam` fit, i.e., the model only containing the constant vector and the residual deviance of a `cgam` fit.

### 3.3. Basic usage of the main routine: `wps`

The `wps` routine implements two-dimensional isotonic regression using warped-plane splines discussed in Meyer (2016b). Parametrically-modeled covariates can be incorporated in the regression. Three symbolic subroutines are used in a `wps` formula to specify the relationship between the mean surface and the two predictors to be one of the three: “doubly-decreasing”, “doubly-increasing”, and “decreasing-increasing”. To be specific, a basic `wps` formula will look like

```
R> fit.dd <- wps(y ~ dd(x1, x2))
```

The argument `dd` specifies that  $E(\mathbf{y})$  is decreasing in both  $\mathbf{x}_1$  and  $\mathbf{x}_2$ . Similarly, `ii` specifies that the relationship is doubly-increasing, and `di` specifies that  $E(\mathbf{y})$  is decreasing in  $\mathbf{x}_1$  and increasing in  $\mathbf{x}_2$ . The knots vector for each predictor can be defined similarly as a `cgam` fit.

The user can also choose to use a penalized version by providing a “large” number of knots for  $\mathbf{x}_1$  and  $\mathbf{x}_2$  with a penalty term. For example, we want to use ten equally spaced knots for each predictor with a penalty term to be .1 in a “doubly-decreasing” formula. Then we can write

```
R> fit.dd <- wps(y ~ dd(x1, x2, numknots = c(10, 10), space = c("E", "E")),
+ pen = .1)
```

For a `wps` fit, the main values returned are the estimated mean value  $\hat{\mu}$  and the constrained effective degrees of freedom (`edfc`). The generalized cross validation value (`GCV`) for the constrained fit is also returned, which could be used to choose the penalty parameter  $\lambda$ . The `summary` routine works the same way as for a `cgam` fit when there is any parametrically modeled covariate.

### 3.4. Basic usage of the graphical routine: `plotpersp`

This routine is an extension of the generic R graphics routine `persp`. For a `cgam` object, which has at least two non-parametrically modeled predictors, this routine will make a three-dimensional plot of the fit with a set of two non-parametrically modeled predictors in the formula, which will be marked as the `x` and `y` labs in the plot. If there are more than two



non-parametrically modeled predictors, any other such predictor will be evaluated at the largest value which is smaller than or equal to its median value. This routine also works for a **wps** object, which only has two isotonically modeled predictors. For a **cgam** fit, the **z** lab represents the estimated regression surface of the mean or the systematic component according to the user's choice, and for a **wps** fit, the **z** lab represents the constrained or the unconstrained regression surface of the mean according to the user's choice. If there is any categorical covariate in a **cgam** or **wps** model and if the user specifies the argument **categ** to be a character representing a categorical covariate in the formula, then a three-dimensional plot with multiple parallel surfaces, which represent the levels of the categorical covariate in an ascending order, will be created; otherwise, a three-dimensional plot with only one surface will be created. Each level of a categorical covariate will be evaluated at its mode.

The basic form of this routine is defined as

```
R> plotpersp(object,...)
```

The argument **object** represents an object of the **wps** class, or an object of the **cgam** class with at least two non-parametrically modeled predictors. When there are more than two non-parametrically modeled predictors in a **cgam** formula, the user may choose to write

```
R> plotpersp(object, x1, x2,...)
```

The arguments **x1** and **x2** represent two non-parametrically modeled predictors in the model. If the user omits the two arguments, the first two non-parametrically modeled predictors in the formula will be used.

## 4. User guide

We demonstrate the main routines using simulated data sets and real data sets in detailed examples. For a more complete explanation and demonstration of each routine, see the official reference manual of this package at <http://CRAN.R-project.org/package=cgam>. This package depends on the package **coneproj**, which conducts the cone projection algorithm. (See [Liao and Meyer \(2014\)](#) for more details.)

```
R> install.packages("cgam")
R> library("cgam")
Loading required package: coneproj
```

### 4.1. Fitting constrained surface to “Rubber”

The “Rubber” data set in the package **MASS** has 30 observations and three variables relevant to accelerated testing of tyre rubber, i.e., “loss” (abrasion loss), “hard” (hardness), and “tens” (tensile strength). Assuming that “loss” is decreasing in both “hard” and “tens”, the effects are additive, and the response is Gaussian, we can model the relationship as following:

```
R> library("MASS")
R> data("Rubber")
```

```
R> loss <- Rubber$loss
R> hard <- Rubber$hard
R> tens <- Rubber$tens
R> set.seed(123)
R> fit.decr <- cgam(loss ~ decr(hard) + decr(tens))
```

Alternatively, we can model the relationship using *I*-spline basis functions:

```
R> set.seed(123)
R> fit.s.decr <- cgam(loss ~ s.decr(hard) + s.decr(tens))
```

For a spline-based fit without constraints:

```
R> set.seed(123)
R> fit.s <- cgam(loss ~ s(hard) + s(tens))
```

For each fit, we use the default that `nsim = 100` to get the CIC parameter. According to the CIC value of each fit, the non-smooth fit is better than another two smooth fits for this data set.

```
R> fit.decr$cic
[1] 8.945583
R> fit.s.decr$cic
[1] 10.05648
R> fit.s$cic
[1] 10.16878
```

We can call the routine `plotpersp` to make a 3D plot of the estimated mean surface based on `fit.decr`, `fit.s.decr`, and `fit.s`, which is shown in Figure 3.

```
R> par(mfrow = c(1, 3))
R> plotpersp(fit.decr, hard, tens, th = 120, main = "(a)")
R> plotpersp(fit.s.decr, hard, tens, th = 120, main = "(b)")
R> plotpersp(fit.s, hard, tens, th = 120, main = "(c)")
```

## 4.2. Fitting parallel surfaces to “plasma”

The “plasma” data set, available at [http://axon.cs.byu.edu/data/statlib/numeric/plasma\\_retinol.arff](http://axon.cs.byu.edu/data/statlib/numeric/plasma_retinol.arff), contains 314 observations of blood plasma beta carotene measurements along with several covariates. High levels of blood plasma beta carotene are believed to be protective against cancer, and it is of interest to determine the relationships with covariates. Here we use the logarithm of “plasma” level as the response, and choose “bmi”, the logarithm of “dietfat”, “cholest”, “fiber”, “betacarot” and “retinol” as shape-restricted predictors. In addition, we include “smoke” and “vituse” as categorical covariates.

```
R> set.seed(123)
R> fit <- cgam(logplasma ~ s.decr(bmi) + s.decr(logdietfat) + s.decr(cholest)
+ + s.incr(fiber) + s.incr(betacarot) + s.incr(retinol) + factor(smoke)
+ + factor(vituse), data = plasma)
```

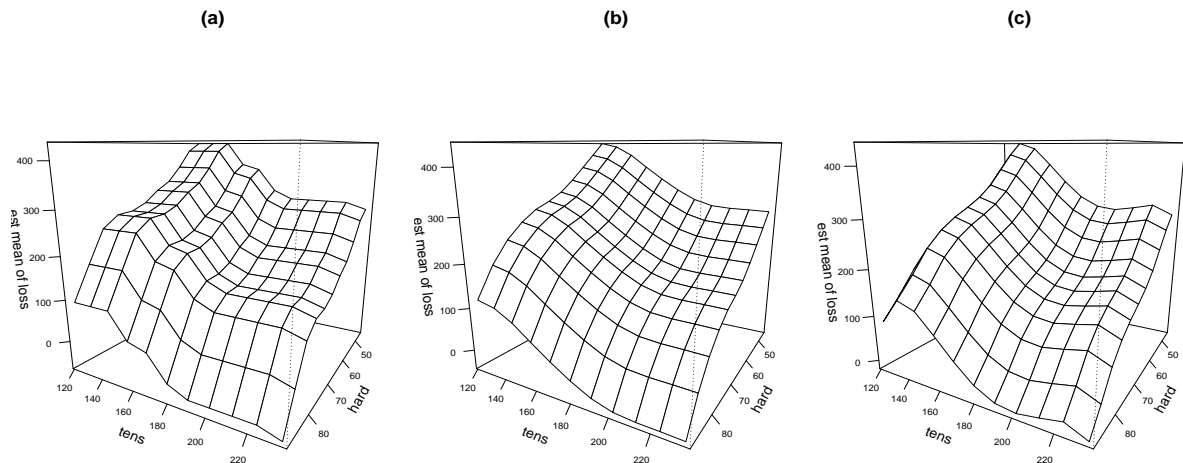


Figure 3: Demonstration of constrained regression using the `cgam` routine with the “Rubber” data set. (a) the estimated surface is decreasing in both predictors without smoothing. (b) the estimated surface is smooth and decreasing in both predictors. (c) the estimated surface is smooth in both predictors without shape constraint.

We can call `summary` to check the estimate, the standard error, the approximate  $t$  value and the  $p$  value for the coefficient of the categorical covariates. The CIC value is also simulated and returned.

```
R> summary(fit)
```

Call:

```
cgam(formula = logplasma ~ s.decr(bmi) + s.decr(logdietfat) + s.decr(cholest)
      + s.incr(fiber) + s.incr(betacarot) + s.incr(retinol) + factor(smoke)
      + factor(vituse), data = plasma)
```

Coefficients:

	Estimate	StdErr	t.value	p.value
(Intercept)	4.8398	0.1298	37.2895	<2e-16 ***
factor(smoke)2	0.2145	0.1279	1.6769	0.0947 .
factor(smoke)3	0.3232	0.1272	2.5402	0.0116 *
factor(vituse)2	-0.0936	0.1032	-0.9070	0.3652
factor(vituse)3	-0.2688	0.0948	-2.8345	0.0049 **

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 0.4164 )

Null deviance: 174.9801 on 313 degrees of freedom

Residual deviance: 128.6684 on 277.8 observed degrees of freedom

CIC: 4.9661

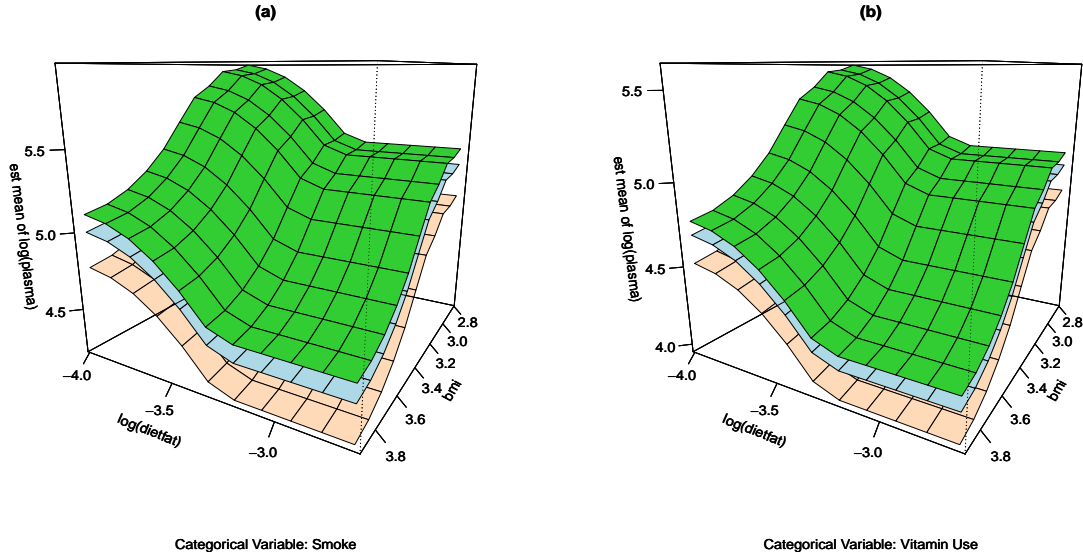


Figure 4: Demonstration of constrained regression using the `cgam` function with the “plasma” data set. (a) parallel surfaces representing the effects of three levels of “smoke” in an ascending order. (b) parallel surfaces representing the effects of three levels of “vituse” in an ascending order.

Again, we use `plotpersp` to show the estimated mean surface based on the fit in Figure 4, where `xlab` is “bmi”, `ylob` is the logarithm of “dietfat”, and the effects of the levels of each categorical covariate are shown in an ascending order. Other shape-restricted predictors are evaluated at the median value.

### 4.3. Partial-Ordering examples: tree-ordering and umbrella-ordering

We simulate a data set as a tree-ordering example such that  $\mathbf{x}$  is a categorical variable with five levels:  $x_1 = 0$  (placebo),  $x_2 = 1$ ,  $x_3 = 2$ ,  $x_4 = 3$  and  $x_5 = 4$ . Each level has 20 observations. We also include a categorical covariate  $\mathbf{z}$  with two levels “a” and “b”, which could be a treatment variable people are concerned about, in the model such that when  $\mathbf{x}$  is fixed, the mean response is one unit larger if  $\mathbf{z}$  is “a”. We use `cgam` to estimate the effect for each level given  $\mathbf{z}$ . The fit is in Figure 5(a).

```
R> n <- 100
R> x <- rep(0:4, each = 20)
R> z <- rep(c("a", "b"), 50)
R> y <- x + I(z == "a") + rnorm(n, 1)
R> fit.tree <- cgam(y ~ tree(x) + factor(z))
```

The estimated effect of  $\mathbf{z}$  can be checked by `summary`.

```
R> summary(fit.tree)
Coefficients:
              Estimate StdErr t.value    p.value
(Intercept)   2.1617   0.2292   9.4324 < 2.2e-16 ***
factor(z)b    -1.0402   0.1871  -5.5588 < 2.2e-16 ***
```

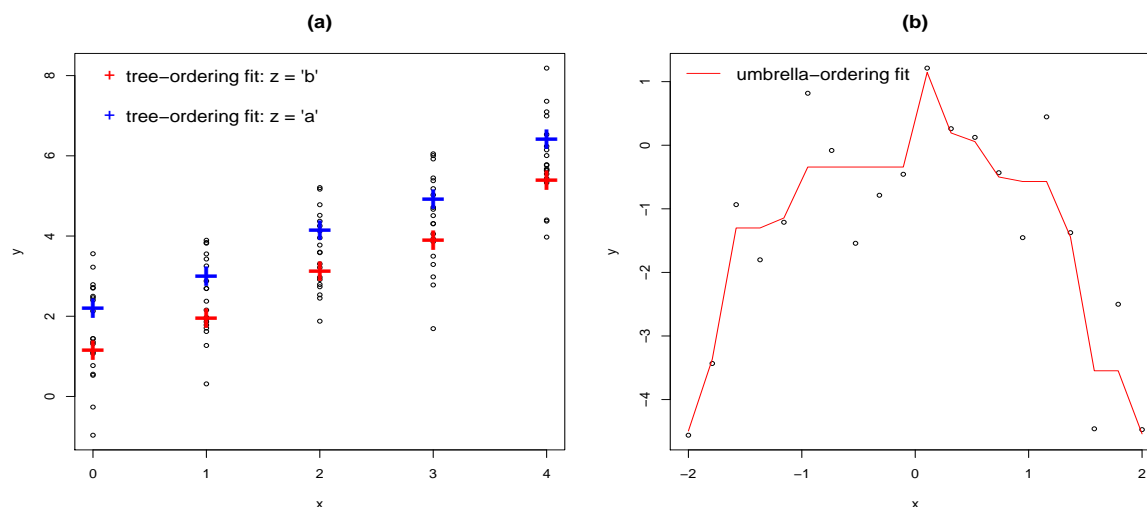


Figure 5: Demonstration of constrained regression using the `cgam` function with a partial ordering constraint. (a) a tree-ordering fit with a categorical covariate  $z$ . (b) an umbrella-ordering fit.

---

*Signif. codes:* 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

For an umbrella-ordering example, we simulate a data set such that  $x_0 = 0$  (mode) and for  $x_1, x_2 \geq x_0$ , the estimated mean curve is decreasing, while for  $x_1, x_2 \leq x_0$ , it is increasing. The fit is in Figure 5(b).

```
R> n <- 20
R> x <- seq(-2, 2, length = n)
R> y <- -x^2 + rnorm(n)
R> fit.umb <- cgam(y ~ umbrella(x))
```

#### 4.4. Binomial response example

We use the “kyphosis” data set with 81 observations in the **gam** package to show that how `cgam` works given a binomial response. In this example, we treat the variable “Kyphosis” as the response which is binary, and model the log-odds of “Kyphosis” as concave in “Age” (age of child in months), increasing-concave in “Number” (number of vertebra involved in the operation), and decreasing-concave in “Start” (start level of the operation). The non-smooth fit and the smooth fit are shown in Figure 6 by `plotpersp`.

```
R> library("gam")
R> data("kyphosis")
R> fit <- cgam(Kyphosis ~ conc(Age) + incr.conc(Number) + decr.conc(Start),
+ family = binomial(), data = kyphosis)
R> fit.s <- cgam(Kyphosis ~ s.conc(Age) + s.incr.conc(Number)
+ + s.decr.conc(Start), family = binomial(), data = kyphosis)
```

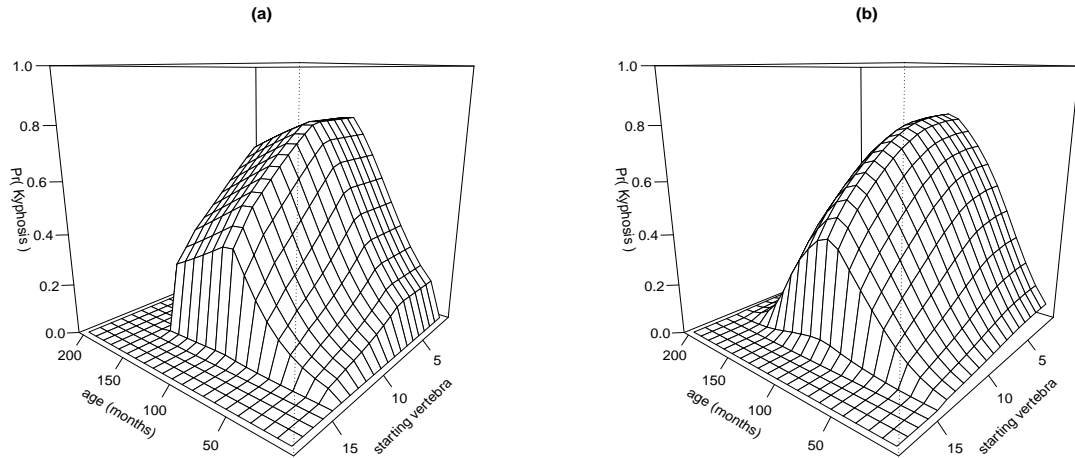


Figure 6: Demonstration of constrained regression using the `cgam` function with the “kyphosis” data set. The surface represents the estimated probability of the response Kyphosis to be present. (a) surface without smoothing (b) smooth surface.

Next, we consider the “bpd” data set in the **SemiPar** package. It has 223 observations with two variables: birth weight of babies and BPD, which is a binary variable indicating the presence of bronchopulmonary dysplasia. It is known that bronchopulmonary dysplasia is more often found in babies with low birth weight, and we can model the relationship between the probability of bronchopulmonary dysplasia and birth weight as smoothly decreasing. The fit is shown in Figure 7. We also include the linear and quadratic log-odds fit in the plot as a comparison. The linear log-odds fit might overly simplify the underlying relationship, while the quadratic fit starts increasing at the end although it seems to be better than the linear fit.

```
R> library("SemiPar")
R> data("bpd")
R> BPD <- bpd$BPD
R> birthweight <- bpd$birthweight
R> fit.s.decr <- cgam(BPD ~ s.decr(birthweight, space = "Q"),
+ family = binomial())
```

#### 4.5. Poisson response example

Another data set is an attendance data set of 316 high school juniors from two urban high schools. We use the variable “daysabs” (days absent) as a Poisson response. The variables “math” and “langarts” are the standardized test scores for math and language arts. A categorical variable “male” is also included in this data set, which indicates the gender of a student. With *a priori* knowledge that “daysabs” is decreasing with respect to each continuous predictor, we can try modeling the relationship between “daysabs” and “math” and “langarts” as decreasing with “male” as a categorical covariate. First, we model the relationship with ordinal basis functions.

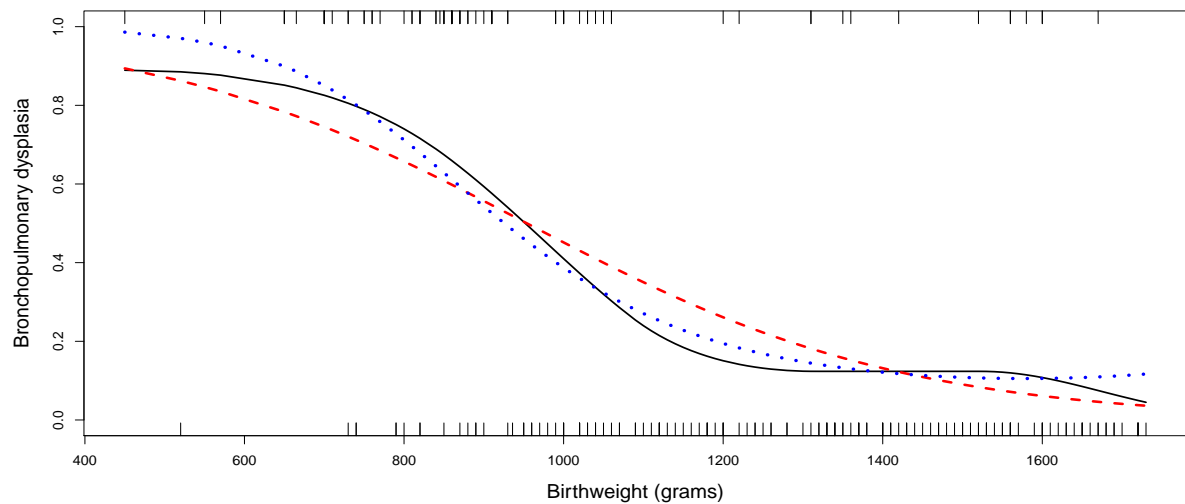


Figure 7: The estimated probability of bronchopulmonary dysplasia as a function of birth weight. The data are shown as tick marks at the presence (1) and the absence (0) of the condition. The solid curve is the smoothly decreasing fit, the dashed curve is the linear log-odds fit, and the dotted curve is the quadratic log-odds fit.

```
R> set.seed(123)
R> fit.cgam <- cgam(daysabs ~ decr(math) + decr(langarts) + factor(male),
+ family = poisson())
R> summary(fit.cgam)
Call:
cgam(formula = daysabs ~ decr(math) + decr(langarts) + factor(male),
      family = poisson())
```

Coefficients:

	Estimate	StdErr	z.value	p.value
(Intercept)	1.8715	0.0317	58.9843	< 2.2e-16 ***
factor(male)1	-0.4025	0.0496	-8.1155	< 2.2e-16 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 2409.8204 on 315 degrees of freedom

Residual deviance: 2100.6701 on 296 observed degrees of freedom

CIC: -9.7546

Next, we try modeling the relationship with smooth *I*-splines.

```
R> set.seed(123)
R> fit.cgam.s <- cgam(daysabs ~ s.decr(math) + s.decr(langarts) + factor(male),
+ family = poisson())
```



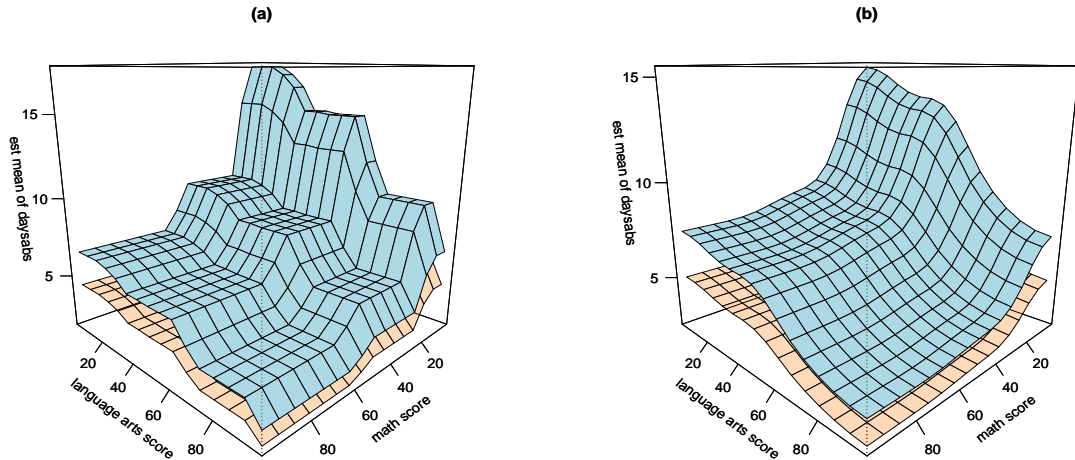


Figure 8: Demonstration of constrained regression using the `cgam` function with the “attendance” data set. (a) parallel surfaces representing the gender effect in an ascending order. (b) parallel surfaces representing the gender effect in an ascending order with smoothing.

```
R> summary(fit.cgam.s)
Call:
cgam(formula = daysabs ~ s.decr(math) + s.decr(langarts) + factor(male),
      family = poisson())

Coefficients:
                Estimate StdErr z.value  p.value
(Intercept)      1.8982  0.0309 61.4593 < 2.2e-16 ***
factor(male)1    -0.3988  0.0487 -8.1894 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 2409.8204 on 315 degrees of freedom
Residual deviance: 2201.237 on 304.4 observed degrees of freedom
CIC: -9.4567
```

According to the simulated CIC value of each fit, it is suggested that the non-smooth fit is better than the fit using smooth *I*-splines. Moreover, the gender effect is significant in both fits. The fits are shown in Figure 8.

#### 4.6. Fitting “doubly-decreasing” surface to “plasma”

We again use the “plasma” data set as an example to illustrate the routine `wps`, and now we assume that the logarithm of “plasma” is doubly-decreasing in “bmi” and the logarithm of “dietfat”, and the effects of the two predictors are not necessarily additive. We also include

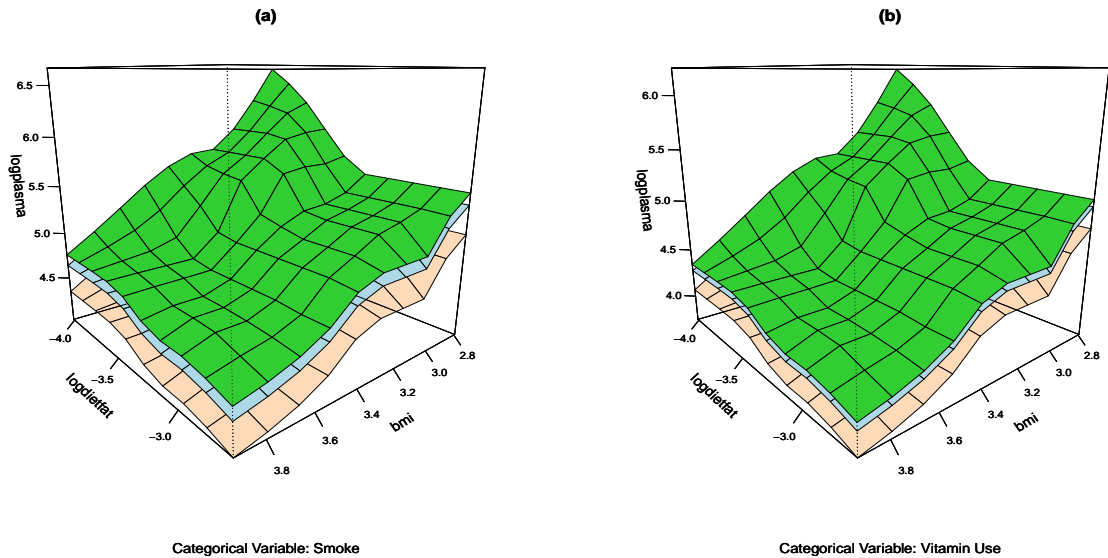


Figure 9: Demonstration of constrained regression using the `wps` routine with the “plasma” data set. In each plot, the estimated surface is constrained to be doubly-decreasing in both predictors without the assumption of additivity. Parallel surfaces representing the effects of three levels of (a) “smoke” and (b) “vituse,” in an ascending order.

“smoke” and “vituse” as two categorical covariates. We can model the relationship as following, and we choose 10 equally-spaced knots for each predictor with the penalty term to be .505, which is calculated inside `wps` and can be checked as an output:

```
R> fit <- wps(logplasma ~ dd(bmi, logdietfat) + factor(smoke) + factor(vituse),
+ data = plasma, pnt = TRUE)
R> fit$pen
[1] 0.5047263
R> summary(fit)
Call:
wps(formula = logplasma ~ dd(bmi, logdietfat) + factor(smoke) + factor(vituse),
    data = plasma, pnt = TRUE)
```

Coefficients:

	Estimate	StdErr	t.value	p.value
(Intercept)	4.0144	0.1248	32.1730	<2e-16 ***
factor(smoke)2	0.2988	0.1261	2.3689	0.0185 *
factor(smoke)3	0.4184	0.1228	3.4079	0.0007 ***
factor(vituse)2	-0.0667	0.1013	-0.6581	0.5110
factor(vituse)3	-0.2757	0.0935	-2.9490	0.0034 **

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

With 100 simulations, the CIC value is 5.02 for the doubly-decreasing model, and 4.97 for the additive model in Section 4.2; this is evidence that the additive model is adequate.

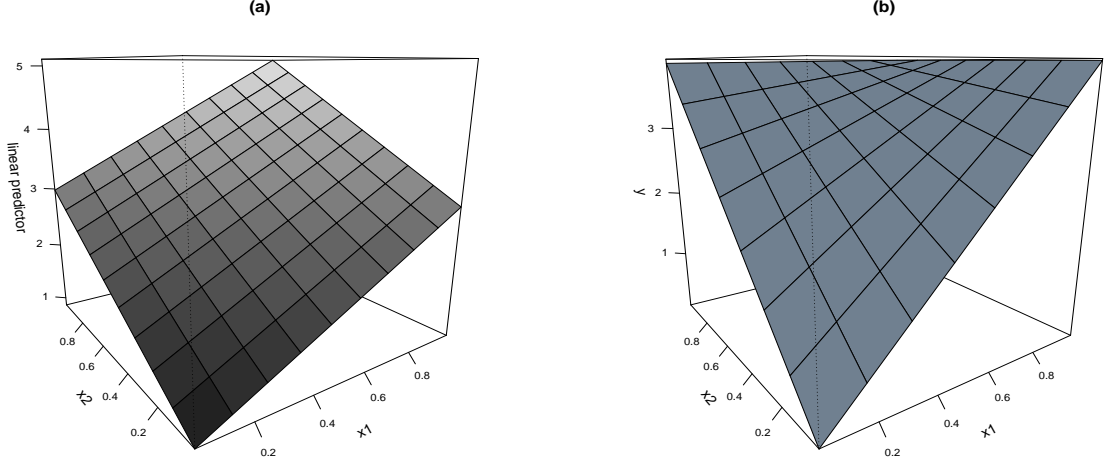


Figure 10: (a) estimated doubly-increasing regression surface using the `scam` routine. (b) estimated doubly-increasing regression surface using the `wps` routine.

## 5. Discussion and comparison to similar packages

The R package `scam` (shape constrained additive model, [Pya and Wood \(2015\)](#)) uses penalized splines to fit shape-constrained model components. The  $P$ -splines proposed in [Eilers and Marx \(1996\)](#) are used with coefficients subject to linear constraints. The shape options are similar to those in `cgam`, but the back-fitting method for estimation in `scam` is slower than the single cone projection used in `cgam`. To compare the speeds for smooth fitting of regression functions, we simulated 1000 datasets from a regression model with three predictors, and fit isotonic additive models using `cgam` and `scam`. Specifically,  $x_{1i}$ ,  $x_{2i}$ , and  $x_{3i}$ ,  $i = 1, \dots, n$ , were simulated independently and uniformly on  $[0, 1]$  along with independent standard normal errors  $\varepsilon_i$ ,  $i = 1, \dots, n$ , then,  $y_i = x_{1i} + x_{2i}^2 + x_{3i}^3 + \varepsilon_i$ . When  $n = 100$ , the time used by `cgam` and `scam` are about 17 and 611 seconds, respectively. When  $n = 500$ , the time used by `cgam` and `scam` are about 58 and 842 seconds. The speed comparisons were made on a laptop with a 2.16GHz dual-core Intel(R) Celeron(R) CPU.

The `scam` package also has a routine to fit a bivariate isotonic regression function without additivity. However, the constraints are sufficient but not necessary, and in fact severely over-constrain the fit. In particular, it can not fit surfaces that are doubly monotone, but whose rate of increase is decreasing in at least one of the dimensions. To demonstrate this, we simulated from the regression surface  $f(x) = 4(x_1 + x_2 - x_1x_2)$  which is increasing over the unit square. The sample size  $n = 50$ , each predictor is uniformly generated on the unit interval, and we choose not to add errors in this example. We use the default settings in `scam` and `cgam` to get doubly-increasing fits. An example of each method is shown in Figure 10 using the same data set.

The package `scar` (shape constrained additive regression, [Chen and Samworth \(2016\)](#)), provides the maximum likelihood estimator of the generalized additive regression with shape constraints, but without smoothing.

The packages `gam` and `mgcv` fit the generalized additive model, but without constraints.

The main routine **gam** in the **gam** package uses local regression or smoothing splines with the local scoring algorithm, which iteratively fits weighted additive models by back-fitting, while in the **mgcv** package, the main routine **gam** only uses penalized smoothing splines and smoothness selection by criteria such as GCV is part of its model fitting. Besides, this **gam** routine provides an option for modeling smooth interactions of any number of variables via scale invariant tensor product smooths.

The function specification of **cgam** is modeled after the popular **gam** function in the **mgcv** package with additional options for indicating shape. For example, a user may specify penalized smoothing in the **mgcv** package by

```
R> gam(y ~ s(x))
```

with no shape constraint, and choose spline-based regression with a shape constraint such as increasing by

```
R> cgam(y ~ s.incr(x))
```

in the **cgam** package.

## References

- Chen Y, Samworth RJ (2016). “Generalized Additive and Index Models with Shape Constraints.” *Journal of the Royal Statistical Society B*.
- Eilers PHC, Marx BD (1996). “Flexible Smoothing with B-Splines and Penalties (with discussion).” *Statistical Science*, **11**, 89–121.
- Hastie TJ, Tibshirani RJ (1990). *Generalized Additive Models*. Chapman and Hall, Washington, D. C.
- Liao X, Meyer MC (2014). “**coneproj**: An R Package for the Primal or Dual Cone Projections with Routines for Constrained Regression.” *Journal of Statistical Software*, **61**(12), 1–22. URL <http://www.jstatsoft.org/v61/i12/>.
- McCullagh P, Nelder JA (1989). *Generalized Linear Models*. 2 edition. Chapman and Hall, London.
- Meyer MC (1999). “An Extension of the Mixed Primal-Dual Bases Algorithm to the Case of More Constraints than Dimensions.” *Journal of Statistical Planning and Inference*, **81**, 13–31.
- Meyer MC (2008). “Inference Using Shape-Restricted Regression Splines.” *The Annals of Applied Statistics*, **2**(3), 1013–1033.
- Meyer MC (2013a). “Semi-Parametric Additive Constrained Regression.” *Journal of Non-parametric Statistics*, **25**(3), 715–743.
- Meyer MC (2013b). “A Simple New Algorithm for Quadratic Programming with Applications in Statistics.” *Communications in Statistics*, **42**(5), 1126–1139.

- Meyer MC (2016a). “Constrained Partial Linear Regression Splines.” *submitted*.
- Meyer MC (2016b). “Estimation and Inference for Smooth Regression Surfaces using Constrained Splines.” *submitted*.
- Meyer MC, Woodroffe M (2000). “On the Degrees of Freedom in Shape-Restricted Regression.” *Ann. Statist.*, **28**, 1083–1104.
- Meyer MC, Woodroffe M (2004). “Consistent Maximum Likelihood Estimation of a Unimodal Density using Shape Restrictions.” *Canadian Journal of Statistics*, **32**(1), 85–100.
- Pya N, Wood SN (2015). “Shape Constrained Additive Models.” *Statistics and Computing*, **25**, 543–559.
- Ruppert D, Wand MP, Carroll RJ (2003). *Semiparametric Regression*. Cambridge University Press, Cambridge, UK.

**Affiliation:**

Xiyue Liao  
Department of Statistics  
Colorado State University  
Fort Collins, Colorado 80523, United States of America  
E-mail: [xiyue@rams.colostate.edu](mailto:xiyue@rams.colostate.edu)

Mary C. Meyer  
Department of Statistics  
Colorado State University  
Fort Collins, Colorado 80523, United States of America  
E-mail: [meyer@stat.colostate.edu](mailto:meyer@stat.colostate.edu)