

# Smooth effect types & Big Data methods

Matteo Fasiolo (University of Bristol, UK)

*matteo.fasiolo@bristol.ac.uk*

June 27, 2018

# Smooth effect types & Big Data methods

## Structure:

- 1 GAM model fitting
- 2 Types of smooth effects
- 3 Big Data methods

# Structure of the talk

## Structure:

- 1 **GAM model fitting**
- 2 Types of smooth effects
- 3 Big Data methods

# GAM model fitting

Recall the GAM model structure:

$$y|\mathbf{x} \sim \text{Distr}\{y|\mu(\mathbf{x}), \boldsymbol{\theta}\}$$

where  $\mu(\mathbf{x}) = \mathbb{E}(y|\mathbf{x}) = g^{-1}\{\sum_{j=1}^m f_j(\mathbf{x})\}$ .

The  $f_j$ 's can be

- parametric e.g.  $f_j(\mathbf{x}) = \beta_1 x_j + \beta_2 x_j^2$
- random effects
- spline-based smooths such as

$$f_j(x_j) = \sum_{i=1}^r \beta_{ji} b_{ji}(x_j)$$

where  $\beta_{ji}$  are coefficients and  $b_{ji}(x_j)$  are known spline basis functions.

NB: we call  $\sum_{j=1}^m f_j(\mathbf{x})$  **linear predictor** because it is linear in  $\boldsymbol{\beta}$ .

# GAM model fitting

$\hat{\beta}$  is the maximizer of **penalized** log-likelihood

$$\hat{\beta} = \operatorname{argmax}_{\beta} \operatorname{PenLogLik}(\beta|\gamma) = \operatorname{argmax}_{\beta} \left\{ \overbrace{L_y(\beta)}^{\text{goodness of fit}} - \underbrace{\operatorname{Pen}(\beta|\gamma)}_{\text{penalize complexity}} \right\}$$

where:

- $L_y(\beta) = \sum_i \log p(y_i|\beta)$  is log-likelihood
- $\operatorname{Pen}(\beta|\gamma)$  penalizes the complexity of the  $f_j$ 's
- $\gamma > 0$  smoothing parameters ( $\uparrow \gamma \uparrow$  smoothness)

# GAM model fitting

We use a hierarchical fitting framework:

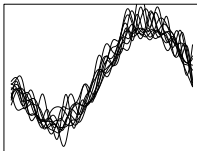
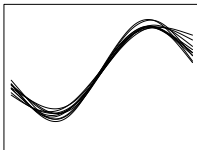
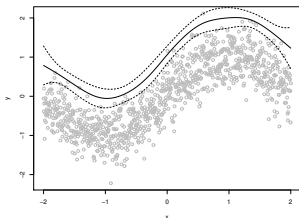
- 1 Select  $\gamma$  determine smoothness

$$\hat{\gamma} = \operatorname{argmax}_{\gamma} \text{LAML}(\gamma)$$

where  $\text{LAML}(\gamma) \approx p(y|\gamma) = \int p(y, \beta|\gamma) d\beta$ .

- 2 For fixed  $\gamma$ , estimate  $\beta$  to determine actual fit

$$\hat{\beta} = \operatorname{argmax}_{\beta} \text{PenLogLik}(\beta|\gamma).$$



Alternatives to Laplace Approximate Marginal Likelihood (LAML) for  $\gamma$  selection:

- Generalized Cross-Validation (GCV)
- Akaike Information Criterion (AIC)

but LAML is most widely applicable in `mgcv` (under name “REML”).

Variance parameters of random effects can be included in  $\gamma$  and estimated by LAML.

Extra parameters  $\theta$  of  $y|\mathbf{x} \sim \text{Distr}\{y|\mu(\mathbf{x}), \theta\}$  handled similarly.

# Structure of the talk

## Structure:

- 1 GAM model fitting
- 2 **Types of smooth effects**
- 3 Big Data methods



# Types of smooths

mgcv offers a wide variety of smooths (see `?smooth.terms`).

Univariate types:

- $s(x) = s(x, bs = "tp")$  thin-plate-splines
- $s(x, bs = "cr")$  cubic regression spline
- $s(x, bs = "ad")$  adaptive smooth

Multivariate type:

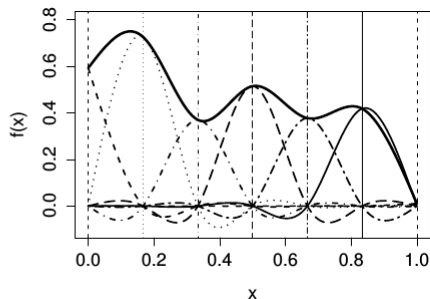
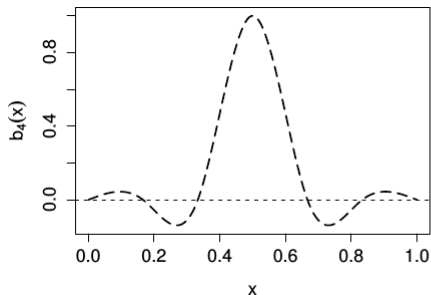
- $s(x_1, x_2) = s(x_1, x_2, bs = "tp")$  thin-plate-splines (isotropic)
- $te(x_1, x_2)$  tensor-product-smooth (anisotropic)
- $s(x, y, bs = "sos")$  smooth on sphere

They can depends on factors:

- $s(x, by = \text{Subject})$
- $s(x, \text{Subject}, bs = "fs")$

# Types of smooths

`s(x, bs = "cr", k = 20)`

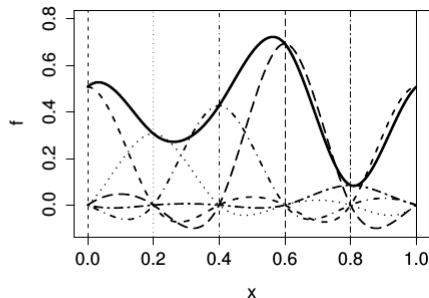
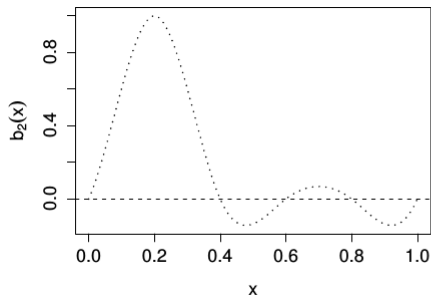


Cubic regression spline are related to the optimal solution to

$$\sum_{i=1}^n \{y_i - f(x_i)\}^2 + \gamma \int f''(x)^2 dx.$$

# Types of smooths

`s(x, bs = "cc")`

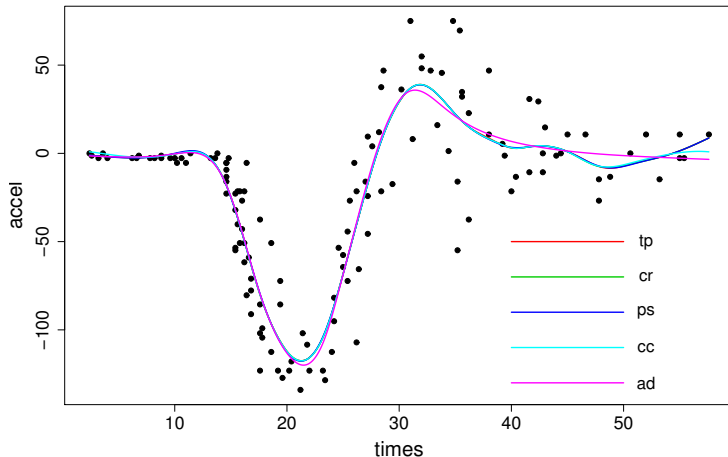


Cyclic cubic regression spline make so that

- $f(x_{min}) = f(x_{max})$
- $f'(x_{min}) = f'(x_{max})$

# Types of smooths

`s(x, bs = "ad")`



The wiggleness or smoothness of  $f(x)$  depends on  $x$ .

# Types of smooths

$s(x_1, x_2), s(x_1, x_2, x_3), \dots$

Based on thin plate regression splines basis.

Related to optimal solution to:

$$\sum_i \{y_i - f(x_i, z_i)\}^2 + \gamma \int f_{xx}^2 + 2f_{xz}^2 + f_{zz}^2 dx dz$$

A single smoothing parameter  $\gamma$ .

Isotropic: same smoothness along  $x_1, x_2, \dots$

# Types of smooths

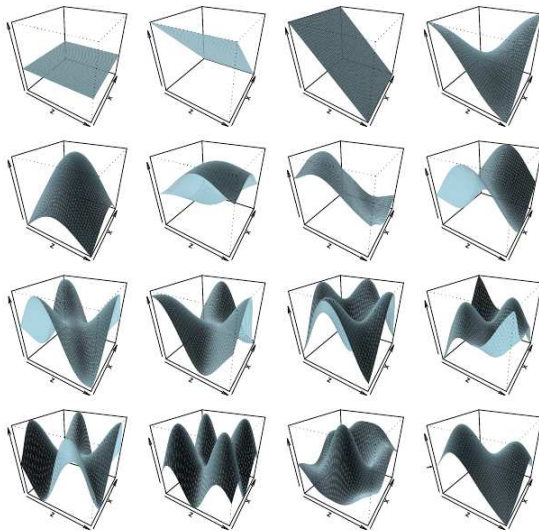


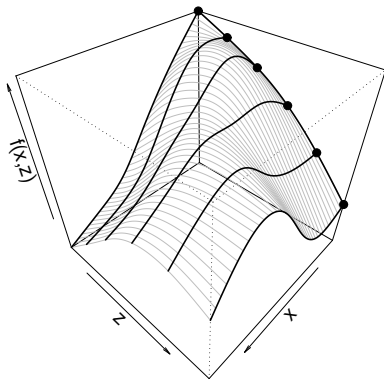
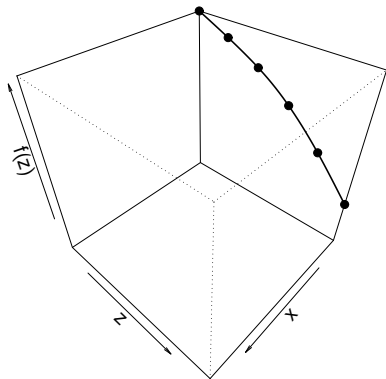
Figure: Rank 17 2D TPRS basis. Courtesy of Simon Wood.

# Types of smooths

Isotropic effect of  $x_1, x_2$  are in same unit (e.g. Km).

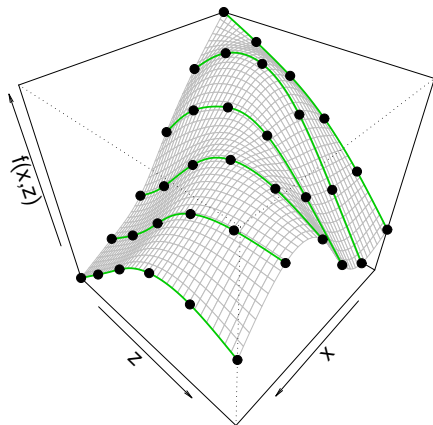
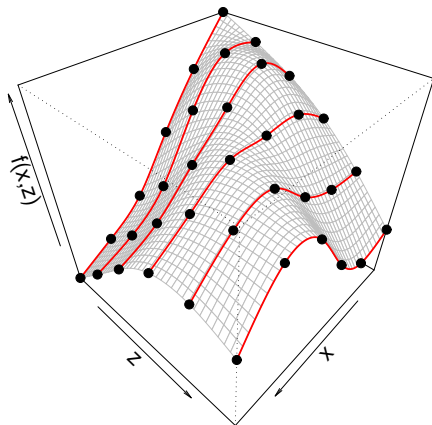
If different units better use tensor product smooths  $\text{te}(x_1, x_2)$ .

Construction: make a spline  $f_z(z)$  a function of  $x$  by letting its coefficients vary smoothly with  $x$



# Types of smooths

- x-penalty: average wiggleness of red curves
- z-penalty: average wiggleness of green curves





# Types of smooths

Can use (almost) any kind of marginal:

- `te(x1, x2, x2)` product of 3 thin-plate-spline bases
- `te(x1, x2, bs = c("cc", "cr"), k = c(10, 6))`
- `te(L0, LA, t, d=c(2,1), k=c(20,10), bs=c("tp","cc"))`

Basis of `te` contains functions of the form  $f(x_1) + f(x_2)$ .

To fit  $f(x_1) + f(x_2) + f(x_1, x_2)$  separately use:

```
y ~ ti(x1) + ti(x2) + ti(x1, x2)
```

# Types of smooths

Sometimes data is allocated to irregular partitions of space (eg regions).

Markov random fields (MRF) can be used for smoothing such data.

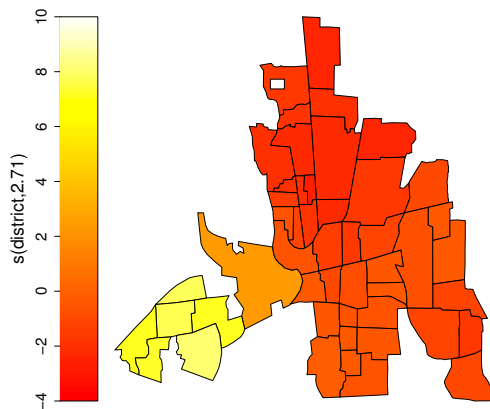
The smooth has a coefficient,  $\gamma_i$ , for each region.

Let  $N_i$  be set of indices of neighbours of region  $i$ .

Then the simplest penalty is  $\sum_i \left\{ \sum_{j \in N_i} (\gamma_i - \gamma_j) \right\}^2$

# Types of smooths

```
data(columb.polys) ## district shapes list  
xt <- list(polys=columb.polys)  
gam(crime ~ s(district,bs="mrf",xt=xt),data=columb)
```



# Types of smooths

## By-factor smooths

Approach (1) is  $s(x, \text{by} = \text{subject})$ , which means

- $\mu_\tau(x) = f_1(x) + \dots$  if subject = 1
- $\mu_\tau(x) = f_2(x) + \dots$  if subject = 2
- ...

Approach (2) is  $s(x, \text{subject}, \text{bs} = "fs")$ , which means

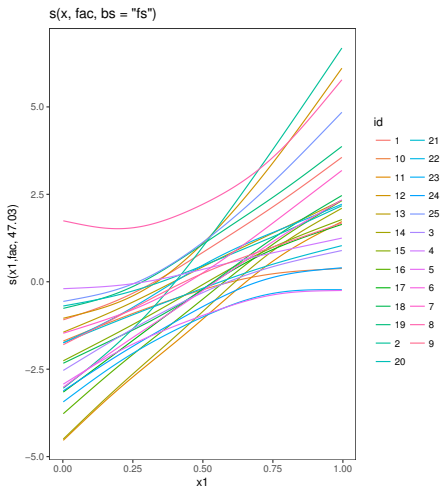
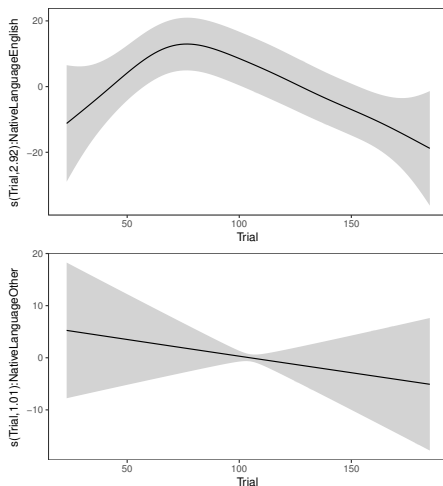
- $\mu_\tau(x) = b_1 + f_1(x) + \dots$  if subject = 1
- $\mu_\tau(x) = b_2 + f_2(x) + \dots$  if subject = 2
- ...

where  $b_1, b_2, \dots \sim N(0, \gamma_{\mathbf{b}} \mathbf{I})$  are random effects.

In (1) each  $f_j$  has its own smoothing parameter.

In (2) all  $f_j$ 's have the same smoothing parameter.

# Types of smooths



# Structure of the talk

## Structure:

- 1 GAM model fitting
- 2 Types of smooth effects
- 3 **Big Data methods**

Recall the GAM model structure:

$$\mu(\mathbf{x}) = \mathbb{E}(y|\mathbf{x}) = g^{-1}\left\{\sum_{j=1}^m f_j(\mathbf{x})\right\}$$

Linear predictor  $\sum_{j=1}^m f_j(\mathbf{x}_i)$  can be written as  $\mathbf{X}_i\boldsymbol{\beta}$ , where:

$$\mathbf{X} = \begin{bmatrix} A_{11} & A_{12} & \cdots & b_{11}(x_{11}) & b_{12}(x_{11}) & \cdots & b_{21}(x_{21}) & \cdots \\ A_{21} & A_{22} & \cdots & b_{11}(x_{12}) & b_{12}(x_{12}) & \cdots & b_{21}(x_{22}) & \cdots \\ \cdot & \cdot & \cdots & \cdot & \cdot & \cdots & \cdot & \cdots \\ \cdot & \cdot & \cdots & \cdot & \cdot & \cdots & \cdot & \cdots \end{bmatrix}$$

has  $n$  rows and

$$p = m + k_1 + \cdots + k_j + \cdots + k_s$$

columns.

# Big Data methods

Bottom line:  $\mathbf{X}$  can get very big, which causes problems:

- storing  $\mathbf{X}$  takes too much memory
- computing things involving  $\mathbf{X}$  (e.g.  $\mathbf{X}^T \mathbf{X}$ ) takes time

Solution implemented in `mgcv::bam` function:

- do not create  $\mathbf{X}$  but only sub-blocks:

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_{11} & \mathbf{X}_{12} \\ \mathbf{X}_{21} & \mathbf{X}_{22} \end{bmatrix}$$

do not store them either, but create them when needed;

- any computation involving  $\mathbf{X}$  is based on the blocks;
- use parallelization when possible;



# Big Data methods

Further acceleration and memory savings by discretization.

Instead of having  $n$  unique rows of  $\mathbf{X}$  discretize to  $b \ll n$  rows.

As long as number of bins  $b = O(\sqrt{n})$  this is ok (statistically).

In mgcv:

```
fit <- bam(y ~ s(x),  
           discrete = TRUE,  
           nthreads = 2,  
           ...)
```

## THANK YOU!

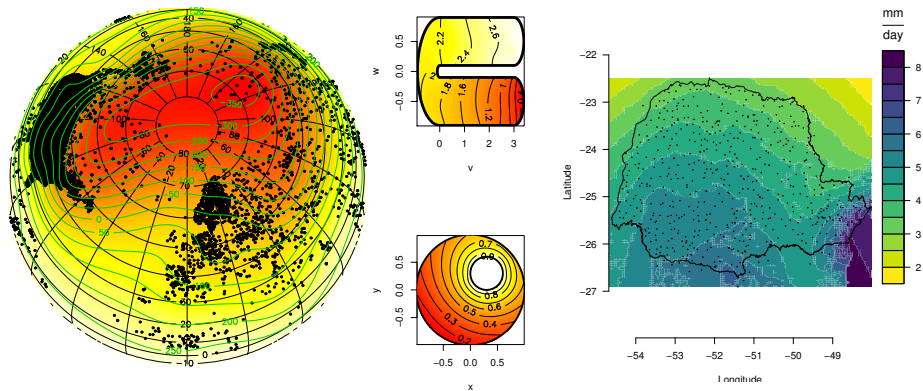


Figure: Examples of quantile GAMs from ?.

# References I