Journal of Statistical Software

May 2019, Volume 89, Issue 5. *doi: 10.18637/jss.v089.i05*

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 individual components for continuous predictors, and various types of orderings for the ordinal predictors. In addition, the user may specify parametrically modeled covariates. Two-way interactions between continuous variables, where the relationship with the response is constrained to be monotone, are modeled with "warped-plane splines." 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 (CIC) is provided and may be used to compare fits for combinations of variables and shapes. The graphical routine `plotpersp` will plot an estimated mean surface for a selected pair of predictors, given an object fitted with **cgam**. This package is available from the Comprehensive R Archive Network (CRAN) at `https://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. Overview and comparison with other packages

The generalized additive model is a useful and popular data analysis method. Several R (R Core Team 2019) packages provide estimation and inference for the effects of a set of predictors on a response variable, through specification of a link function. For example, suppose $y_i = 1$ if the ith observation is a success, and the probability of a success is believed to be a function of continuous predictors x_1 and x_2, a treatment variable x_3, and a nominal predictor z with two levels. The log-odds for the ith observation might be modeled as

$$\eta_i = f_1(x_{1i}) + f_2(x_{2i}) + \alpha_0 d_{0i} + \alpha_1 d_{1i} + \alpha_2 d_{2i} + \beta z_i,$$

where α_0, α_1, and α_2 represent effects for the placebo and two treatments, and β represents the effect of the nominal predictor. The `glm` function in R requires parametric specifications of the components, and does not allow ordered estimation with ordinal predictors. For example in the usage

```
fit <- glm(y ~ x1 + x2 + as.factor(tr) + z, family = binomial)
```

the functions f_1 and f_2 are specified as linear in their predictors.

A popular method for fitting such a model with only smoothness assumptions for f_1 and f_2 is provided with the `gam` function in the R package **mgcv** (Wood 2017, 2019), where the usage might be

```
fit <- gam(y ~ s(x1) + s(x2) + as.factor(tr) + z, family = binomial)
```

The package **cgam** (Meyer and Liao 2019) has similar usage, but provides shape and ordering options. Suppose that functional forms are not known for f_1 and f_2; instead we know that f_1 is smooth, increasing, and convex, while f_2 is smooth and decreasing. For example,

```
fit <- cgam(y ~ s.incr.conv(x1) + s.decr(x2) + tree(tr) + z,
  family = binomial)
```

will specify the shapes of f_1 and f_2, and impose a tree ordering on the treatment variable. This is appropriate if we can assume that the treatment effects are at least as large as the placebo effect.

The **cgam** package also has an option to fit a bivariate isotonic regression function without additivity assumptions. For example

```
fit <- cgam(y ~ s.incr.incr(x1, x2) + tree(tr) + z, family = binomial)
```

will fit a "warped-plane spline" that is increasing in both predictors, using linear constraints that are necessary and sufficient for monotonicity.

Two other R packages provide fits for constrained models. The package **scar** (**s**hape **c**onstrained **a**dditive **r**egression; Chen and Samworth 2016), provides the maximum likelihood estimator of the generalized additive regression with shape constraints, but without smoothing. In this case, the estimated predictor function η is a step function in the predictors that are constrained to be monotone, and piece-wise linear for components constrained to be convex.

The R package **scam** (**s**hape **c**onstrained **a**dditive **m**odel; Pya and Wood 2015) uses penalized splines to fit shape-constrained model components, but it does not include options for ordered effects for ordinal predictors. The P-splines proposed in Eilers and Marx (1996) are used with coefficients subject to linear constraints. The shape options for the smoothed terms are similar to those in **cgam**, but the penalized likelihood method for estimation uses back-fitting which is slower than the single cone projection used in **cgam**.

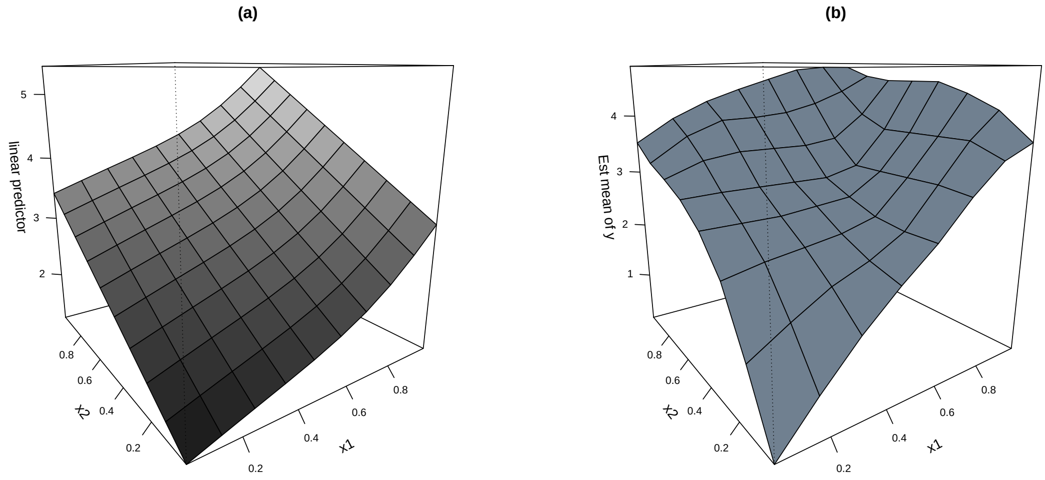

Figure 1: Estimated doubly-increasing regression surfaces using (a) the **scam** routine, and (b) the **cgam** routine, where the true surface is $f(x_1, x_2) = 4(x_1 + x_2 - x_1x_2)$.

The **scam** package will fit a bivariate isotonic regression function without additivity assumptions, using tensor product cubic splines. However, the constraints are sufficient but not necessary, and in fact severely over-constrain the fit. In particular, it cannot fit surfaces that are doubly monotone, but whose rate of increase in x_1 is decreasing in x_2 or vice-versa. 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 is $n = 50$, and each predictor is uniformly generated on the unit interval, and errors are i.i.d. normal with zero mean and unit variance. We use the default settings in **scam** and **cgam** to get doubly-increasing fits. An example of each method is shown in Figure 1 using the same data set.

2. Demonstrations of the routines in the cgam package

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 `https://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
```

2.1. Fitting a constrained surface to the `Rubber` data set

The `Rubber` data set in the package **MASS** (Venables and Ripley 2002) 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> data("Rubber", package = "MASS")
R> fit.decr <- cgam(loss ~ decr(hard) + decr(tens), family = gaussian,
+    data = Rubber)
```

Alternatively, we can model the relationship using splines:

```
R> fit.s.decr <- cgam(loss ~ s.decr(hard) + s.decr(tens), family = gaussian,
+    data = Rubber)
```

For a spline-based fit without constraints:

```
R> fit.s <- cgam(loss ~ s(hard) + s(tens), family = gaussian, data = Rubber)
```

For each fit, we use the default nsim = 100 to get the CIC parameter. According to the CIC value of each fit, the smooth and decreasing fit is better than the fit with only smoothness.

```
R> fit.s.decr$cic

[1] 10.04874

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 2.

```
R> par(mfrow = c(1, 3))
R> plotpersp(fit.decr, hard, tens, th = 120, main = "(a)", ngrid = 31)
R> plotpersp(fit.s.decr, hard, tens, th = 120, main = "(b)", ngrid = 31)
R> plotpersp(fit.s, hard, tens, th = 120, main = "(c)", ngrid = 31)
```

2.2. Fitting parallel surfaces to the plasma data set

The plasma data set in Nierenberg, Stukel, Baron, Dain, and Greenberg (1989) 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, betacaro and retinol as shape-restricted predictors. In addition, we include smoke and vituse as categorical covariates.

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

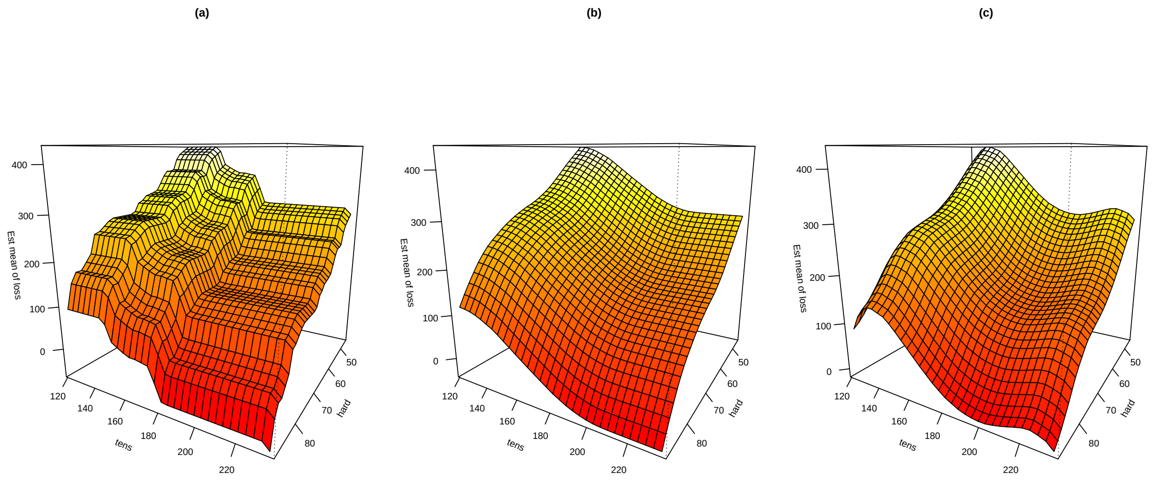

Figure 2: Demonstration of constrained regression using the **cgam** routine with the `Rubber` data set. (a) the estimated surface is decreasing in both predictors without smoothing. (b) the estimated surface is smooth and decreasing in both predictors. (c) the estimated surface is smooth in both predictors 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(betacaro) + s.incr(retinol) +
    factor(smoke) + factor(vituse), data = plasma)

Coefficients:
                 Estimate  StdErr t.value p.value
(Intercept)        4.8519  0.1270 38.2013  <2e-16 ***
factor(smoke)2     0.2063  0.1275  1.6182  0.1068
factor(smoke)3     0.3113  0.1261  2.4685  0.0142 *
factor(vituse)2   -0.0987  0.1003 -0.9835  0.3262
factor(vituse)3   -0.2739  0.0923 -2.9689  0.0032 **
---
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:  127.1359  on 280.4  observed degrees of freedom
...
CIC: 4.9671
```

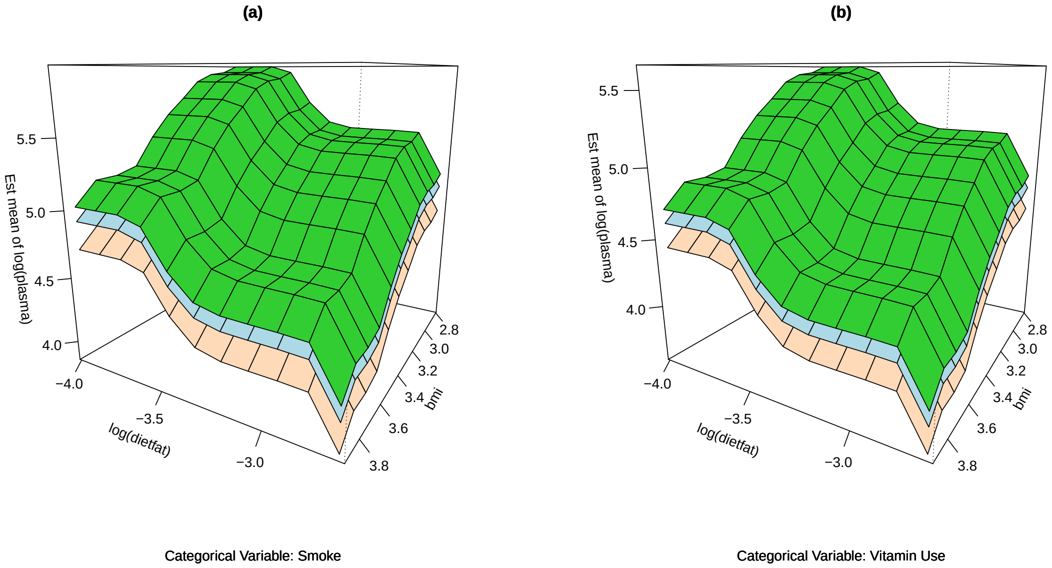

Figure 3: Demonstration of constrained regression using the **cgam** function with the `plasma` data set. (a) parallel surfaces representing the effects of three levels of `smoke` in an ascending order. (b) parallel surfaces representing the effects of three levels of `vituse` in an ascending order.

Again, we use `plotpersp` to show the estimated mean surface based on the fit in Figure 3, where `xlab` is `bmi`, `ylab` 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.

2.3. Partial-ordering examples: Tree-ordering and umbrella-ordering

We simulate a data set as a tree-ordering example such that $\boldsymbol{x}$ 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 $\boldsymbol{z}$ with two levels "a" and "b", which could be a treatment variable people are concerned about, in the model such that when $\boldsymbol{x}$ is fixed, the mean response is one unit larger if $\boldsymbol{z}$ is "a". We use **cgam** to estimate the effect for each level given $\boldsymbol{z}$. The fit is shown in Figure 4(a).

```
R> set.seed(123)
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 $\boldsymbol{z}$ can be checked by `summary`.

```
R> summary(fit.tree)
```

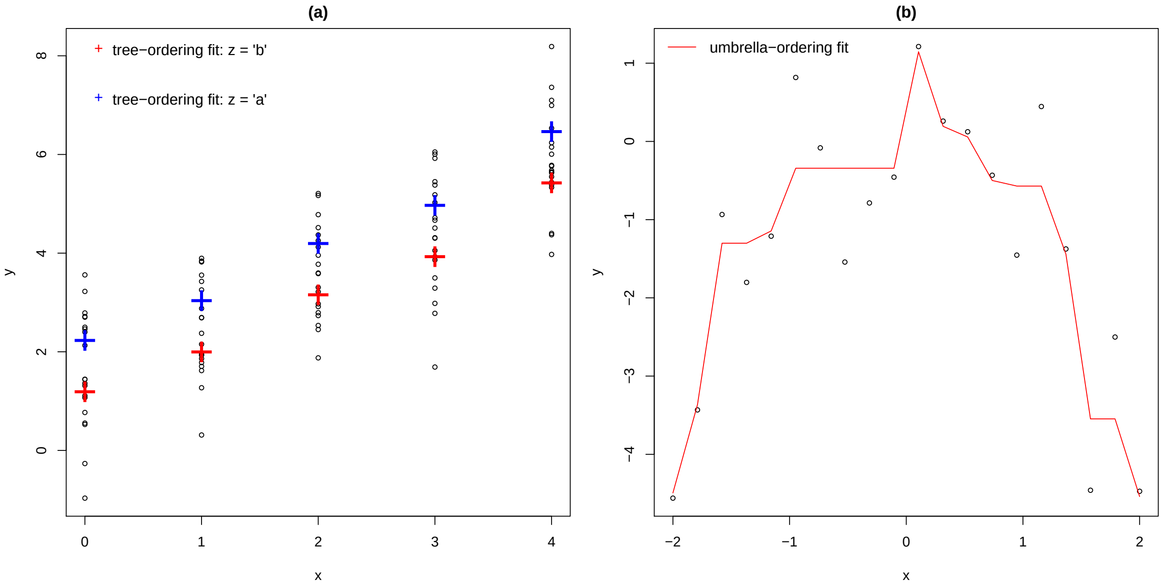

Figure 4: Demonstration of constrained regression using the **cgam** function with a partial ordering constraint. (a) a tree-ordering fit with a categorical covariate $\boldsymbol{z}$. (b) an umbrella-ordering fit.

```
Coefficients:
            Estimate  StdErr t.value   p.value
(Intercept)   2.1617  0.2289  9.4428 < 2.2e-16 ***
factor(z)b   -1.0402  0.1869 -5.5649 < 2.2e-16 ***
---
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 shown in Figure 4(b).

```
R> set.seed(123)
R> n <- 20
R> x <- seq(-2, 2, length = n)
R> y <- - x^2 + rnorm(n)
R> fit.umb <- cgam(y ~ umbrella(x))
```

2.4. Proportional odds model example

We use the “mental impairment” data set `mental` from Chapter 3 of Agresti (2010) as an example. The data set comes from a study of mental health for a random sample of 40 adult residents of Alachua County, Florida. Mental impairment is an ordinal response with 4 categories: well, mild symptom formation, moderate symptom formation, and impaired, which are recorded as 1, 2, 3, and 4. Life event index is a composite measure of the number and severity of important life events that occurred with the past three years, e.g., birth of a

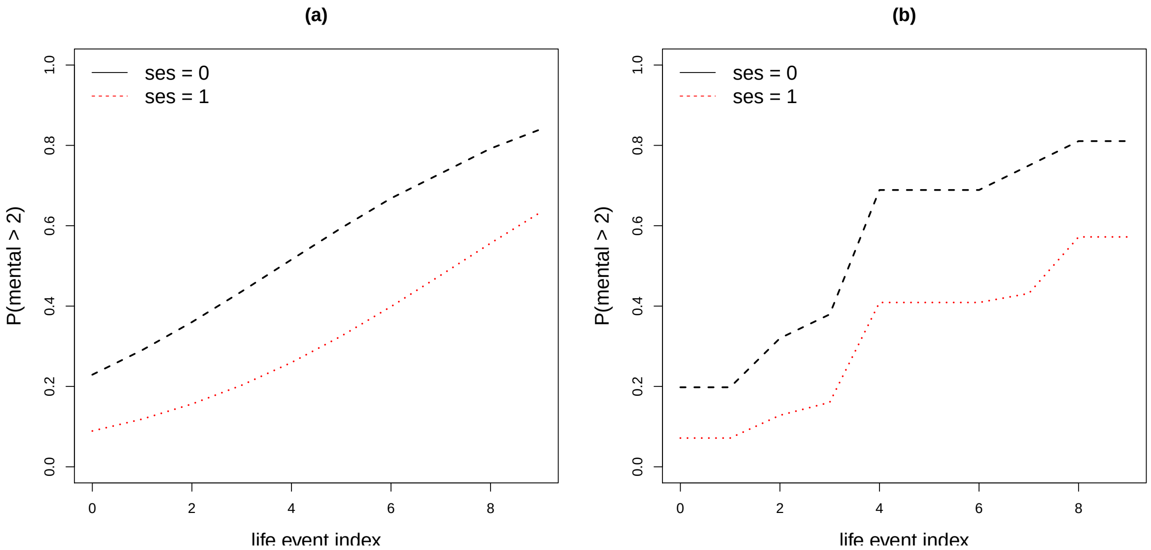

Figure 5: Demonstration of the proportional odds model with the `mental` data set. The estimated probability that mental impairment is moderate or impaired is plotted against the life event index at two socio-economic levels. (a) `polr` fit , and (b) `cgam` fit.

child, new job, divorce, or death of a family member. It is an integer from 0 to 9. Another covariate is socio-economic status and it is measured as binary: high = 1, low = 0. We model the relationship between the latent variable and life event index as monotonically increasing with socio-economic status as a binary covariate. The increasing fit is compared with the fit by the `polr` routine in **MASS**, and the increasing fit has a smaller residual deviance.

```
R> data("mental", package = "cgam")
R> fit.polr <- polr(factor(mental) ~ life + ses, data = mental)
R> fit.incr <- cgam(mental ~ incr(life) + ses, data = mental, family = Ord)
R> fit.polr$deviance

[1] 99.0979

R> fit.incr$deviance

[1] 95.86395
```

For each fit, we can check the estimated probability that the response mental impairment is in each of the four categories.

```
R> probs.polr <- fitted(fit.polr)
R> probs.incr <- fitted(fit.incr)
R> head(probs.polr)

          1         2          3          4
1 0.6249159 0.2564222 0.07131474 0.04734725
2 0.1150236 0.2518325 0.24398557 0.38915837
```

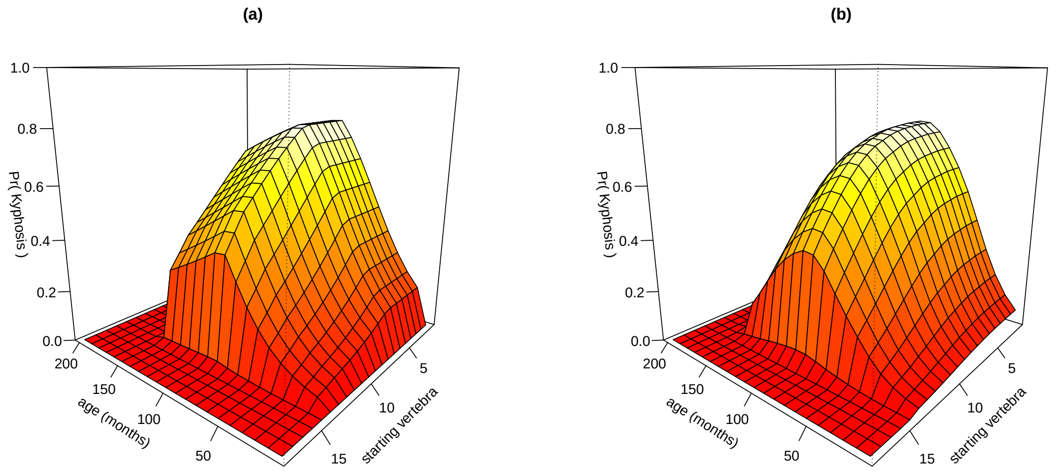

Figure 6: Demonstration of constrained regression using the **cgam** function with the `kyphosis` data set. The surface represents the estimated probability of the response `Kyphosis` to be present. (a) surface without smoothing, and (b) smooth surface.

```
3 0.3902843 0.3502175 0.14495563 0.11454256
4 0.4682289 0.3287373 0.11707594 0.08595788
5 0.2850342 0.3548980 0.18808642 0.17198137
6 0.6962128 0.2146344 0.05428168 0.03487110
```

```
R> head(probs.incr)
```

```
          1         2          3          4
1 0.7236150 0.2048051 0.04540199 0.02617791
2 0.1311317 0.2966802 0.25416212 0.31802593
3 0.2258156 0.3651875 0.21461052 0.19438639
4 0.5138317 0.3258075 0.09792746 0.06243331
5 0.3002185 0.3798198 0.17903211 0.14092958
6 0.7236150 0.2048051 0.04540199 0.02617791
```

2.5. Binomial response example

We use the `kyphosis` data set with 81 observations from the **gam** package to show 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` (starting vertebra). The non-smooth fit and the smooth fit are shown in Figure 6 by `plotpersp`.

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

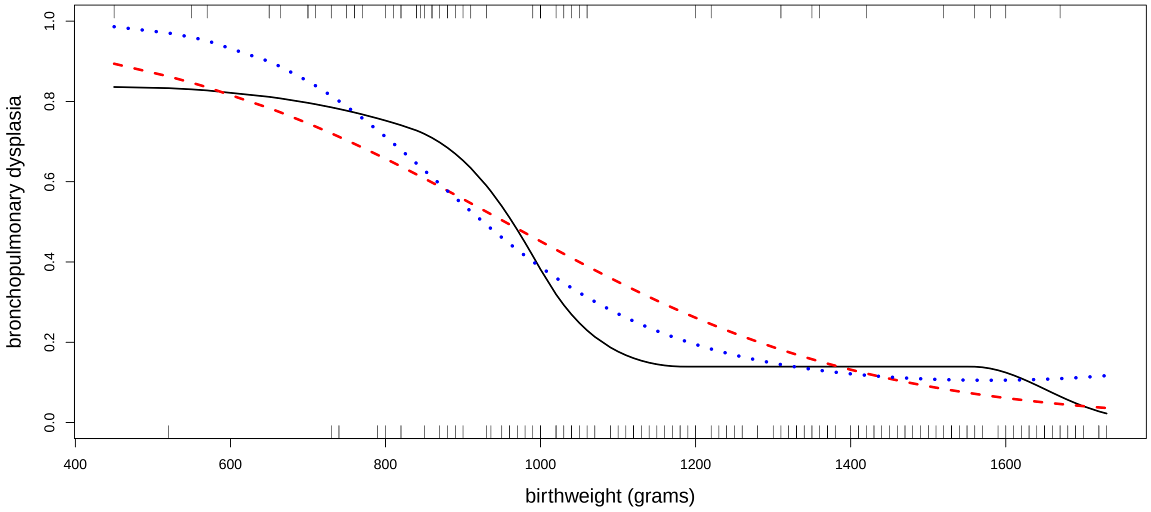

Figure 7: The estimated probability of bronchopulmonary dysplasia as a function of birth weight. The data are shown as tick marks at the presence (1) and the absence (0) of the condition. The solid curve is the smoothly decreasing fit, the dashed curve is the linear log-odds fit, and the dotted curve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

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

Next, we consider the `bpd` data set in the **SemiPar** package (Wand 2018). 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> data("bpd", package = "SemiPar")
R> fit.s.decr <- cgam(BPD ~ s.decr(birthweight, space = "Q"),
+    family = binomial, data = bpd)
```

2.6. 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.

```
R> fit.cgam <- cgam(daysabs ~ decr(math) + decr(langarts) +
+    factor(male), family = poisson, data = attendance)
R> summary(fit.cgam)

Call:
cgam(formula = daysabs ~ decr(math) + decr(langarts) + factor(male),
    family = poisson, data = attendance)

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: 2409.82 on 315 degrees of freedom
Residual deviance: 2100.67 on 298 observed degrees of freedom
CIC: -9.7549
```

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

```
R> fit.cgam.s <- cgam(daysabs ~ s.decr(math) + s.decr(langarts) + factor(male),
+    family = poisson, data = attendance)
R> summary(fit.cgam.s)

Call:
cgam(formula = daysabs ~ s.decr(math) + s.decr(langarts) + factor(male),
    family = poisson, data = attendance)

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: 2409.82 on 315 degrees of freedom
Residual deviance: 2201.237 on 306.4 observed degrees of freedom
CIC: -9.4558
```

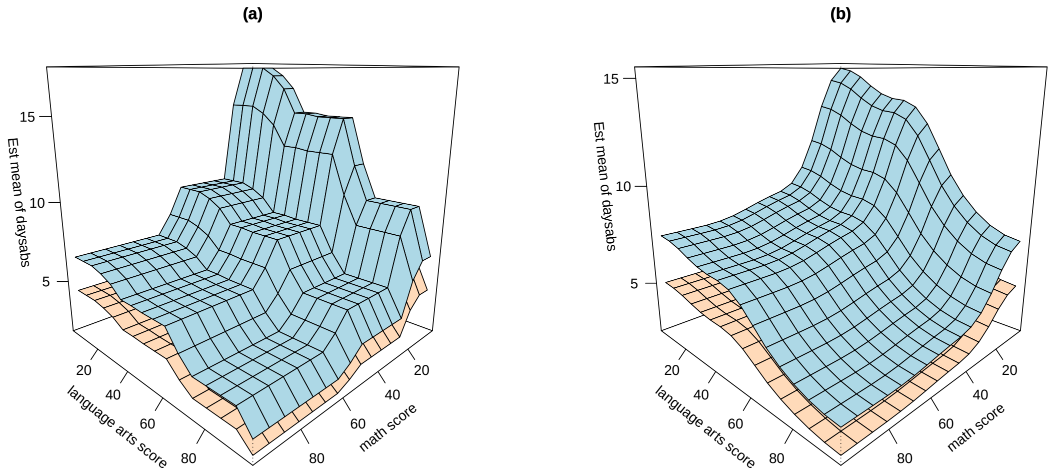

Figure 8: Demonstration of constrained regression using the **cgam** function with the `attendance` data set. (a) parallel surfaces representing the gender effect in an ascending order. (b) parallel surfaces representing the gender effect in an ascending order with smoothing.

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.

2.7. Fitting a "doubly-decreasing" surface to the `plasma` data set

We again use the `plasma` data set as an example to illustrate the routine `s.decr.decr`, 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 `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 `cgam` and can be checked as an output:

```
R> data("plasma", package = "cgam")
R> fit <- cgam(logplasma ~ s.decr.decr(bmi, logdietfat,
+     numknots = c(10, 10)) + factor(smoke) + factor(vituse), data = plasma)
R> fit$pen

[1] 0.5047263

R> summary(fit)

Call:
cgam(formula = logplasma ~ s.decr.decr(bmi, logdietfat,
     numknots = c(10, 10)) + factor(smoke) + factor(vituse), data = plasma)

Coefficients:
```

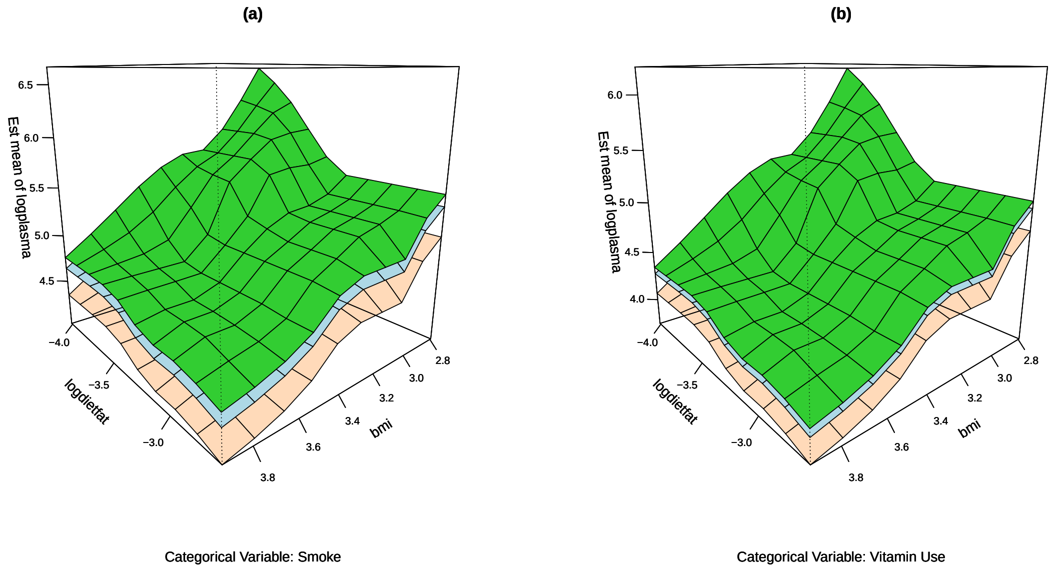

Figure 9: Demonstration of constrained regression using the `s.decr.decr` routine with the `plasma` data set. In each plot, the estimated surface is constrained to be decreasing in both predictors without the assumption of additivity. Parallel surfaces representing the effects of three levels of (a)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

```
                  Estimate  StdErr t.value p.value
(Intercept)         4.0144  0.1237 32.4498  <2e-16 ***
factor(smoke)2      0.2988  0.1251  2.3893  0.0175 *
factor(smoke)3      0.4184  0.1217  3.4372  0.0007 ***
factor(vituse)2    -0.0667  0.1004 -0.6638  0.5073
factor(vituse)3    -0.2757  0.0927 -2.9744  0.0032 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

CIC: 4.9908
```

With 100 simulations, the CIC value is 4.99 for the doubly-decreasing model, and 4.97 for the additive model in Section 2.2; this is evidence that the additive model is adequate.

3. Main routines in package cgam

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_ℓ, 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_ℓ.

3.1. Symbolic functions to specify the form of the component functions

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 the components, 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 quadratic 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 cubic spline (quadratic spline) basis functions are used.

For smooth estimates of monotone surfaces, the functions are: `s.incr.incr`, `s.incr.decr`, `s.decr.incr`, `s.decr.decr`, and two predictors are specified. For example,

```
s.incr.decr(x1, x2)
```

will specify a warped-plane spline fit that is increasing in $\boldsymbol{x}_1$ and decreasing in $\boldsymbol{x}_2$.

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

In the `cgam` routine, the specification of the model can be one of the four exponential families: Gaussian, Poisson, binomial and Gamma. The response can also be an ordered categorical variable, whose distribution does not belong to a single-parameter exponential family, and **cgam** implements the proportional odds model in McCullagh (1980). The symbolic functions are used to specify how the predictors are related to the response. For example,

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

specifies that the response $\boldsymbol{y}$ is from the Gaussian distribution, and $\mathsf{E}(\boldsymbol{y})$ is smoothly increasing and concave in $\boldsymbol{x}_1$, while component function for $\boldsymbol{x}_2$ 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 simulations can be time-consuming, so the default is `nsim` = 100. For example, we can write

```
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 $\hat{\boldsymbol{\eta}}$ and the estimated mean value $\hat{\boldsymbol{\mu}}$, obtained by transforming $\hat{\boldsymbol{\eta}}$ by the inverse of the link function. The CIC value will also be returned if the `nsim` option is given.

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.

For the doubly-monotone fit, the user can also choose to use a penalized version by providing a "large" number of knots for $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ with a penalty term. For example, we want to use ten equally spaced knots for each predictor with a penalty parameter to be 0.1 in a "doubly-decreasing" formula. Then we can write

```
fit.dd <- cgam(y ~ s.decr.decr(x1, x2, numknots = c(10, 10),
  space = c("E", "E")), pen = 0.1)
```

For a doubly-monotone fit, the main values returned are the estimated mean value $\hat{\boldsymbol{\mu}}$ 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.

3.3. 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. The `z` lab represents the estimated regression surface of the mean or the systematic component according to the user's choice. If there is any categorical covariate in a **cgam** model and if the user specifies the argument `categ`, a three-dimensional plot with multiple parallel surfaces, which represent the levels of the categorical covariate in an ascending order, will be created. If `categ` is not specified, a three-dimensional plot with only one surface will be created. Each level of a categorical covariate not used in the plot will be evaluated at its mode.

The basic form of this routine is defined as

```
plotpersp(object,...)
```

The argument `object` represents 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

```
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. Details for the methodology

The package **cgam** gives a comprehensive framework for the generalized additive model with shape and order constraints. We consider models with independent observations from an exponential family with density of the form

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

where the specifications of the functions b and c determine the sub-family of models. The mean vector $\boldsymbol{\mu} = \mathsf{E}(\boldsymbol{y})$ 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 $\boldsymbol{z} \in \mathbb{R}^p$ is a vector of covariates to be parametrically modeled. We specify an additive model

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

where the parameter vector $\boldsymbol{\alpha} \in \mathbb{R}^p$ and the functions f_ℓ, $\ell = 1,\dots,L$, are to be estimated simultaneously. The η function is the "systematic component" (McCullagh and Nelder 1989; Hastie and Tibshirani 1990). We consider the Gaussian, Poisson, binomial and Gamma families in this package; the default is Gaussian.

When the response is ordered categorical, then its density is not of the form (1) and we cannot use one link function to specify the relationship between the response and the predictors. Instead, we use the proportional odds model in McCullagh (1980). Suppose that the response $\boldsymbol{y}$ has the ordered values $1,2,\dots,K$, then the model is

$$\text{logit}[P(y_i \le k\,|\,\boldsymbol{x})] = \gamma_k - \sum_{j=1}^{L} f_j(x_{ji}),\ \ k = 1,\dots,K. \tag{3}$$

The γ_k's are called "cut-points" and $-\infty = \gamma_0 < \gamma_1 < \ldots < \gamma_k = \infty$. An interpretation of the model is that there is an unobserved latent variable $\boldsymbol{y}^*$ which has a continuous distribution $F(\boldsymbol{y}^* - \boldsymbol{\eta})$ where $\boldsymbol{\eta}$ is a location parameter and we model $\boldsymbol{\eta}$ by the additive model (2). The observed y_i is in the kth category if and only if y_i^* is in $(\gamma_{k-1},\gamma_k]$, which implies that $P(y_i \le k\,|\,\boldsymbol{x}) = F(\gamma_k - \eta_i(\boldsymbol{x}))$, and when F is the logistic distribution, we have the model (3).

For modeling smooth constrained f_ℓ, 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 10; see Meyer (2008) for details about these spline bases.

The I-spline basis functions, together with the 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 $\boldsymbol{\phi}_{\ell,i} = f_\ell(x_{\ell,i})$, $i = 1,\dots,n$, for a continuous predictor x_ℓ, and define $\boldsymbol{s}_{\ell,j}$, $j = 1,\dots,m_\ell$ to be the spline basis vectors appropriate for the constraints associated with

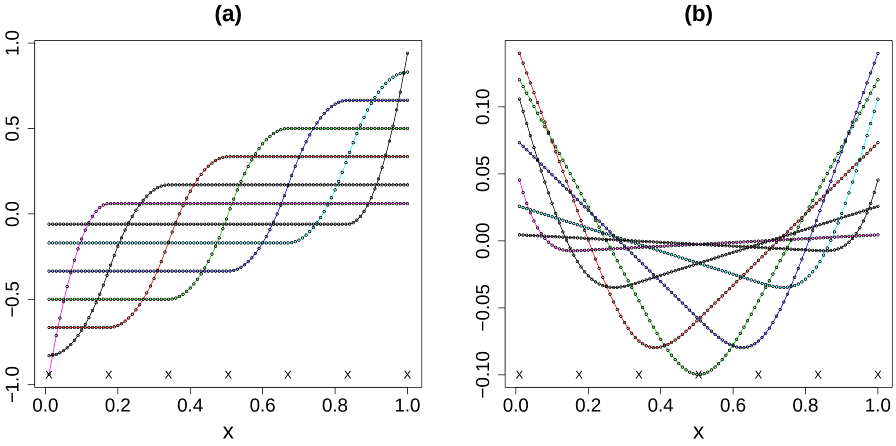

Figure 10: Spline basis functions using seven equally spaced knots marked as "×" for a data set with $n = 100$ observations with values marked as dots. (a) increasing with each basis function centered, and (b) convex with each basis function scaled to be orthogonal to $\mathbf{1}$ and $\boldsymbol{x}$.

x_ℓ. The constraints are satisfied if $\boldsymbol{\phi}_\ell \in \mathcal{C}_\ell$ where for increasing or decreasing constraints,

$$\mathcal{C}_\ell = \left\{ \boldsymbol{\phi} \in \mathrm{I\!R}^n : \boldsymbol{\phi} = a_0\mathbf{1} + \sum_{j=1}^{m_\ell} b_j \boldsymbol{s}_j, \;\; b_j \geq 0, \;\; j = 1, \ldots, m_\ell \right\},$$

and for convex or concave constraints,

$$\mathcal{C}_\ell = \left\{ \boldsymbol{\phi} \in \mathrm{I\!R}^n : \boldsymbol{\phi} = a_0\mathbf{1} + a_1\boldsymbol{x} + \sum_{j=1}^{m_\ell} b_j \boldsymbol{s}_j, \;\; b_j \geq 0, \;\; j = 1, \ldots, m_\ell \right\}.$$

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

$$\mathcal{C}_\ell = \{\boldsymbol{\phi}_\ell \in \mathrm{I\!R}^n : \boldsymbol{A}_\ell \boldsymbol{\phi}_\ell \geq \mathbf{0} \;\text{ and }\; \boldsymbol{B}_\ell \boldsymbol{\phi}_\ell = \mathbf{0}\},$$

for constraint matrices $\boldsymbol{A}_\ell$ and $\boldsymbol{B}_\ell$ 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 $\boldsymbol{A}_\ell$ and $\boldsymbol{B}_\ell$ 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 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 of 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 $\boldsymbol{w}_1, \dots, \boldsymbol{w}_{m_\ell}$, given $\boldsymbol{A}_\ell$ and $\boldsymbol{B}_\ell$, so that

$$\mathcal{C}_\ell = \{\boldsymbol{\phi}_\ell \in \mathrm{I\!R}^n : \boldsymbol{\phi}_\ell = \boldsymbol{v} + \sum_{j=1}^{m_\ell} b_j \boldsymbol{w}_j, \;\; b_j \geq 0, \;\; j = 1, \dots, m_\ell\}. \tag{4}$$

The vector $\boldsymbol{v}$ 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 $\boldsymbol{\eta} = \boldsymbol{\phi}_1 + \cdots + \boldsymbol{\phi}_L + \boldsymbol{Z}\boldsymbol{\beta}$, where $\boldsymbol{\phi}_\ell \in \mathcal{C}_\ell$ for $\ell = 1, \dots, L$, and the rows of the matrix $\boldsymbol{Z}$ are $\boldsymbol{z}_i$, $i = 1, \dots, n$. Each set $\mathcal{C}_\ell$ is a polyhedral convex cone, and let $\mathcal{V}_z$ be the column space of $\boldsymbol{Z}$. Meyer (2013a) showed that $\mathcal{C} = \mathcal{C}_1 + \cdots + \mathcal{C}_L + \mathcal{V}_z$ is also a polyhedral convex cone, where $\boldsymbol{\eta} \in \mathcal{C}$ if $\boldsymbol{\eta} = \boldsymbol{\phi}_1 + \cdots + \boldsymbol{\phi}_L + \boldsymbol{v}$ with $\boldsymbol{v} \in \mathcal{V}_z$ 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 $\boldsymbol{Z}$, together with "edge" vectors $\boldsymbol{e}_1, \dots, \boldsymbol{e}_m$ that are orthogonal to $\mathcal{L}$, so that we can write

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

4.1. Iteratively re-weighted cone projection

For the Gaussian family, fitting the additive model (2) involves a projection of the data vector $\boldsymbol{y}$ onto $\mathcal{C} \subseteq \mathrm{I\!R}^n$. This is accomplished using the `coneB` routine of the R package (see **coneproj**; Liao and Meyer 2014). For other exponential families, an iteratively re-weighted cone projection algorithm is used. The negative log-likelihood

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

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

$$\psi_k(\boldsymbol{\eta}) = \ell(\boldsymbol{\eta}_k) + \nabla \ell(\boldsymbol{\eta}_k)^\top (\boldsymbol{\eta} - \boldsymbol{\eta}_k) + \frac{1}{2} (\boldsymbol{\eta} - \boldsymbol{\eta}_k)^\top \boldsymbol{Q}_k (\boldsymbol{\eta} - \boldsymbol{\eta}_k), \tag{5}$$

where $\nabla \ell(\boldsymbol{\eta}_k)$ is the gradient vector and $\boldsymbol{Q}_k$ 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.

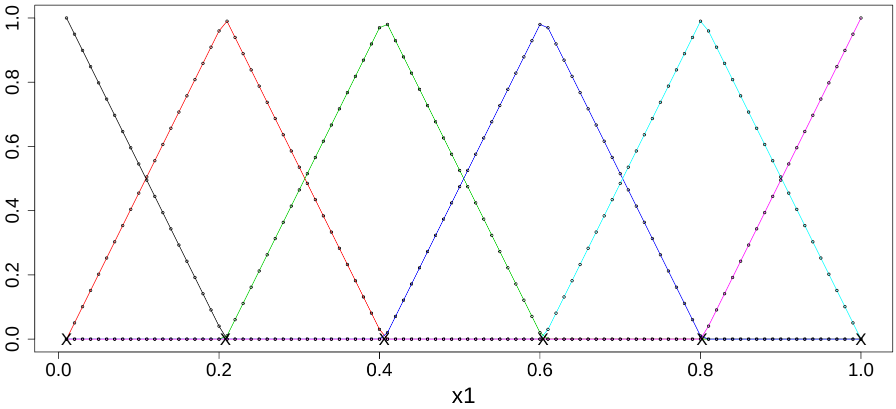

Figure 11: A set of linear basis functions for a predictor in two-dimensional isotonic regression with $n = 100$ observations with values marked as dots and knots marked as "X".

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 maximum likelihood estimate $\hat{\boldsymbol{\mu}}_k$.

When the response is ordered categorical, we use a two-step procedure similar to the backfitting method in Hastie and Tibshirani (1987). At the first step, given an initial valid set of increasing values of the cut-points γ_k's, $\boldsymbol{\eta}$ is estimated using the previous iteratively reweighted cone projection algorithm, and then we update γ_k's by the maximum likelihood estimates. We iterate between the two steps until a convergence criterion is met.

4.2. Two-dimensional monotone regression

For two-dimensional isotonic regression without additivity assumptions, the "warped-plane spline" (WPS) of Meyer (2016) is implemented in **cgam** using the function `s.incr.incr`. The least-squares model has the form

$$y_i = f(x_{1i}, x_{2i}) + \boldsymbol{z}_i^\top \boldsymbol{\alpha} + \sigma\varepsilon_i, \quad \text{for} \;\; i = 1, \ldots, n, \tag{6}$$

where $\boldsymbol{\alpha} \in \mathbb{R}^p$, $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n \in \mathbb{R}^p$ contain values of parametrically modeled covariates, and the ε_i's are mean-zero random errors. We know *a priori* that f is continuous and monotone in both dimensions; that is, for fixed x_1, if $x_{2a} \le x_{2b}$, then $f(x_1, x_{2a}) \le f(x_1, x_{2b})$, 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}, $i = 1, \ldots, n$, we define knots $t_{1,1} < \ldots < t_{1,k_1}$, where $t_{1,1} \le \min(\boldsymbol{x}_1)$ and $t_{1,k_1} \ge \max(\boldsymbol{x}_1)$ are defined by evaluating the basis functions at the design points, that is, $\delta_{1,l_1,i} = \delta_{1,l_1}(x_{1,i}), l_1 = 1, \ldots, k_1$. These basis functions span the space of continuous piece-wise linear functions with given knots, and if we replace $\boldsymbol{\delta}_{1,1}$ with the constant function $\boldsymbol{\delta}_0(\boldsymbol{x}) = \mathbf{1}$, then $\boldsymbol{\delta}_0, \boldsymbol{\delta}_{1,2}, \ldots, \boldsymbol{\delta}_{1,k_1}$ span the same space. Similarly, spline basis functions $\boldsymbol{\delta}_{2,1}, \ldots, \boldsymbol{\delta}_{2,k_2}$ can be defined with knots $t_{2,1} < \ldots < t_{2,k_2}$, where $t_{2,1} \le \min(\boldsymbol{x}_2)$ and $t_{2,k_2} \ge \max(\boldsymbol{x}_2)$. An example of a set of basis functions is in Figure 11. Let the $n \times (k_1 - 1)$ matrix $\boldsymbol{B}_1$ have as columns $\boldsymbol{\delta}_{1,2}, \ldots, \boldsymbol{\delta}_{1,k_1}$, and let the $n \times (k_2 - 1)$ matrix $\boldsymbol{B}_2$ have as columns $\boldsymbol{\delta}_{2,2}, \ldots, \boldsymbol{\delta}_{2,k_2}$. Finally let the $n \times (k_1 - 1)(k_2 - 1)$ matrix $\boldsymbol{B}_{12}$ contain the products of basis vectors, so that column $(l_1 - 2)(k_1 - 1) + l_2 - 1$ of $\boldsymbol{B}_{12}$ is the element-wise product of $\boldsymbol{\delta}_{1,l_1}$ and $\boldsymbol{\delta}_{2,l_2}$, for $l_1 = 2, \ldots, k_1$ and $l_2 = 2, \ldots, k_2$. The columns of $\boldsymbol{B}_1$, $\boldsymbol{B}_2$, and $\boldsymbol{B}_{12}$, 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 $\boldsymbol{\theta}_{ij} = f(x_{1i}, x_{2j})$ be the values of the regression function evaluated at the design points. This is approximated by $\beta_0 \mathbf{1} + \boldsymbol{B}_1 \boldsymbol{\beta}_1 + \boldsymbol{B}_2 \boldsymbol{\beta}_2 + \boldsymbol{B}_{12} \boldsymbol{\beta}_3 = \boldsymbol{B}\boldsymbol{\beta}$, where $\boldsymbol{B} = [\mathbf{1}|\boldsymbol{B}_1|\boldsymbol{B}_2|\boldsymbol{B}_{12}]$ and $\boldsymbol{\beta}^\top = [\beta_0|\boldsymbol{\beta}_1^\top|\boldsymbol{\beta}_2^\top|\boldsymbol{\beta}_3^\top]$. A constraint matrix $\boldsymbol{A}$ will give the necessary and sufficient conditions for monotonicity of the spline basis functions in both predictors, as $\boldsymbol{A}\boldsymbol{\beta} \geq \mathbf{0}$. Here, $\boldsymbol{A}$ 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 $\boldsymbol{y}$ onto the cone

$$\mathcal{C} = \{\boldsymbol{\mu} \in \mathrm{I\!R}^n : \boldsymbol{\mu} = \boldsymbol{B}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{\alpha}; \boldsymbol{A}\boldsymbol{\beta} \geq \mathbf{0}\}; \tag{7}$$

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 across cells is used as the penalty term, where $\lambda > 0$ is a penalty parameter and it will control the constrained "effective degrees of freedom" (EDFC$_\lambda$) 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:

$$\mathrm{GCV}(\lambda) = \frac{\sum_{i=1}^{n}[y_i - \hat{\mu}_{\lambda,i}]^2}{(1 - \mathrm{EDFC}_\lambda/n)^2}. \tag{8}$$

A range of values of λ can be tried and the GCV choice of λ minimizes the criterion. (See Meyer (2016) for more details.)

4.3. Inference methods

For inference regarding a parameter vector $\boldsymbol{\alpha}$, 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 $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_m$, indexed by $J \subseteq \{1, \ldots, m\}$ such that the projection of $\boldsymbol{y}$ onto $\mathcal{C}$ coincides with projection of $\boldsymbol{y}$ onto the linear space spanned by $\boldsymbol{e}_j$, $j \in J$, and the basis vectors for $\mathcal{L}$. Suppose $\boldsymbol{X}_0$ is a matrix so that the columns of $\boldsymbol{X}_0$ and $\boldsymbol{Z}$ span the linear space $\mathcal{L}$. Then if $\boldsymbol{P}_J$ is the projection matrix for the spaced spanned by $\boldsymbol{e}_j$, $j \in J$ and the columns of $\boldsymbol{X}_0$, we can write

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

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

$$\hat{\sigma}^2 = \frac{\mathrm{SSR}}{n - c\mathrm{EDF}},$$

where SSR is the sum of squared residuals 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

Woodroofe (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 σ^2. In the `cgam` routine, the default is $c = 1.2$, but the user can specify $c \in [1, 2]$ using the option `cpar = 1.5` for example.

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 $[\boldsymbol{Z}^\top(\boldsymbol{I} - \boldsymbol{P}_J)\boldsymbol{Z}]^{-1}\hat{\sigma}^2$. See Meyer (2018) and Meyer (2016) 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

$$\mathrm{CIC} = -\frac{2}{n}\log(L) + \log\left\{\frac{2[\mathsf{E}_0(\mathrm{EDF}) + d_0]}{n - d_0 - 1.5\mathsf{E}_0(\mathrm{EDF})} + 1\right\},$$

where L is the likelihood maximized over the cone $\mathcal{C}$, d_0 is the dimension of the linear space $\mathcal{L}$, and $\mathsf{E}_0(\mathrm{EDF})$ is the null expected dimension of the face of $\mathcal{C}$ on which the projection lands. To compute $\mathsf{E}_0(\mathrm{EDF})$, we simulate from (1) and (2) with $f_\ell \equiv 0$ for $\ell = 1\ldots, L$. In this way we get the expected degrees of freedom for the constrained model, in the case where the f_ℓ 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.

5. Speed and utility comparison

To compare the speeds for smooth fitting of regression functions, we simulated 1000 data sets from two regression models, where the first model has three continuous predictors and the second model includes one more categorical covariate. We fit isotonic additive models using `cgam`, `gam` and `scam`. In the first example, x_{1i}, x_{2i}, and x_{3i}, $i = 1, \ldots, n$, were simulated independently and uniformly on $[0, 1]$ along with independent standard normal errors ε_i, $i = 1, \ldots, n$, then, $f_i = 5(x_{1i}^{1/2} + x_{2i}^2 + x_{3i}^3)$ and $y_i = f_i + \varepsilon_i$; in the second example, x_{1i}'s, were simulated uniformly on $[0, 1]$, and x_{2i}'s was simulated to have a positive correlation with x_{1i}'s; x_{3i}'s were simulated uniformly on $[0, 1]$, and a categorical covariate $\boldsymbol{z}$ with values 0 and 1 was simulated such that $z_i = 1$ will be observed more likely for a larger x_{3i}, then, $f_i = 5(x_{1i}^{1/2} + x_{2i}^2 + x_{3i}^3) + z_i$ and $y_i = f_i + \varepsilon_i$. Four sample sizes were used and the results showing the average elapsed time per simulation for each routine are given in Figure 12. The speed comparisons were made on a Mac laptop with a 1.6 GHz dual-core Intel Core i5 CPU.

In addition, we compare the accuracy of each routine in the simulations for the second model using the square root of mean squared error (SMSE) criterion defined as

$$\mathrm{SMSE} = \frac{1}{N}\sum_{k=1}^{N}\sqrt{\frac{\sum_{i=1}^{n}[\hat{f}_{ik} - f_{ik}]^2}{n}},$$

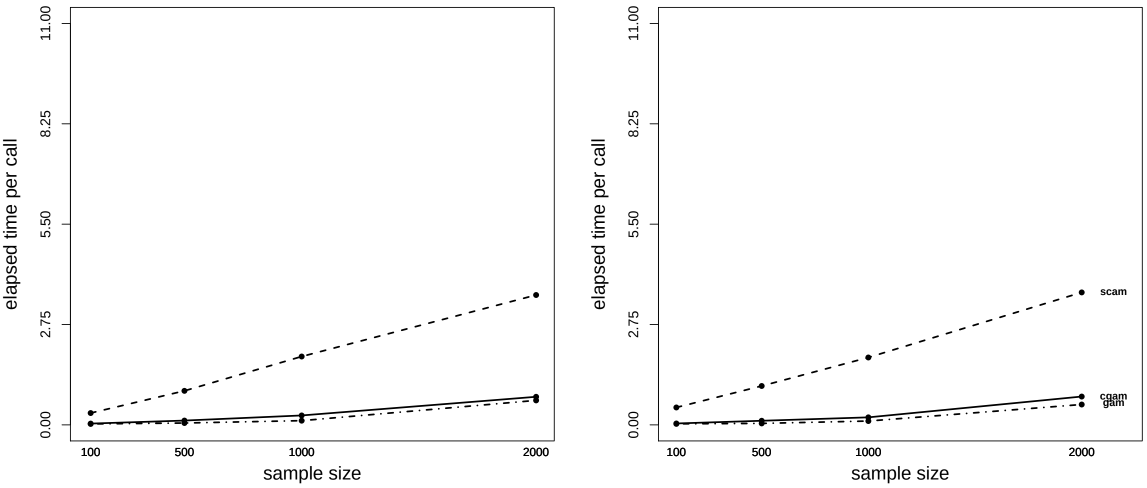

Figure 12: Speed comparisons of `cgam`, `gam` and `scam`. Left: $f = 5(x_1^{1/2} + x_2^2 + x_3^3)$. Right: $f = 5(x_1^{1/2} + x_2^2 + x_3^3) + z$. Solid lines are for `cgam`; dot-dash lines are for `gam`; dashed lines are for `scam`.

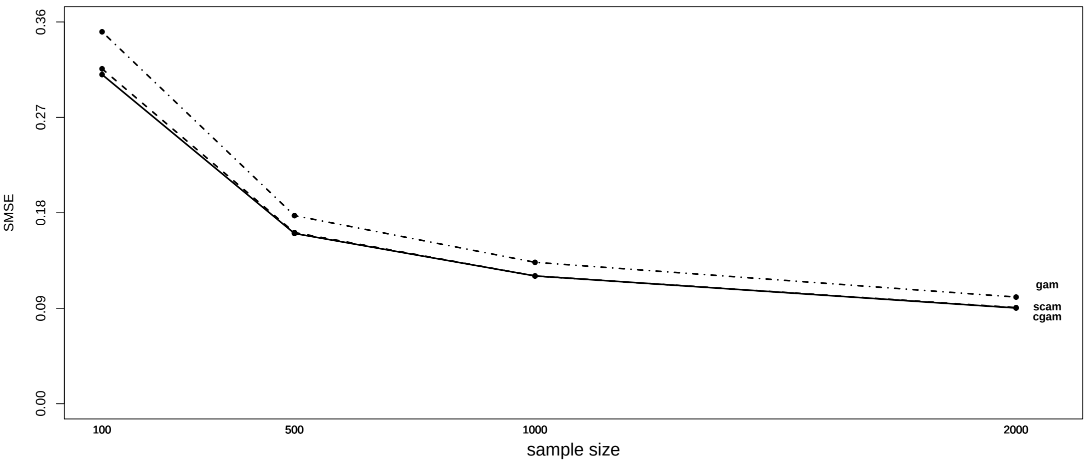

Figure 13: Accuracy comparisons of `cgam`, `gam` and `scam`. $f = 5(x_1^{1/2} + x_2^2 + x_3^3) + z$. Solid lines are for `cgam`; dot-dash lines are for `gam`; dashed lines are for `scam`.

where N is the number of simulations and n is the sample size in each simulation. Comparisons are shown in Figure 13, which shows that `cgam` performs slightly better than `scam` and `gam`.

Continuing with the doubly-increasing example in Section 1, we simulated from $f(x) = 4(x_1 + x_2 - x_1x_2)$ over the unit square with three sample sizes to compare the speed of `cgam` and `scam`. For each sample size, we did 1000 simulations. When $n = 100$, the average elapsed time is about 16 milliseconds by **cgam** and 7.52 seconds by **scam**; when $n = 500$, the time per call by **cgam** and **scam** are about 100 milliseconds and 14.07 seconds; when $n = 2000$, the time per call by **cgam** and **scam** are about 1.21 and 43.45 seconds.

References

Agresti A (2010). *Analysis of Ordinal Categorical Data.* 2nd edition. John Wiley & Sons. doi:10.1002/9780470594001.

Chen Y, Samworth RJ (2016). "Generalized Additive and Index Models with Shape Constraints." *Journal of the Royal Statistical Society B*, **78**(4), 729–754. doi:10.1111/rssb.12137.

Eilers PHC, Marx BD (1996). "Flexible Smoothing with B-Splines and Penalties." *Statistical Science*, **11**(2), 89–121. doi:10.1214/ss/1038425655.

Hastie T, Tibshirani R (1987). "Non-Parametric Logistic and Proportional Odds Regression." *Journal of the Royal Statistical Society C*, **36**(3), 260–276. doi:10.2307/2347785.

Hastie TJ, Tibshirani RJ (1990). *Generalized Additive Models.* Chapman and Hall, Washington, DC.

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. doi:10.18637/jss.v061.i12.

McCullagh P (1980). "Regression Models for Ordinal Data." *Journal of the Royal Statistical Society B*, **42**(2), 109–142. doi:10.1111/j.2517-6161.1980.tb01109.x.

McCullagh P, Nelder JA (1989). *Generalized Linear Models.* 2nd 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**(1), 13–31. doi:10.1016/s0378-3758(99)00025-7.

Meyer MC (2008). "Inference Using Shape-Restricted Regression Splines." *The Annals of Applied Statistics*, **2**(3), 1013–1033. doi:10.1214/08-aoas167.

Meyer MC (2013a). "Semi-Parametric Additive Constrained Regression." *Journal of Nonparametric Statistics*, **25**(3), 715–743. doi:10.1080/10485252.2013.797577.

Meyer MC (2013b). "A Simple New Algorithm for Quadratic Programming with Applications in Statistics." *Communications in Statistics – Simulation and Computation*, **42**(5), 1126–1139. doi:10.1080/03610918.2012.659820.

Meyer MC (2016). "Estimation and Inference for Smooth Regression Surfaces Using Constrained Splines." Unpublished manuscript.

Meyer MC (2018). "Constrained Partial Linear Regression Splines." *Statistica Sinica*, **28**(1), 277–292. doi:10.5705/ss.202016.0342.

Meyer MC, Liao X (2019). ***cgam**: Constrained Generalized Additive Model.* R package version 1.14, URL https://CRAN.R-project.org/package=cgam.

Meyer MC, Woodroofe M (2000). "On the Degrees of Freedom in Shape-Restricted Regression." *The Annals of Applied Statistics*, **28**(4), 1083–1104. doi:10.1214/aos/1015956708.

Nierenberg D, Stukel T, Baron J, Dain B, Greenberg E (1989). "Determinants of Plasma Levels of Beta-Carotene and Retinol." *American Journal of Epidemiology*, **130**(3), 511–521. doi:10.1093/oxfordjournals.aje.a115365.

Pya N, Wood SN (2015). "Shape Constrained Additive Models." *Statistics and Computing*, **25**(3), 543–559. doi:10.1007/s11222-013-9448-7.

R Core Team (2019). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Ruppert D, Wand MP, Carroll RJ (2003). *Semiparametric Regression*. Cambridge University Press, Cambridge.

Venables WN, Ripley BD (2002). *Modern Applied Statistics with S*. 4th edition. Springer-Verlag, New York. doi:10.1007/978-0-387-21706-2.

Wand M (2018). ***SemiPar**: Semiparametic Regression*. R package version 1.0-4.2, URL https://CRAN.R-project.org/package=SemiPar.

Wood S (2019). ***mgcv**: Mixed GAM Computation Vehicle with Automatic Smoothness Estimation*. R package version 1.8-27, URL https://CRAN.R-project.org/package=mgcv.

Wood SN (2017). *Generalized Additive Models: An Introduction with R*. 2nd edition. Chapman & Hall/CRC.

Affiliation:

Xiyue Liao
Department of Statistics
Colorado State University
Fort Collins, Colorado 80523, United States of America
E-mail: liaoxiyue2011@gmail.com

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

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
May 2019, Volume 89, Issue 5 *Submitted:* 2016-06-02
doi:10.18637/jss.v089.i05 *Accepted:* 2018-01-07