

Generalized Additive Models

Mark Andrews

Psychology Department, Nottingham Trent University

✉ `mark.andrews@ntu.ac.uk`

Generalized additive models

- ▶ The polynomial and spline regression models can be regarded as special cases of a more general type of regression model known as a *generalized additive model* (GAM).
- ▶ Given n observations of a set of L predictor variables $x_1, x_2 \dots x_1 \dots x_L$ and outcome variable y , where $y_i, x_{1i}, x_{2i} \dots x_{1i} \dots x_{Li}$ are the values of the outcome and predictors on observation i , then a GAM regression model of this data is:

$$y_i \sim D(\mu_i, \psi), \quad \mu_i = f_1(x_{1i}) + f_2(x_{2i}) + \dots + f_L(x_{Li}), \quad \text{for } i \in 1 \dots n,$$

where D is some probability distribution with parameters ψ , and each predictor variable f_l is a *smooth function* of the predictor variable's values. Usually each smooth function f_l is a weighted sum of basis functions such as spline basis functions or other common types, some of which we describe below.

Generalized additive models “smooths”

- The smooth functions f_l might be defined as follows:

$$f_l(x_{li}) = \beta_{l0} + \sum_{k=1}^K \beta_{lk} \phi_{lk}(x_{li}),$$

where ϕ_{lk} is a basis function of x_{li} .

More general GAMs

- Instead of the outcome variable being described by a probability distribution D where the value of μ_i is the sum of smooth functions of the values of predictor variable at observation i , just as in the case of generalized linear models, we could transform μ_i by a deterministic *link function* g as follows:

$$y_i \sim D(g(\mu_i), \psi), \quad \mu_i = f_1(x_{1i}) + f_2(x_{2i}) + \dots + f_L(x_{Li}), \quad \text{for } i \in 1 \dots n.$$

More general still GAMs

- ▶ More generally still, each smooth function may in fact be a multivariate function, i.e. a function of multiple predictor variables.
- ▶ Thus, for example, a more general GAM than above might be as follows:

$$y_i \sim D(g(\mu_i)),$$

$$\mu_i = f_1(x_{1i}) + f_2(x_{2i}, x_{3i}, x_{4i}) + \dots + f_L(x_{Li}), \quad \text{for } i \in 1 \dots n,$$

where in this case, f_2 is a 3-dimensional smooth function.

Using *mgcv*

- ▶ The R package *mgcv* is a powerful and versatile toolbox for using GAMs in R.
- ▶ We will use a classic data-set often used to illustrate nonlinear regression

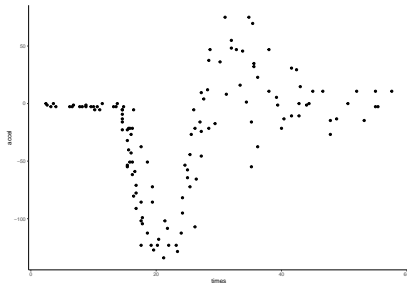


Figure 1: Head acceleration over time in a simulated motorcycle crash.

Using mgcv

- ▶ The main function we will use from mgcv is `gam`.
- ▶ By default, `gam` behaves just like `lm`.

```
library(mgcv)
```

```
M_0 <- gam(accel ~ times, data = mcycle)
```

- ▶ In order to use `gam` to do basis function regression, we must apply what mgcv calls *smooth terms*.
- ▶ There are many smooth terms to choose from in mgcv and there are many methods to specify them.
- ▶ Here, we will use the function simply named `s` to set up the basis functions.
- ▶ The default basis functions used with `s` are *thin plate splines*.

```
M_1 <- gam(accel ~ s(times), data = mcycle)
```

gam with s fit

- The plot of the fit of the above model can be accomplished using the base R plot function.

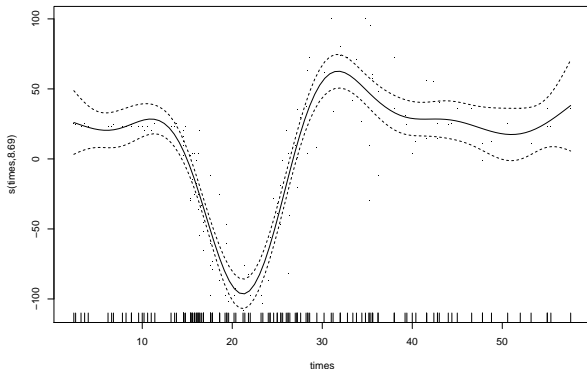


Figure 2: A thin plate spline basis function regression model applied to the cycle data set.

gam with s summary

```
summary(M_1)$s.table  
#>                edf   Ref.df         F      p-value  
#> s(times) 8.693314 8.971642 53.51503 2.957613e-71
```

- ▶ The edf is the effective degrees of freedom of the smooth term.
- ▶ We can interpret its values in terms of polynomial terms.
- ▶ In other words, an edf close to one means the smooth term is effectively a linear function, while an edf close to 2 or close to 3, and so on, are effectively quadratic, cubic, and so on, models.
- ▶ The F statistic and p-value that accompanies this value tells us whether the function is significantly different from a horizontal line, which is a linear function with a zero slope.
- ▶ Even if the edf is greater than 1, the p-value may be not significant because there is too much uncertainty in the nature of the smooth function.

gam rank

- ▶ The number of basis functions used by `s` is reported by the `rank` attribute of the model.
- ▶ In our model, we see that it is 10.

```
M_1$rank  
#> [1] 10
```

- ▶ In general, `mgcv` will use a number of different methods and constraints, which differ depending on the details of the model, in order to optimize the value of `k`.
- ▶ We can always, however, explicitly control the number of basis functions used by setting the value of `k` in the `s` function.

```
M_2 <- gam(accel ~ s(times, k = 5), data = mcycle)  
M_2$rank  
#> [1] 5
```

gam rank optimization

- ▶ How models with different numbers of bases differ in terms of AIC can be easily determined using the AIC function.
- ▶ To illustrate this, we will fit the same model with a range of value of k from 3 to 30.

```
M_k_seq <- map(seq(3, 20) %>% set_names(.,.),  
              ~gam(accel ~ s(times, k = .), data = mcycle))  
model_aic <- map_dbl(M_k_seq, AIC)  
which.min(model_aic)  
#> 9  
#> 7
```

gam.check and k.check

- ▶ `gam.check` and `k.check` can be used for diagnosis and checking the number of basis functions

```
k.check(M_2)
#>           k'      edf  k-index p-value
#> s(times)  4 3.725477 0.3954199      0
M_3 <- gam(accel ~ s(times, k = 10), data = mcycle)
k.check(M_3)
#>           k'      edf  k-index p-value
#> s(times)  9 8.693314 1.148959    0.96
```

Smoothing penalty

- ▶ In addition to explicitly setting the number of basis functions, we can also explicitly set the *smoothing penalty* with the `sp` parameter used inside the `s` function.
- ▶ In general, the higher the smoothing penalty, the *less* flexibility in the nonlinear function.
- ▶ For example, very high values of the smoothing penalty effectively force the model to be a linear model.
- ▶ On the other hand, low values of the smoothing penalty may be overly flexible and overfit the data, as we saw above.

Smoothing penalty

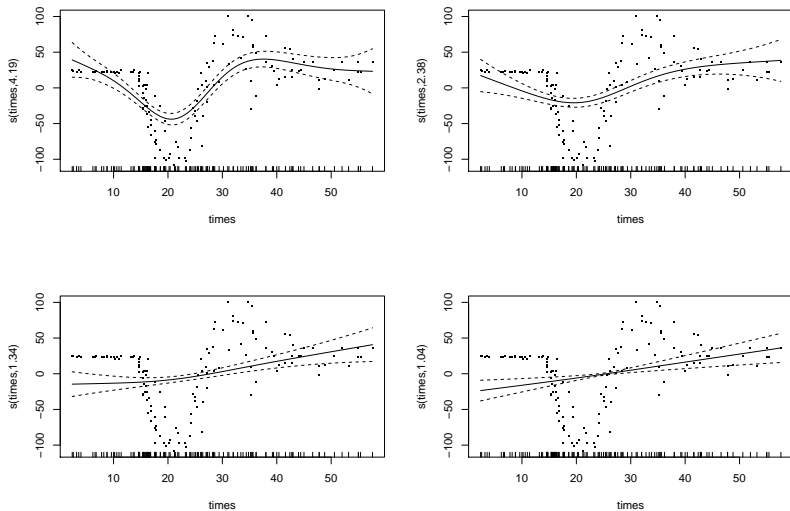


Figure 3: Plots of the fits of Gam models to the `mcycle` data with different

Optimizing smoothing penalty

- ▶ As with `k`, if `sp` is not explicitly set, `mgcv` uses a different methods, including cross-validation, to optimize the value of `sp` for any given model.

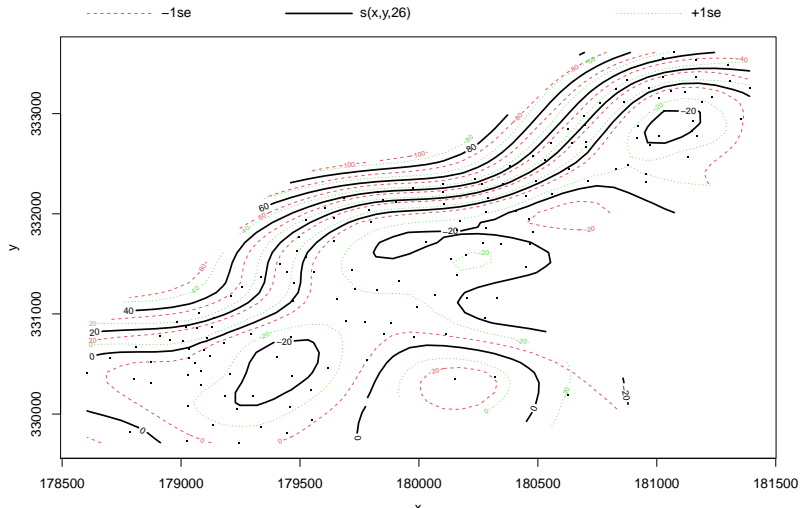
by factor smooth for interactions

- ▶ To model interactions with a categorical predictor variable, we must use *by factor* smooths.
- ▶ This effectively allows us to fit a separate smooth function for each value of the interacting categorical variable.

Multivariate basis functions for spatial etc models

```
meuse <- read_csv('../data/meuse.csv')
```

```
M <- gam(copper ~ s(x, y), data = meuse)
```



Multivariate basis functions for continuous-continuous interactions

- ▶ Gams can handle continuous-continuous interactions not possible otherwise.
- ▶ Let's say we have two predictors x_1 , which is continuous, and x_2 which is binary, then a varying intercept linear model is

$$\mu_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2.$$

- ▶ When $x_2 = 0$, we have

$$\begin{aligned}\mu_i &= \beta_0 + \beta_1 x_1 + \underbrace{\beta_2 x_2}_{=0}, \\ &= \beta_0 + \beta_1 x_1.\end{aligned}$$

- ▶ When $x_2 = 1$, we have

$$\begin{aligned}\mu_i &= \beta_0 + \beta_1 x_1 + \underbrace{\beta_2 x_2}_{=\beta_2}, \\ &= (\beta_0 + \beta_2) + \beta_1 x_1.\end{aligned}$$

- ▶ In R, this $y \sim x_1 + x_2$.

Multivariate basis functions for continuous-continuous interactions

- ▶ A varying slope and varying intercept linear model is

$$\mu_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2.$$

- ▶ When $x_2 = 0$, we have

$$\begin{aligned}\mu_i &= \beta_0 + \beta_1 x_1 + \underbrace{\beta_2 x_2}_{=0} + \underbrace{\beta_3 x_1 x_2}_{=0}, \\ &= \beta_0 + \beta_1 x_1.\end{aligned}$$

- ▶ When $x_2 = 1$, we have

$$\begin{aligned}\mu_i &= \beta_0 + \beta_1 x_1 + \underbrace{\beta_2 x_2}_{=\beta_2} + \underbrace{\beta_3 x_1 x_2}_{=\beta_3 x_1}, \\ &= (\beta_0 + \beta_2) + (\beta_1 + \beta_3) x_1.\end{aligned}$$

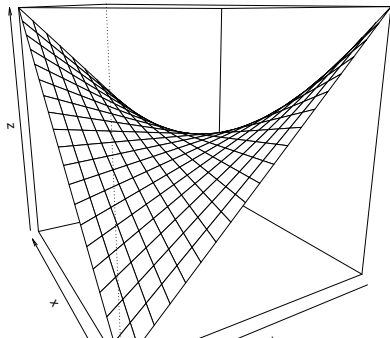
- ▶ In R, this $y \sim x_1 * x_2$.

Multivariate basis functions for continuous-continuous interactions

- ▶ What if x_1 and x_2 are both continuous?
- ▶ What does $y \sim x_1 * x_2$ do?
- ▶ It is still

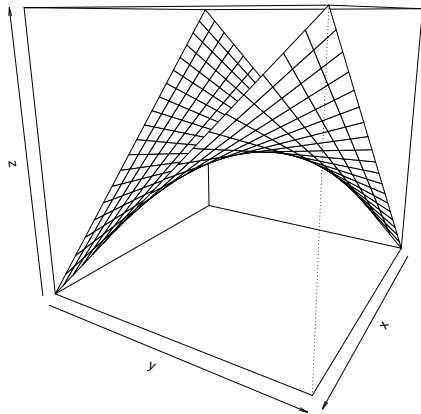
$$\mu_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2,$$

which means that x_1 and x_2 are being multiplied. This means, our function from x_1 and x_2 to μ is essentially



Problematic

- We can also use these types of smooths to deal with continuous-continuous interactions



Multilevel GAM

- Recall that an example of a simple multilevel normal linear model can be defined as follows:

$$y_{ji} \sim N(\mu_{ji}, \sigma^2), \quad \mu_{ji} = \alpha_j + \beta_j x_{ji}, \quad \text{for } i \in 1 \dots n$$

with $\alpha_j \sim N(a, \tau_\alpha^2), \quad \beta_j \sim N(b, \tau_\beta^2) \quad \text{for } j \in 1 \dots J.$

- This model can be rewritten as

$$y_{ji} \sim N(\mu_{ji}, \sigma^2),$$
$$\mu_{ji} = a + v_j + b x_{ji} + \xi_j x_{ji}, \quad \text{for } i \in 1 \dots n, \quad j \in 1 \dots J,$$

with $v_j \sim N(0, \tau_\alpha^2), \quad \xi_j \sim N(0, \tau_\beta^2), \quad \text{for } j \in 1 \dots J.$

Multilevel GAM

- A GAM version of this model might be as follows.

$$\begin{aligned}y_{ji} &\sim N(\mu_{ji}, \sigma^2), \\ \mu_{ji} &= \alpha + \nu_j + f_1(x_{ji}) + f_{2j}(x_{ji}), \quad \text{for } i \in 1 \dots n, \quad j \in 1 \dots J, \\ \text{with } \nu_j &\sim N(0, \tau_\alpha^2), \quad f_{2j} \sim F(\Omega), \quad \text{for } j \in 1 \dots J.\end{aligned}$$

Here, $f_{21}, f_{22} \dots f_{2j} \dots f_{2J}$ are *random smooth functions*, sampled from some function space $F(\Omega)$, where Ω specifies the parameters of that function space.