A toolbox of smooth effects and penalties

Matteo Fasiolo

matteo.fasiolo@bristol.ac.uk

Material available at:

`https://github.com/mfasiolo/GAM_Workshop_Dortmund_25`

These slides cover:

1. More on smoothing penalties

2. A toolbox of smooth effects

3. Practical modelling with mgcViz

Recall the GAM model structure:

$$y|\boldsymbol{x} \sim \text{Distr}\{y|\mu(\boldsymbol{x}), \theta_1, \dots, \theta_p\}$$

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

The f_j's can be

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

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

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

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

The β vector estimated by Maximum a Posterior (MAP)

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\text{argmax}} \; \log p(\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{\lambda}) = \underset{\boldsymbol{\beta}}{\text{argmax}} \left\{ \overbrace{\log p(\boldsymbol{y}|\boldsymbol{\beta})}^{\text{goodness of fit}} + \underbrace{\log p(\boldsymbol{\beta}|\boldsymbol{\lambda})}_{\text{prior penalising complexity}} \right\}$$

In `mgcv` we use a Gaussian prior with log-density

$$\log p(\boldsymbol{\beta}|\boldsymbol{\lambda}) \propto -\boldsymbol{\beta}^{\mathsf{T}} \boldsymbol{S}^{\boldsymbol{\lambda}} \boldsymbol{\beta},$$

where

$$\boldsymbol{S}^{\boldsymbol{\lambda}} = \sum_j \lambda_j \boldsymbol{S}_j.$$

and the $\boldsymbol{S}_j$ are positive semi-definite matrices.

NOTE: the penalty is linear w.r.t. $\boldsymbol{\lambda}$.

But penalty $\lambda\boldsymbol{\beta}^\mathsf{T}\boldsymbol{S}\boldsymbol{\beta}$ is not very interpretable.

How do we relate it to something like $\lambda\int f''(x)^2\,dx$?

Consider a spline-based effect

$$f(x) = \sum_k b_k(x)\beta_k = \boldsymbol{b}(x)^\mathsf{T}\boldsymbol{\beta}.$$

We have that

$$\begin{aligned}
\int f''(x)^2 dx &= \int \{\boldsymbol{b}''(x)^\mathsf{T}\boldsymbol{\beta}\}^2 dx \\
&= \int (\boldsymbol{b}''(x)^\mathsf{T}\boldsymbol{\beta})\boldsymbol{b}''(x)^\mathsf{T}\boldsymbol{\beta} dx \\
&= \int \boldsymbol{\beta}^\mathsf{T}\boldsymbol{b}''(x)\boldsymbol{b}''(x)^\mathsf{T}\boldsymbol{\beta} dx \\
&= \boldsymbol{\beta}^\mathsf{T}\left(\int \boldsymbol{b}''(x)\boldsymbol{b}''(x)^\mathsf{T} dx\right)\boldsymbol{\beta} = \boldsymbol{\beta}^\mathsf{T}\boldsymbol{S}\boldsymbol{\beta}.
\end{aligned}$$

So, interpretable penalties can be "translated" to $\lambda \boldsymbol{\beta}^\mathsf{T} \mathbf{S} \boldsymbol{\beta}$.

Note that $\lambda \int f''(x)^2 \, dx$ is just one type of penalty.

The type of penalty and type of bases function are often inter-related.

E.g., cubic regression splines (c.r.s.) approximate the function minimising

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

So, if you want $f(x)$ to be smooth in terms of $f''(x)$, use c.r.s..

Conversely, if you use c.r.s. it makes sense to penalise $f''(x)$.

But there are cases where you can "mix-and-match" bases and penalties.

These slides cover:

1. More on smoothing penalties
2. A toolbox of smooth effects
3. Practical modelling with mgcViz

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

Univariate types:

- `s(x, bs = "bs")` B-splines regression spline
- `s(x, bs = "ad")` adaptive smooth
- `s(x) = s(x, bs = "tp")` thin-plate-splines

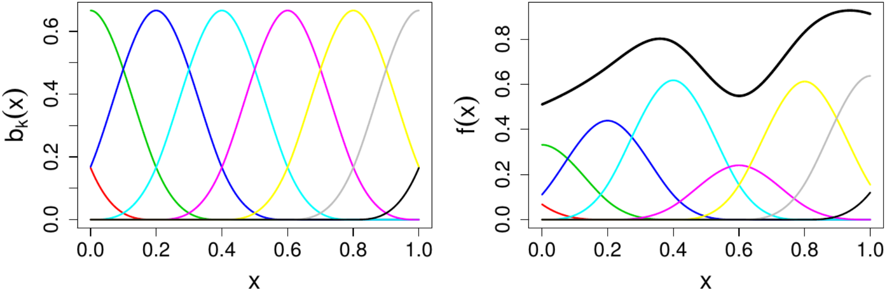
Multivariate type:

- `s(x1, x2) = s(x1, x2, bs = "tp")` thin-plate-splines
- `te(x1, x2)` tensor-product-smooth

They can depend on factors:

- `s(Subject, bs = "re")`
- `s(x, by = Subject)`
- `s(x, Subject, bs = "fs")`

B-splines: `s(x, bs = "bs", m = c(4, 2))`



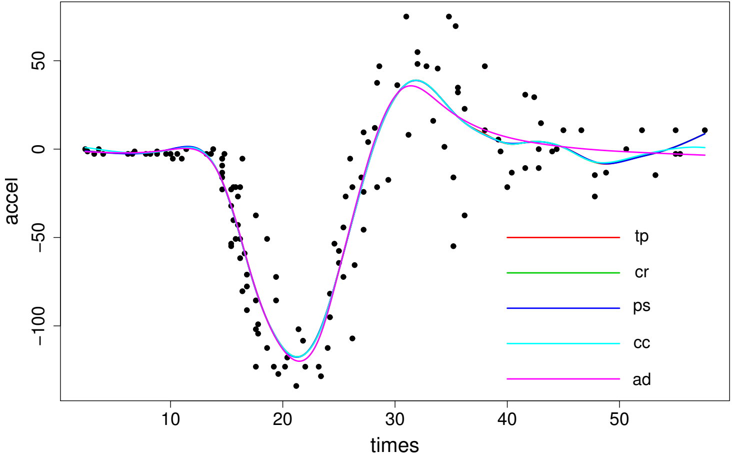
`m[1]` and `m[2]` are orders of the spline basis and penalty.

P-splines are B-splines with penalty such as $\sum_k (\beta_{k+1} - \beta_k)^2$.

Order of basis and penalty can be different, e.g.:

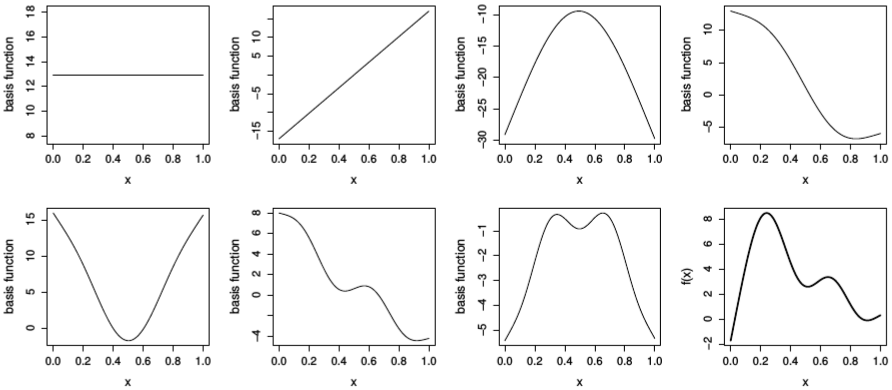
`s(x, bs = "ps", m = c(3, 1))`

Adaptive P-splines: `s(x, bs = "ad")`

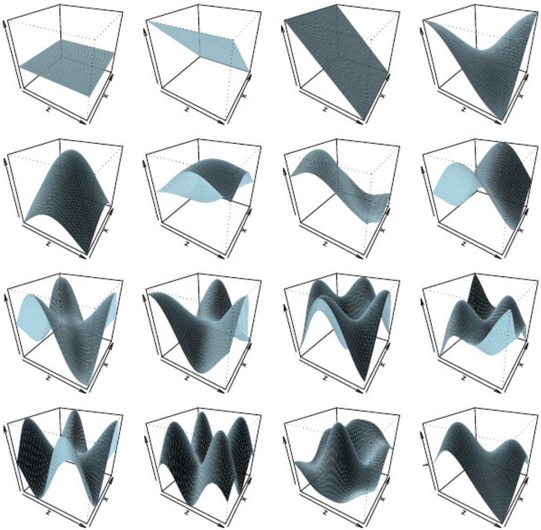


The wiggliness or smoothness of $f(x)$ depends on x.

Thin plate regression splines (TPRS): `s(x)`



Rank 7 TPRS basis. Image from Wood (2006).



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

`s(x1, x2)`, `s(x1, x2, x3)`, ...

Based on thin plate regression splines basis.

They approximate the function minimizing

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

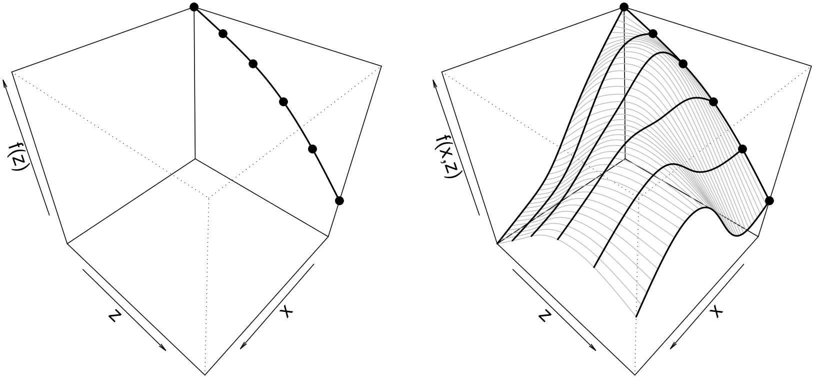
A single smoothing parameter λ.

Isotropic: same smoothness along x_1, x_2, ...

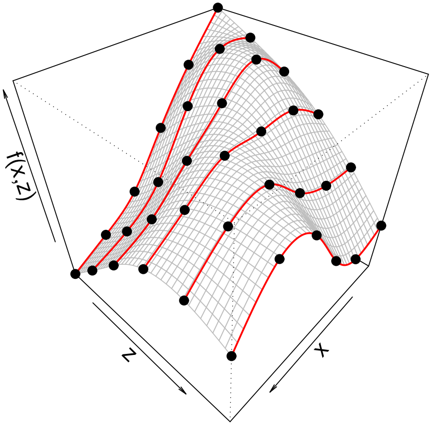
Isotropic effect useful when x_1, x_2 are in same unit (e.g. Km).

If different units better use tensor product smooths `te(x1, x2)`.

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



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



Can use (almost) any kind of marginal:

- `te(x1, x2, x3)` product of 3 cubic regression splines bases
- `te(x1, x2, bs = c("cc", "cr"), k = c(10, 6))`
- `te(LO, LA, t, d=c(2,1), k=c(20,10), bs=c("tp","cc"))`

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

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

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

Random effects

Suppose we have data on bone mineral density (y) vs age (x).

We have m subjects and n data pairs per subject

- subj 1: $\{y_{11}, x_{11}\}, \dots, \{y_{n1}, x_{n1}\}$
- subj j: $\{y_{1j}, x_{1j}\}, \dots, \{y_{nj}, x_{nj}\}$
- subj m: $\{y_{1m}, x_{1m}\}, \dots, \{y_{nm}, x_{nm}\}$

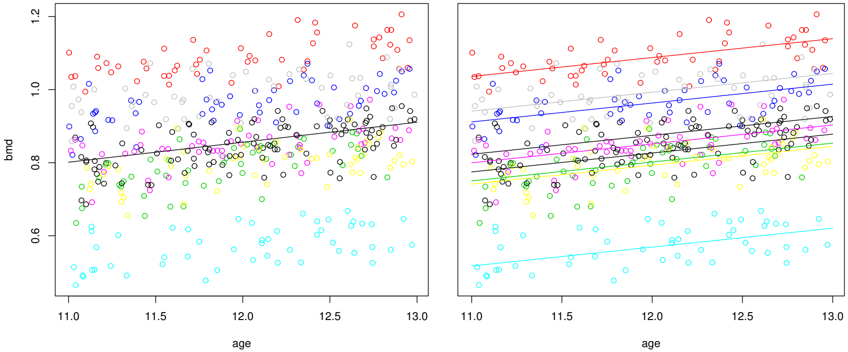
Standard linear model ignores individual differences

$$\mathbb{E}(y|x_{ij}) = \mu(x_{ij}) = \alpha + \beta x_{ij}.$$

We can include random intercept per subject

$$\mu(x_{ij}) = \alpha + \beta x_{ij} + \mathrm{a}_j,$$

with penalty $\lambda_a \sum_j a_j^2 = \lambda_a \boldsymbol{a}^\mathsf{T} \boldsymbol{a}$ via `s(subject, bs = "re")`



We can also include random slopes

$$\mu(x_{ij}) = \alpha + (\beta + b_j)x_{ij} + a_j,$$

with penalty $\lambda_b \boldsymbol{b}^\top \boldsymbol{b}$ via `s(x, subject, bs = "re")`.

By-factor smooths

Approach (1) is `s(x, by = subject)`, which means

- $\mu(x) = f_1(x) + ...$ if subject $= 1$
- $\mu(x) = f_2(x) + ...$ if subject $= 2$
- ...

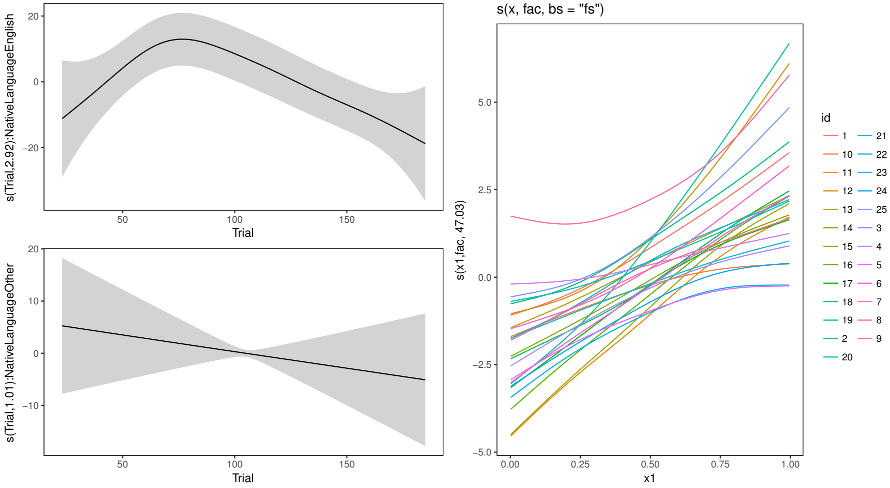
Approach (2) is `s(x, subject, bs = "fs")`, which means

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

where $b_1, b_2, \dots$ are random intercepts.

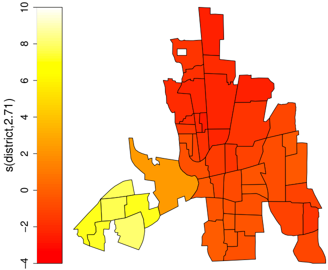
In (1) each f_j has its own smoothing parameter.

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


s(Trial,2.92):NativeLanguageEnglish
20
0
−20
50
100
150
Trial
s(Trial,1.01):NativeLanguageOther
20
10
0
−10
50
100
150
Trial
s(x, fac, bs = "fs")
s(x1,fac, 47.03)
5.0
2.5
0.0
−2.5
−5.0
0.00
0.25
0.50
0.75
1.00
x1
id
1
10
11
12
13
14
15
16
17
18
19
2
20
21
22
23
24
25
3
4
5
6
7
8
9

Markov random field effects

Sometimes data come allocated to irregular partitions of space.



- Markov random fields area a popular way of smoothing such data.
- The smooth has a coefficient, β_i, for each region.
- N_i is the set of indices of the neighbours of region i, then penalty is

$$\lambda \sum_i \sum_{j \in N_i} (\beta_i - \beta_j)^2.$$

Some extra smooths available in `mgcv`:

- smooth on sphere
- soap film smooths
- functional smooths (see `?linear.functional.terms`)

Most effects in `mgcv` are centered $\sum_i f(x_i) = 0$

Effects above are linear w.r.t. $\boldsymbol{\beta}$: $f(\boldsymbol{x}) = \sum_k b_k(\boldsymbol{x})\beta_k$.

`scam` package provides shape-constrained effects.

These are non-linear w.r.t. $\boldsymbol{\beta}$.

`gamFactory` provides nested smooth effect

$$f(\boldsymbol{x}) = s(\tilde{s}(\boldsymbol{x})),$$

where

- $\tilde{s}$ is a scalar-valued transformation
- s is a spline-based smooth

Simple example is single index effect

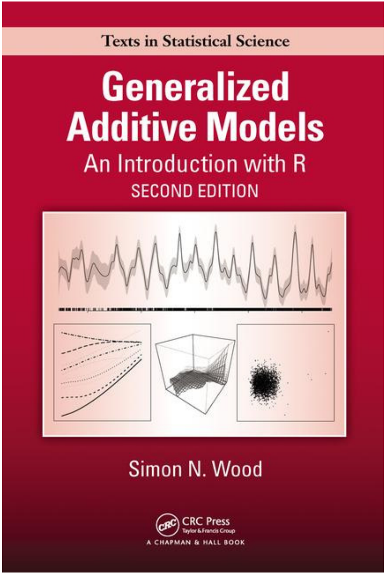
$$f(\boldsymbol{x}) = s(\boldsymbol{\alpha}^\mathsf{T}\boldsymbol{x}).$$

The effect is non linear w.r.t. $\boldsymbol{\alpha}$.

If interested try

```
library(devtools)
install_github("mfasiolo/gamFactory")
```

Further reading



These slides cover:

1. More on smoothing penalties
2. A toolbox of smooth effects
3. Practical modelling with mgcViz

Now we'll look at R_demos/2_effects_and_mgcViz.html

References I

Brezger, A., T. Kneib, and S. Lang (2003). Bayesx: Analysing bayesian structured additive regression models. Technical report, Discussion paper//Sonderforschungsbereich 386 der Ludwig-Maximilians . . . .

Bürkner, P. C. et al. (2017). brms: An r package for bayesian multilevel models using stan. *Journal of Statistical Software 80*(1), 1–28.

Eilers, P. H. and B. D. Marx (1996). Flexible smoothing with b-splines and penalties. *Statistical science 11*(2), 89–121.

Fasiolo, M., R. Nedellec, Y. Goude, and S. N. Wood (2020). Scalable visualization methods for modern generalized additive models. *Journal of computational and Graphical Statistics 29*(1), 78–86.

Hothorn, T., P. Bühlmann, T. Kneib, M. Schmid, and B. Hofner (2010). Model-based boosting 2.0. *The Journal of Machine Learning Research 11*, 2109–2113.

Pya, N. and S. N. Wood (2015). Shape constrained additive models. *Statistics and computing 25*, 543–559.

Rigby, R. A. and D. M. Stasinopoulos (2005). Generalized additive models for location, scale and shape. *Journal of the Royal Statistical Society: Series C (Applied Statistics) 54*(3), 507–554.

References II

Rue, H., S. Martino, and N. Chopin (2009). Approximate bayesian inference for latent gaussian models by using integrated nested laplace approximations. *Journal of the royal statistical society: Series b (statistical methodology) 71*(2), 319–392.

Wood, S. (2006). *Generalized additive models: an introduction with R*. CRC press.