Introduction to penalized splines

3ª Jornada Jóvenes Investigadores SEB

Dae-Jin Lee

Basque Center for Applied Mathematics

https://wp.bcamath.org/jjseb3

Outline

Introduction

Splines, regression splines and Smoothing splines

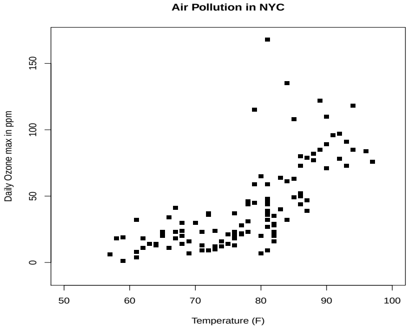
Penalized regression and semi-parametric models

Penalized splines in **R**

From linear regression to penalized regression

Basics

- Scatterplot of pairs $(\boldsymbol{x}_i, \boldsymbol{y}_i)$, $i = 1, ..., n$



- **Assumption:** straight line fits data well

$$\boxed{\mathrm{E}[\boldsymbol{y}_i|\boldsymbol{x}_i] = \boldsymbol{\beta}_0 + \boldsymbol{\beta}_1 x_i}$$

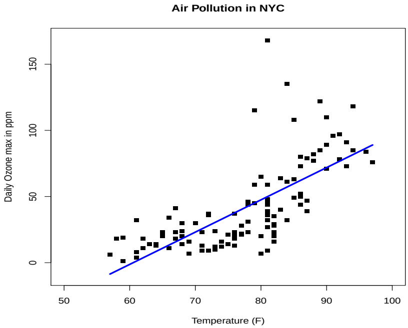
- Minimize least squares criteria

$$\text{mín}\, \|\boldsymbol{y} - \hat{\boldsymbol{y}}\|^2$$

From linear regression to penalized regression

Basics

- Scatterplot of pairs $(\boldsymbol{x}_i, \boldsymbol{y}_i)$, $i = 1, ..., n$



- **Assumption:** straight line fits data well

$$\boxed{\mathsf{E}[\boldsymbol{y}_i|\boldsymbol{x}_i] = \boldsymbol{\beta}_0 + \boldsymbol{\beta}_1 x_i}$$

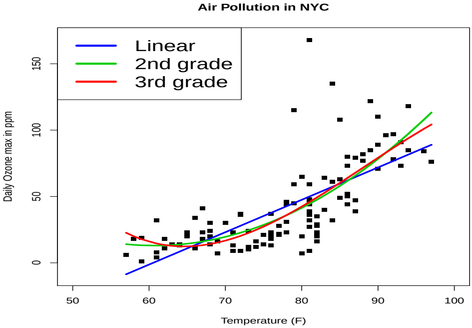
- Minimize least squares criteria

$$\text{mín}\, \|\boldsymbol{y} - \hat{\boldsymbol{y}}\|^2$$

From linear regression to penalized regression

Beyond linear regression

- Linear fit is **too simple** and not always OK
- **Alternative:** use higher degree powers of $\boldsymbol{x}$, with $\boldsymbol{X} = [\ \mathbf{1}_n, \boldsymbol{x}_i, \boldsymbol{x}_i^2, ..., \boldsymbol{x}_i^p\ ]$
- Columns of $\boldsymbol{X}$ are **basis functions** (polynomial regression)



- Same regression equations: $\hat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}$

From linear regression to penalized regression

non-parametric regression or smoothing

- Instead of simple linear regression, we can fit the model:

$$\boxed{\mathsf{E}[\boldsymbol{y}_i|\boldsymbol{x}_i] = f(\boldsymbol{x})}$$

  where $f(\boldsymbol{x})$ is an arbitrary **smooth function**

- The model is "non-parametric" in the sense of no distributional assumptions for the parameters as in a linear model (**scatterplot smoothing**).

- Wide variety of methods (since 80's):
  - kernel smoothing, local linear methods, smoothing splines, etc ... (**out-of-fashion!**)
  - **Penalized regression splines** are getting popular
    - combine a rich set of **basis functions**, with a **roughness penalty**

Splines

- *a kth order spline is a piecewise polynomial function of degree k, that is continuous and has continuous derivatives of orders $1, ...k-1$, at its knot points.*

Splines

- *a* k*th order spline is a piecewise polynomial function of degree* k*, that is continuous and has continuous derivatives of orders* $1, ...k-1$*, at its knot points.*
- *Formally, a function* $f : \mathbb{R} \rightarrow \mathbb{R}$ *is a* k*th order spline with knot points at* $t_1 < ... < t_m$*, if*

Splines

- *a* k*th order spline is a piecewise polynomial function of degree* k*, that is continuous and has continuous derivatives of orders* $1, ...k-1$*, at its knot points.*
- *Formally, a function* $f : \mathbb{R} \rightarrow \mathbb{R}$ *is a* k*th order spline with knot points at* $t_1 < ... < t_m$*, if*
    - f *is a polynomial of degree* k *on each of the intervals* $(-\infty, t_1], [t_1, t_2], ...[t_m, \infty)$*, and*

Splines

- *a kth order spline is a piecewise polynomial function of degree k, that is continuous and has continuous derivatives of orders $1, ...k-1$, at its knot points.*
- *Formally, a function $f : \mathbb{R} \rightarrow \mathbb{R}$ is a kth order spline with knot points at $t_1 < ... < t_m$, if*
  - *f is a polynomial of degree k on each of the intervals $(-\infty, t_1], [t_1, t_2], ...[t_m, \infty)$, and*
  - *$f^{(j)}$, the jth derivative of f, is continuous at $t_1, ..., t_m$, for each $j = 0, 1, ...k-1$.*

Splines

- *a kth order spline is a piecewise polynomial function of degree k, that is continuous and has continuous derivatives of orders $1, ...k-1$, at its knot points.*
- *Formally, a function $f : \mathbb{R} \rightarrow \mathbb{R}$ is a kth order spline with knot points at $t_1 < ... < t_m$, if*
  - *f is a polynomial of degree k on each of the intervals $(-\infty, t_1], [t_1, t_2], ...[t_m, \infty)$, and*
  - *$f^{(j)}$, the jth derivative of f, is continuous at $t_1, ..., t_m$, for each $j = 0, 1, ...k-1$.*
- *By requiring continuous derivatives, we ensure that the resulting function is as smooth as possible.*

Splines

- *a* k*th order spline is a piecewise polynomial function of degree* k*, that is continuous and has continuous derivatives of orders* $1, ...k-1$*, at its knot points.*
- *Formally, a function* $f : \mathbb{R} \rightarrow \mathbb{R}$ *is a* k*th order spline with knot points at* $t_1 < ... < t_m$*, if*
  - f *is a polynomial of degree* k *on each of the intervals* $(-\infty, t_1], [t_1, t_2], ...[t_m, \infty)$*, and*
  - $f^{(j)}$*, the* j*th derivative of* f*, is continuous at* $t_1, ..., t_m$*, for each* $j = 0, 1, ...k-1$*.*
- *By requiring continuous derivatives, we ensure that the resulting function is as smooth as possible.*
- *We can obtain more flexible curves by increasing the degree of the spline and/or by adding knots.*

Splines

- *a kth order spline is a piecewise polynomial function of degree k, that is continuous and has continuous derivatives of orders $1, ...k-1$, at its knot points.*
- *Formally, a function $f : \mathbb{R} \rightarrow \mathbb{R}$ is a kth order spline with knot points at $t_1 < ... < t_m$, if*
  - *f is a polynomial of degree k on each of the intervals $(-\infty, t_1], [t_1, t_2], ...[t_m, \infty)$, and*
  - *$f^{(j)}$, the jth derivative of f, is continuous at $t_1, ..., t_m$, for each $j = 0, 1, ...k-1$.*
- *By requiring continuous derivatives, we ensure that the resulting function is as smooth as possible.*
- *We can obtain more flexible curves by increasing the degree of the spline and/or by adding knots.*
- *However, there is a tradeoff:*

Splines

- *a kth order spline is a piecewise polynomial function of degree k, that is continuous and has continuous derivatives of orders $1, ...k-1$, at its knot points.*
- *Formally, a function $f : \mathbb{R} \to \mathbb{R}$ is a kth order spline with knot points at $t_1 < ... < t_m$, if*
  - *f is a polynomial of degree k on each of the intervals $(-\infty, t_1], [t_1, t_2], ...[t_m, \infty)$, and*
  - *$f^{(j)}$, the jth derivative of f, is continuous at $t_1, ..., t_m$, for each $j = 0, 1, ...k-1$.*
- *By requiring continuous derivatives, we ensure that the resulting function is as smooth as possible.*
- *We can obtain more flexible curves by increasing the degree of the spline and/or by adding knots.*
- *However, there is a tradeoff:*
  - *Few knots/low degree: Resulting class of functions may be too restrictive (bias).*

Splines

- *a kth order spline is a piecewise polynomial function of degree k, that is continuous and has continuous derivatives of orders $1, ...k-1$, at its knot points.*
- *Formally, a function $f : \mathbb{R} \to \mathbb{R}$ is a kth order spline with knot points at $t_1 < ... < t_m$, if*
  - *f is a polynomial of degree k on each of the intervals $(-\infty, t_1], [t_1, t_2], ...[t_m, \infty)$, and*
  - *$f^{(j)}$, the jth derivative of f, is continuous at $t_1, ..., t_m$, for each $j = 0, 1, ...k-1$.*
- *By requiring continuous derivatives, we ensure that the resulting function is as smooth as possible.*
- *We can obtain more flexible curves by increasing the degree of the spline and/or by adding knots.*
- *However, there is a tradeoff:*
  - *Few knots/low degree: Resulting class of functions may be too restrictive (bias).*
  - *Many knots/high degree: We run the risk of overfitting (variance).*

Splines (cont.)

- *The most common case considered is $k = 3$, i.e., that of* **cubic splines**. *These are piecewise cubic functions that are continuous, and have continuous first, and second derivatives.*

Splines (cont.)

- *The most common case considered is $k = 3$, i.e., that of* **cubic splines**. *These are piecewise cubic functions that are continuous, and have continuous first, and second derivatives.*
- *How can we parametrize the set of a splines with knots at a given set of points $t_1, ...t_m$? The most natural way is to use the* **truncated power basis**, $g_1, ...g_{m+k+1}$, *defined as*

$$g_1(x) = 1, \quad g_2(x) = x, ..., \quad g_{k+1}(x) = x^k,$$
$$g_{k+j+1}(x) = (x - t_j)_+^k, \quad j = 1, ..., m.$$

where x_+ denote the positive part of x, i.e. $x_+ = \max\{x, 0\}$.

Splines (cont.)

- *The most common case considered is $k = 3$, i.e., that of* **cubic splines**. *These are piecewise cubic functions that are continuous, and have continuous first, and second derivatives.*
- *How can we parametrize the set of a splines with knots at a given set of points $t_1, ...t_m$? The most natural way is to use the* **truncated power basis**, $g_1, ...g_{m+k+1}$, *defined as*

$$g_1(x) = 1, \quad g_2(x) = x, ..., \quad g_{k+1}(x) = x^k,$$
$$g_{k+j+1}(x) = (x - t_j)_+^k, \quad j = 1, ..., m.$$

  where x_+ denote the positive part of x, i.e. $x_+ = \max\{x, 0\}$.
- *These types of fixed-knot models are referred to as* **regression splines**

Splines (cont.)

- *The most common case considered is $k = 3$, i.e., that of* **cubic splines***. These are piecewise cubic functions that are continuous, and have continuous first, and second derivatives.*
- *How can we parametrize the set of a splines with knots at a given set of points $t_1, ...t_m$? The most natural way is to use the* **truncated power basis***, $g_1, ...g_{m+k+1}$, defined as*

$$g_1(x) = 1, \quad g_2(x) = x, ..., \quad g_{k+1}(x) = x^k,$$
$$g_{k+j+1}(x) = (x - t_j)_+^k, \quad j = 1, ..., m.$$

  where x_+ denote the positive part of x, i.e. $x_+ = \max\{x, 0\}$.
- *These types of fixed-knot models are referred to as* **regression splines**
- *The truncated power basis are:*

Splines (cont.)

- *The most common case considered is $k = 3$, i.e., that of* **cubic splines**. *These are piecewise cubic functions that are continuous, and have continuous first, and second derivatives.*
- *How can we parametrize the set of a splines with knots at a given set of points $t_1, ...t_m$? The most natural way is to use the* **truncated power basis**, $g_1, ...g_{m+k+1}$, *defined as*

$$g_1(x) = 1, \quad g_2(x) = x, ..., \quad g_{k+1}(x) = x^k,$$
$$g_{k+j+1}(x) = (x - t_j)_+^k, \quad j = 1, ..., m.$$

  where x_+ denote the positive part of x, i.e. $x_+ = \max\{x, 0\}$.
- *These types of fixed-knot models are referred to as* **regression splines**
- *The truncated power basis are:*
  - *Conceptually simple.*

Splines (cont.)

- *The most common case considered is $k = 3$, i.e., that of* **cubic splines**. *These are piecewise cubic functions that are continuous, and have continuous first, and second derivatives.*
- *How can we parametrize the set of a splines with knots at a given set of points $t_1, ...t_m$? The most natural way is to use the* **truncated power basis**, $g_1, ...g_{m+k+1}$, *defined as*

$$g_1(x) = 1, \quad g_2(x) = x, ..., \quad g_{k+1}(x) = x^k,$$
$$g_{k+j+1}(x) = (x - t_j)_+^k, \quad j = 1, ..., m.$$

  where x_+ denote the positive part of x, i.e. $x_+ = \max\{x, 0\}$.
- *These types of fixed-knot models are referred to as* **regression splines**
- *The truncated power basis are:*
  - *Conceptually simple.*
  - *Simpler models are nested inside it, leading to straightforward tests of null hypotheses.*

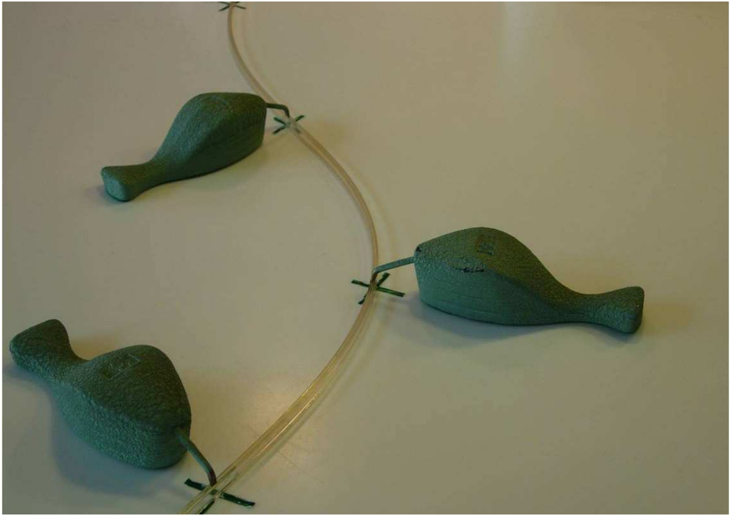
Splines (cont.)

- *The most common case considered is $k = 3$, i.e., that of* **cubic splines**. *These are piecewise cubic functions that are continuous, and have continuous first, and second derivatives.*
- *How can we parametrize the set of a splines with knots at a given set of points $t_1, ...t_m$? The most natural way is to use the* **truncated power basis**, $g_1, ...g_{m+k+1}$, *defined as*

$$g_1(x) = 1, \quad g_2(x) = x, ..., \quad g_{k+1}(x) = x^k,$$
$$g_{k+j+1}(x) = (x - t_j)_+^k, \quad j = 1, ..., m.$$

  where x_+ denote the positive part of x, i.e. $x_+ = \max\{x, 0\}$.
- *These types of fixed-knot models are referred to as* **regression splines**
- *The truncated power basis are:*
  - *Conceptually simple.*
  - *Simpler models are nested inside it, leading to straightforward tests of null hypotheses.*
  - *Computationally inefficient (singularity problems).*

Splines (cont.)

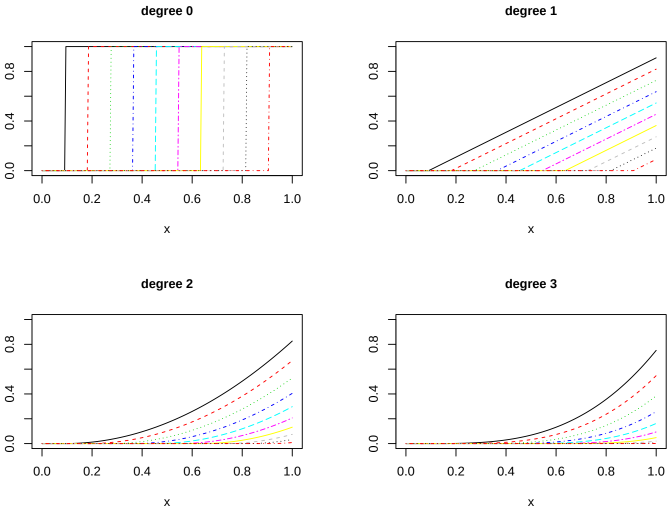


Splines (cont.)

Truncated power basis in R

```
> tpoly<-function(x,t,p){
+ # p order truncated polynomials
+ B=NULL
+ for(i in 1:length(t)){
+ B=cbind(B,(x-t[i])^p*(x>t[i]))}
+ B
+ }
```

Truncated power basis in R



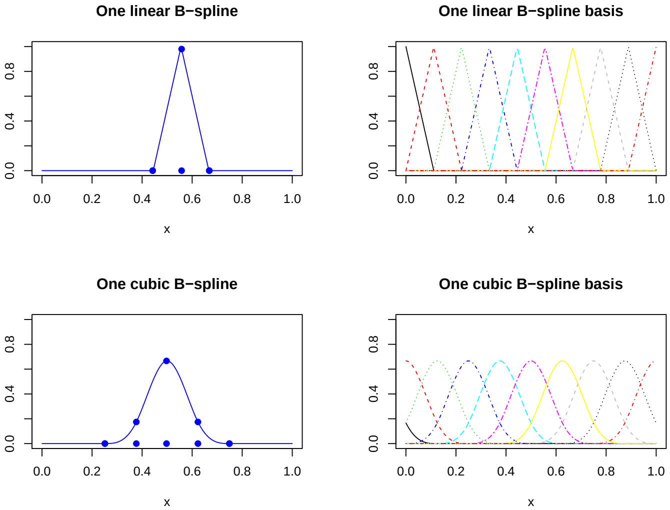
B-splines (de Boor, 1978)

- a much better computational choice, both for speed and numerical accuracy, is the B-spline basis.
- **One linear B-spline**
  - Two pieces, each a straight line, everything else zero
  - Nicely connected at knots (t_1 to t_3) same value
  - Slope jumps at knots
- **One quadratic B-spline**
  - Three pieces, each a quadratic segment, rest zero
  - Nicely connected at knots (t_1 to t_4): same values and slopes
  - Shape similar to Gaussian curve.
- **One cubic B-spline**
  - Four pieces, each a cubic segment, rest zero
  - At knots (t_1 to t_5): same values, first and second derivatives
  - Shape more similar to Gaussian curve.

B-splines in R

```
> library(splines)
> bspline <- function(x, xl, xr, ndx, bdeg){
+  dx <- (xr-xl)/ndx
+  knots <- seq(xl-bdeg*dx, xr+bdeg*dx, by=dx)
+  B <- spline.des(knots,x,bdeg+1,0*x)$design
+  B
+ }
>
> #   xl = left boundry of domain
> #   xr = right boundry of domain
> #  ndx = number of intervals for B-splines.
> # bdeg = degree of B-spline
```

B-splines basis in R



B-splines basis

- Basis matrix B
- Columns are B-splines

$$B = \begin{bmatrix} B_1(x_1) & B_2(x_1) & B_3(x_1) & \dots & B_m(x_1) \\ B_1(x_2) & B_2(x_2) & B_3(x_2) & \dots & B_m(x_2) \\ B_1(x_3) & B_2(x_3) & B_3(x_3) & \dots & B_m(x_3) \\ \vdots & \vdots & \vdots & \dots & \vdots \\ B_1(x_n) & B_2(x_n) & B_3(x_n) & \dots & B_m(x_n) \end{bmatrix}$$

- In each row only a few non-zero elements (degree plus one)
- Demo

```
> library(gamlss.demo)
> demo.BSplines()
```

Smoothing splines

- Regularized regression over the **natural spline basis**
- Minimize *penalized sum-of-squares*:

$$PSS(f,\lambda) = \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \int_{x_1}^{x_n} f^{''}(x)^2 dx$$

- First term balances the goodness-of-fit
- Second term penalizes the second derivative of the function (i.e. the curvature)
- λ is the so-called smoothing parameter that controls the balance between bias and variance.
- They put a knots on every data point $x_1, ...x_n$ (solver the knots selection problem).
- $0 < \lambda < \infty$ if $\lambda = 0$ the fit interpolates the data, when $\lambda \rightarrow \infty$ the second derivative goes to zero, and the fit is linear.
- in `R`, the `smooth.spline()`.

Natural spline basis

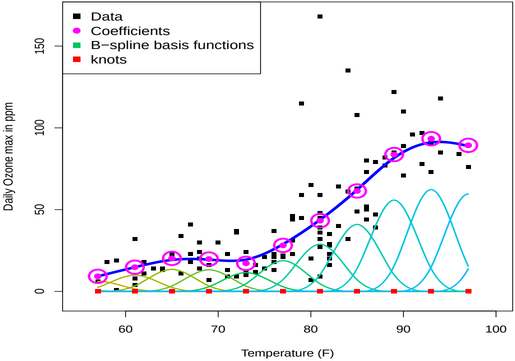
- One problem with regression splines is that the estimates tend to display erractic behavior, i.e., they have high variance, at the boundaries of the domain of $x_1, ...x_n$. This gets worse as the order k gets larger
- A way to remedy this problem is to force the piecewise polynomial function to have a lower degree to the left of the leftmost knot, and to the right of the rightmost knot.
- in `R`, the `ns()`.

- Smoothing splines often deliver similar fits to those from kernel regression.
- However, they are in a sense simpler.
- Both have a tuning parameter the bandwidth h for kernel regression, and the smoothing parameter λ for smoothing splines, which we would typically need to choose by cross-validation.
- But for smoothing splines, we don't require a choice of kernel.
- Smoothing splines are generally much more computationally efficient.

Penalized regression

P-splines (Eilers and Marx, 1996)

- The model: $f(x) = \boldsymbol{B\theta}$, where $\boldsymbol{B}$ is regression basis and $\boldsymbol{\theta}$ the new vector of coefficients which we penalize
- **Example:** Air pollution in NYC



Penalized regression

P-splines (Eilers and Marx, 1996)

- Estimation by **penalized least squares**, such that:

$$\min \|\boldsymbol{y} - \boldsymbol{B}\boldsymbol{\theta}\|^2 \rightarrow \hat{\boldsymbol{\theta}} = (\boldsymbol{B}'\boldsymbol{B} + \boldsymbol{P})^{-1}\boldsymbol{B}'\boldsymbol{y}$$

$$\hat{\boldsymbol{y}} = \boldsymbol{B}\hat{\boldsymbol{\theta}}$$

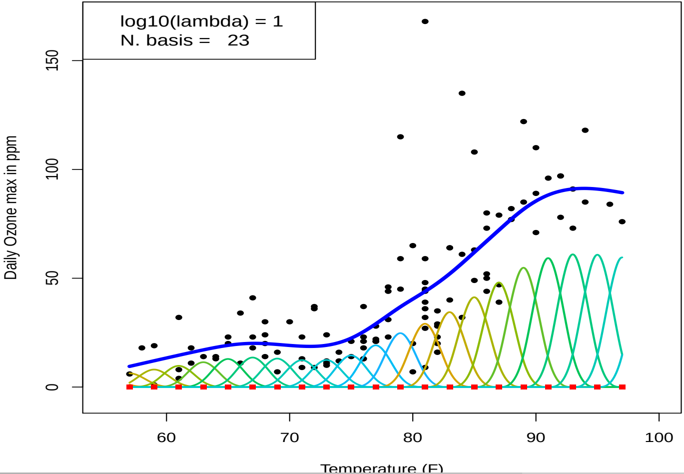
  $\boldsymbol{P}$ is a **roughness penalty for smoothness** controlled by λ

- Choose the size of $\boldsymbol{B}$ and λ (and penalty order)
  - $20 < \text{knots} < 40$
  - $0 < \lambda < \infty$

Penalized regression

P-splines (Eilers and Marx, 1996)

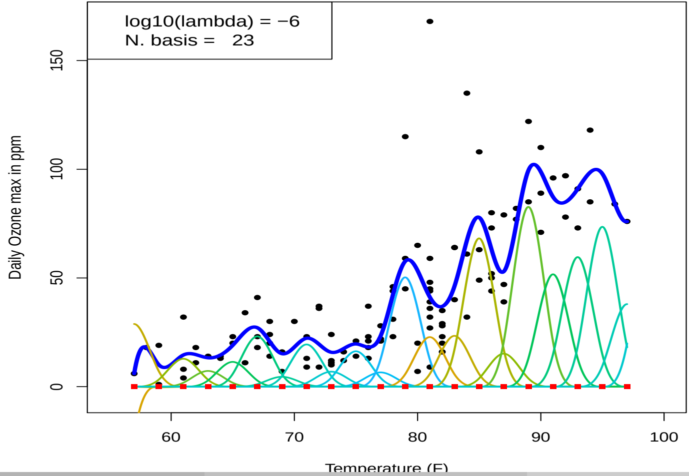
- Choose λ to tune the fit



Penalized regression

P-splines (Eilers and Marx, 1996)

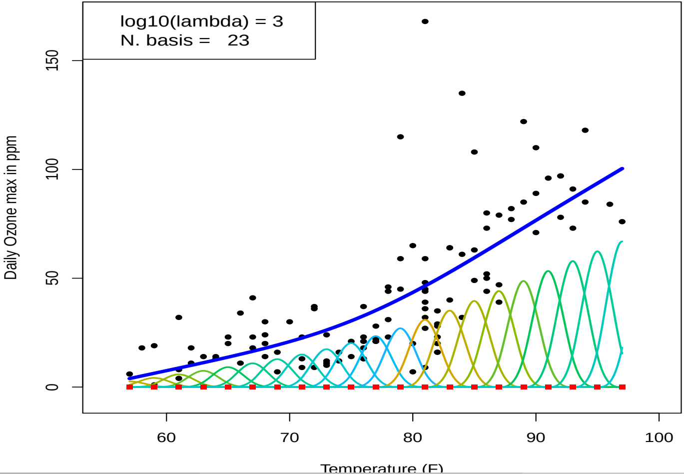
- Choose λ to tune the fit



Penalized regression

P-splines (Eilers and Marx, 1996)

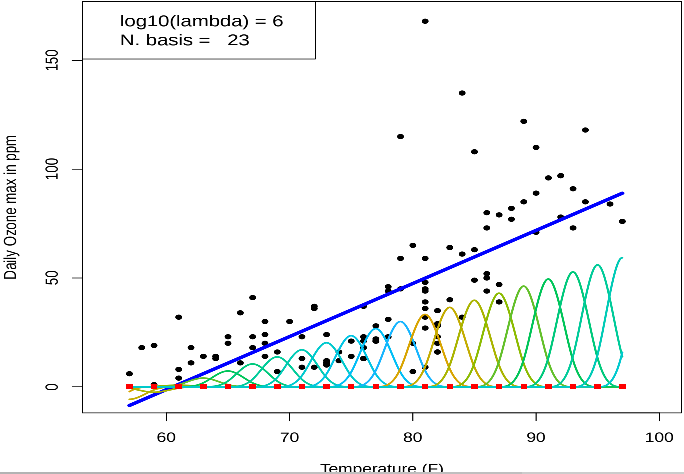
- Choose λ to tune the fit



Penalized regression

P-splines (Eilers and Marx, 1996)

- Choose λ to tune the fit



P-splines

Knots and penalties

- Choose a moderate number of equidistant knots ($k <<< n$), e.g. $20 \leq k \leq 40$
- Add a penalty to the measure of fit, to tune smoothness
  - Discrete differences of coefficients

$$\|\boldsymbol{y} - \boldsymbol{B\theta}\|^2 + \lambda \sum_j (\Delta^d \theta_j)^2$$

where Δ^d is a difference operator of order d, i.e.

$$\Delta \boldsymbol{\theta}_j = \boldsymbol{\theta}_j - \boldsymbol{\theta}_{j-1} \qquad \text{(First order)}$$

$$\Delta^2 \boldsymbol{\theta}_j = \boldsymbol{\theta}_j - 2\boldsymbol{\theta}_{j-1} + \boldsymbol{\theta}_{j-2} \qquad \text{(Second order)}$$

$$\vdots$$

P-splines

Penalties

- We are interested in differences on the coefficients
- $\Delta^d \boldsymbol{\theta} = \boldsymbol{D}_d \boldsymbol{\theta}$ (*Difference matrix operator*)

$$\boldsymbol{D}_1 = \begin{bmatrix} -1 & 1 & 0 & 0 & \cdots \\ 0 & -1 & 1 & 0 & \cdots \\ 0 & 0 & -1 & 1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad \text{or} \quad \boldsymbol{D}_2 = \begin{bmatrix} 1 & -2 & 1 & 0 & \cdots \\ 0 & 1 & -2 & 1 & \cdots \\ 0 & 0 & 1 & -2 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

- The **penalty** becomes $\boldsymbol{P} = \lambda \boldsymbol{D}'\boldsymbol{D}$
- Estimation is done by **penalized least squares**:

$$\text{mín} \, \|\boldsymbol{y} - \boldsymbol{B}\boldsymbol{\theta}\|^2 + \boldsymbol{P} \rightarrow \hat{\boldsymbol{\theta}}_{\lambda^*} = (\boldsymbol{B}'\boldsymbol{B} + \boldsymbol{P})^{-1}\boldsymbol{B}'\boldsymbol{y}$$

- λ^* can be selected by criteria as **AIC**, **BIC**, or **GCV**

Selection of λ

- Optimal λ^* can be selected by criteria as **AIC**, **BIC**, or **GCV**

$$GCV = \sum_{i=1}^{n} \frac{(y_i - \hat{y}_i)^2}{n - trace(\boldsymbol{H})}; \quad \boldsymbol{H} = \boldsymbol{B}(\boldsymbol{B}'\boldsymbol{B} + \lambda \boldsymbol{D}'\boldsymbol{D})^{-1}\boldsymbol{B}'$$

$$AIC = 2log\left(\sum_{i=1}^{n}(y_i - \hat{y}_i)^2\right) - 2\log(n) + 2\log\left(trace(\boldsymbol{H})\right)$$

- Evaluate on a large grid of λ's.

Demo P-splines

- `library(gamlss.demo)`
  `demoPsplines()`

Smoothing and mixed models

a semi-parametric approach

- Reformulate $\boldsymbol{y} = \boldsymbol{B\theta} + \boldsymbol{\epsilon}$ into:

$$\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{Z\alpha} + \boldsymbol{\epsilon}, \quad \begin{pmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\epsilon} \end{pmatrix} \sim \mathcal{N}\left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \boldsymbol{G} & \boldsymbol{0} \\ \boldsymbol{0} & \sigma^2 \boldsymbol{I} \end{pmatrix}\right]$$

  where $\boldsymbol{G} = \sigma_\alpha^2 \boldsymbol{I}$, is the random effects covariance

  λ estimation becomes the ratio σ^2/σ_α^2

- **Flexibility:**

  – Easy incorporation of smoothing in **complex models**: hierarchical models, multi-level models, longitudinal data, correlated errors ...

- **Mixed models theory** for **estimation** and **inference**

- **Extension to non-gaussian data** (Poisson, Binomial, etc ...)

  - Generalized Linear Models (GLM's) to GL(Mixed)M's

Bayesian P-splines

Lang and Brezger (2004)

- The bayesian analogue of P-splines replace differences with Gaussian random walks as priors on the regression coefficients θ_j
- First/second order θ_j, corresponds to a B-splines basis $\boldsymbol{B}$:

$$\theta_j = \theta_{j-1} + v_j \quad \text{or} \quad \theta_j = 2\theta_{j-1} - \theta_{j-2} + v_j$$

  where $v_j \sim N(0\tau^2)$
- The amount of smoothing is controlled by $\tau^2 = \sigma^2/\lambda$
- In general, we can rewrite $\boldsymbol{P}$ as:

$$\boldsymbol{\theta}|\tau^2 \propto exp\left(-\frac{1}{2\tau^2}\boldsymbol{\theta}'\boldsymbol{P}\boldsymbol{\theta}\right)$$

  the **rank** of $\boldsymbol{P}$ is $c-1$ for fist order RW, and $c-2$ for 2nd order (improper prior)
- As previously

$$\boldsymbol{y}|\boldsymbol{\theta} \sim N\left(\boldsymbol{B}\boldsymbol{\theta}, \boldsymbol{I}\sigma^2\right),$$

- Software `BayesX`, `Inla`
- Hierarchical models (mixed model representation): `BuGS`, `JAGS`

Multidimensional P-splines

- (Generalized) Additive Models (Hastie & Tibshirani, 1990)

$$\boldsymbol{\eta} = f(\boldsymbol{x}_1) + f(\boldsymbol{x}_2) + \boldsymbol{\epsilon}$$

- Smooth ANOVA models (Lee and Durban, 2011)

$$\boldsymbol{\eta} = f(\boldsymbol{x}_1) + f(\boldsymbol{x}_2) + f(\boldsymbol{x}_1, \boldsymbol{x}_2) + \boldsymbol{\epsilon}$$

- For higher dimensions one may incur in the **curse of dimensionality** (computational problems)
- Other approaches includes Thin plate splines (radial basis functions). Problems: knots selection and position in larger dimensions.
- Recommended approach use for interactions: **Tensor Products**

GAMs with P-splines

- Use B-splines $\boldsymbol{\eta} = f(x_1 + f(x_2) = B_1\theta_1 + B_2\theta_2$
- Vectorize and combine

$$\boldsymbol{\eta} = [B_1 : B_2] \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \boldsymbol{B\theta}$$

- Difference penalty is blockdiagonal, i.e.

$$\boldsymbol{P} = \text{bdiag}(\boldsymbol{P}_1, \boldsymbol{P}_2) = \begin{bmatrix} \lambda_1 D_1' D_1 & \\ & \lambda_2 D_2' D_2 \end{bmatrix}$$

- In general $\lambda_1 \neq \lambda_2$ (Anisotopic smoothing)

Two-dimensional smoothing with P-splines

- Use B-splines $\boldsymbol{\eta} = f(x_1 + f(x_2) = B_1\theta_1 + B_2\theta_2$
- Vectorize and combine

$$\boldsymbol{\eta} = [B_1 : B_2] \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \boldsymbol{B\theta}$$

- Difference penalty is blockdiagonal, i.e.

$$\boldsymbol{P} = \text{bdiag}(\boldsymbol{P}_1, \boldsymbol{P}_2) = \begin{bmatrix} \lambda_1 D_1' D_1 & \\ & \lambda_2 D_2' D_2 \end{bmatrix}$$

- In general $\lambda_1 \neq \lambda_2$ (Anisotopic smoothing)

$2d$ P-splines

Bivariate smoothing

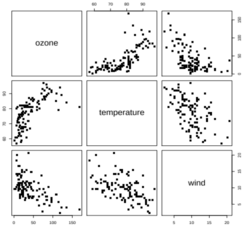
- Bivariate data, $(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{y})$, regression model $\mathsf{E}[\boldsymbol{y}|\boldsymbol{x}_1, \boldsymbol{x}_2] = f(\boldsymbol{x}_1, \boldsymbol{x}_2)$
- B-splines on $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ domains, i.e. $B_k(x_1)$ and $B_l(x_2)$
- We aim to build a **surface** as a **sum of Tensor products**

$$f(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sum_k \sum_l B_k(x_1) B_l(x_2) \theta_{kl}$$

- Now we have a matrix of coefficients $\boldsymbol{A} = [\theta_{kl}]$, to be penalized with λ_1 and λ_2

E.g.: Air quality data in NYC

- Covariates:
  - $\boldsymbol{x}_1 =$ Daily max temperature (in F)
  - $\boldsymbol{x}_2 =$ Wind speed (in mph)



$2d$ P-splines

Bivariate smoothing

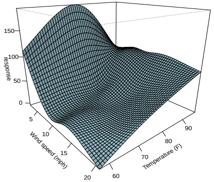
- Bivariate data, $(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{y})$, regression model $\mathsf{E}[\boldsymbol{y}|\boldsymbol{x}_1, \boldsymbol{x}_2] = f(\boldsymbol{x}_1, \boldsymbol{x}_2)$
- B-splines on $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ domains, i.e. $B_k(x_1)$ and $B_l(x_2)$
- We aim to build a **surface** as a **sum of Tensor products**

$$f(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sum_k \sum_l B_k(x_1) B_l(x_2) \theta_{kl}$$

- Now we have a matrix of coefficients $\boldsymbol{A} = [\theta_{kl}]$, to be penalized with λ_1 and λ_2

E.g.: Air quality data in NYC

- Covariates:
  - $\boldsymbol{x}_1 =$ Daily max temperature (in F)
  - $\boldsymbol{x}_2 =$ Wind speed (in mph)



Multidimensional smoothing

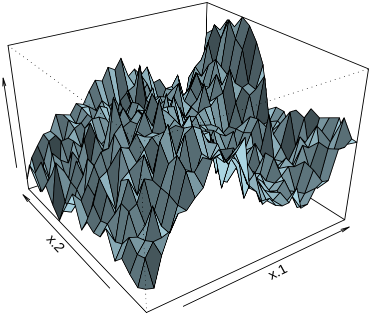
Array structured data

- **Data** $\boldsymbol{Y} = \boldsymbol{y}_{ij}$, $i = 1, \ldots, n_1$ and $j = 1, \ldots, n_2$
- **Array structure**: n_1 rows and n_2 columns

$$\boldsymbol{Y} = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1n_2} \\ y_{21} & y_{22} & \cdots & y_{2n_2} \\ \vdots & \vdots & \ddots & \vdots \\ y_{n_1 1} & \cdots & \cdots & y_{n_1 n_2} \end{bmatrix}$$

- **Regressors:**

$$\boldsymbol{x}_1 = (x_{11}, \cdots, x_{1n_1})'$$
$$\boldsymbol{x}_2 = (x_{21}, \cdots, x_{2n_2})'$$



- **e.g.:** image data, mortality life tables, micro-arrays etc ...

Multidimensional smoothing

Array structured data

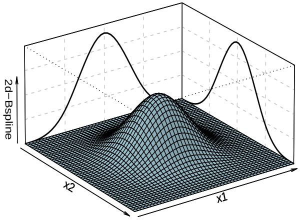
– **Tensor Products of B-splines**

- **Marginal Basis:**
  - $\boldsymbol{B}_1 = \boldsymbol{B}_1(\boldsymbol{x}_1)$, $n_1 \times c_1$
  - $\boldsymbol{B}_2 = \boldsymbol{B}_2(\boldsymbol{x}_2)$, $n_2 \times c_2$
- $2d$ **B-splines Basis:**
  - Kronecker Product ($\otimes$) of marginal basis:

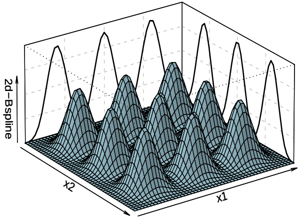
$$\boldsymbol{B} = \boldsymbol{B}_2 \otimes \boldsymbol{B}_1, \quad n_1 n_2 \times c_1 c_2$$

- Computationally efficient methods **(GLAM, Currie, Durban and Eilers, 2006)**

Tensor product of 2 cubic B-splines



B-spline basis of 3×3



References

[1] Hastie T.J. and Tibshirani, R.J. (1990) *Generalized Additive Models* Chapman Hall/CRC Monographs on Statistics Applied Probability

[2] Eilers P.H.C., and Marx B.D. (1996) *Flexible Smoothing with B-splines and Penalties* Statistical Science, Vol. 11, No. 2., pp. 89-102

[3] Lang, S. and Brezger, A. *Bayesian P-Splines* Journal of Computational and Graphical Statistics. Vol 13, 2004, pages 183-212

[4] Lee, D.-J. and Durbán, M. (2011) *P-spline ANOVA-type interaction models for spatio-temporal smoothing* Statistical Modelling, Vol. 11, Issue 1, Pages 49-69.

[5] Ruppert, D., Wand, M.P. and Carroll, R.J. (2009) *Semiparametric regression during 2003â2007* Electron. J. Statist. Volume 3 (2009), 1193-1256.

[6] Currie, ID., Durbán M. and Eilers, PHC. (2006) *Generalized linear array models with applications to multidimensional smoothing* JRSSB, 68:1-22

Penalized splines in R

What's next?

http://idaejin.github.io/bcam-courses/jjseb3/R-jjseb3.html