Smoothing splines

Statistical Learning
CLAMSES - University of Milano-Bicocca

Aldo Solari

References

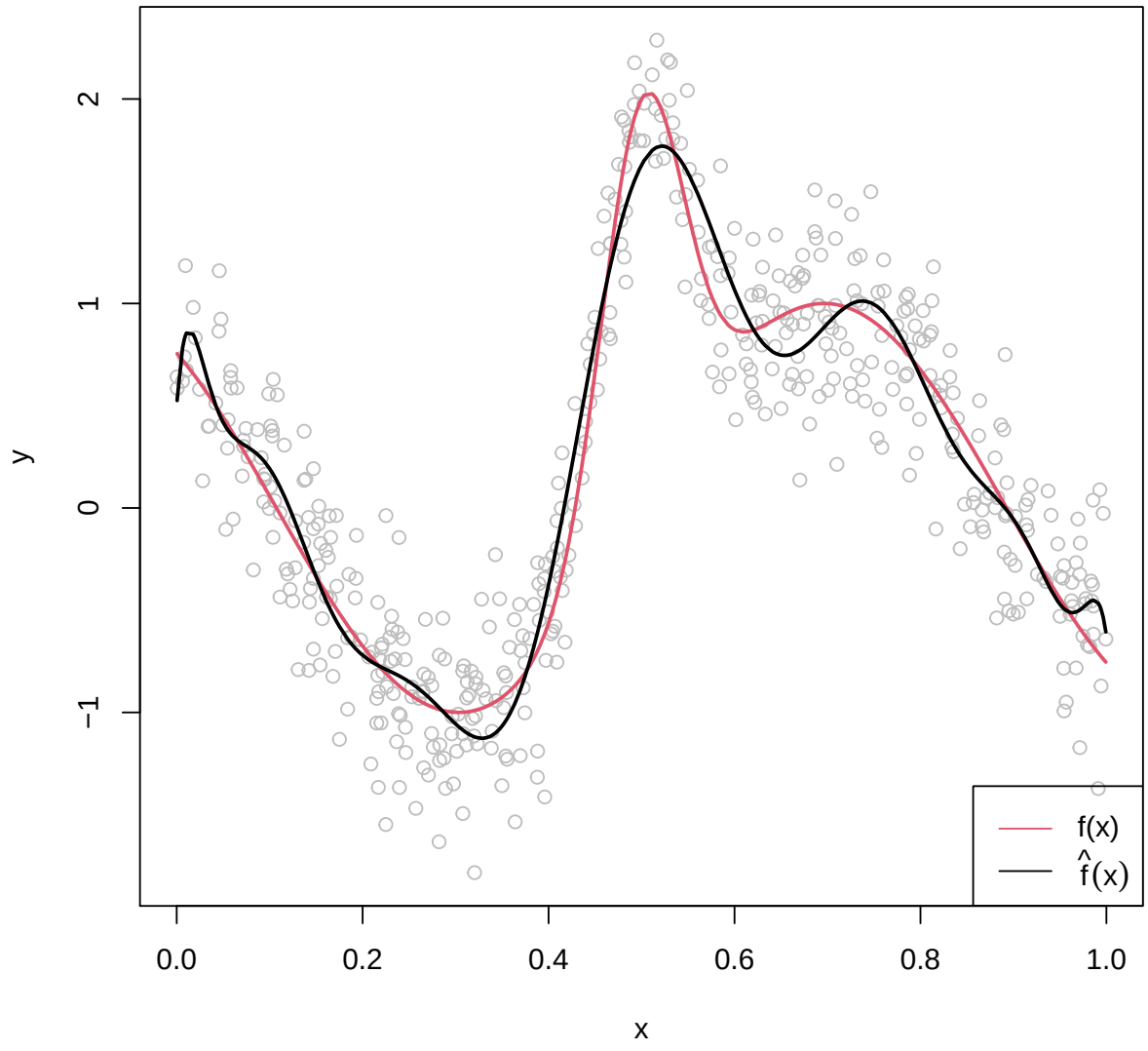
- Eilers, Marx (1996). Flexible smoothing with B-splines and penalties. Statistical science, 11(2), 89–121.
- https://psplines.bitbucket.io/Support/WhyPsplines.pdf

Nonlinearity

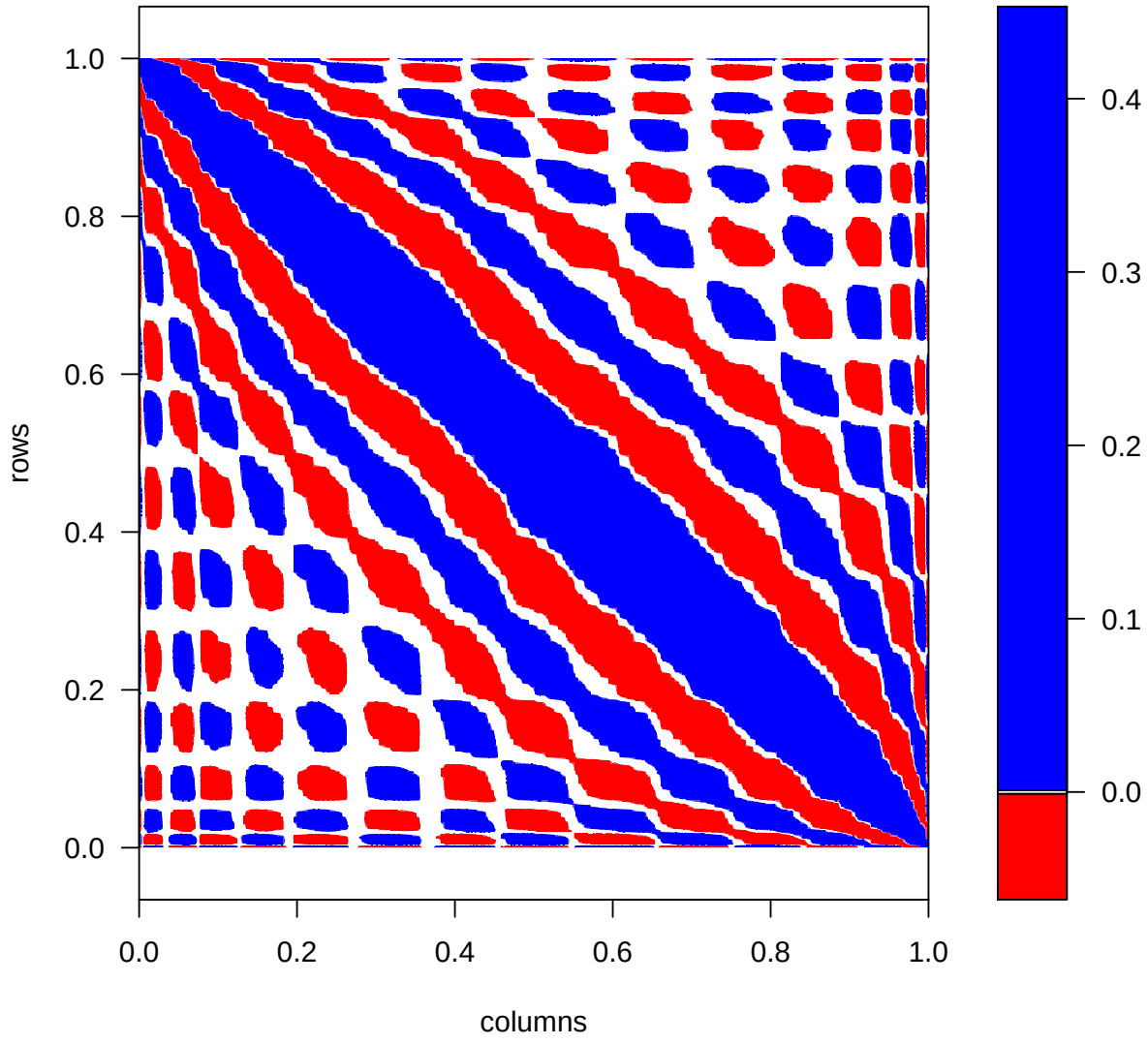
- In many situations, the relationship between x and y is not linear
- Suppose $y = f(x) + \varepsilon$ with

$$f(x) = \sin(2(4x - 2)) + 2e^{-(16^2)(x-.5)^2}$$

- Even with a polynomial of degree 15, the fit is fairly poor in many areas, and 'wiggles' in some places where there doesn't appear to be a need to



Polynomial regression of degree 15



Hat matrix $X(X^tX)^{-1}X^t$ for polynomial regression of degree 15

Piecewise polynomial

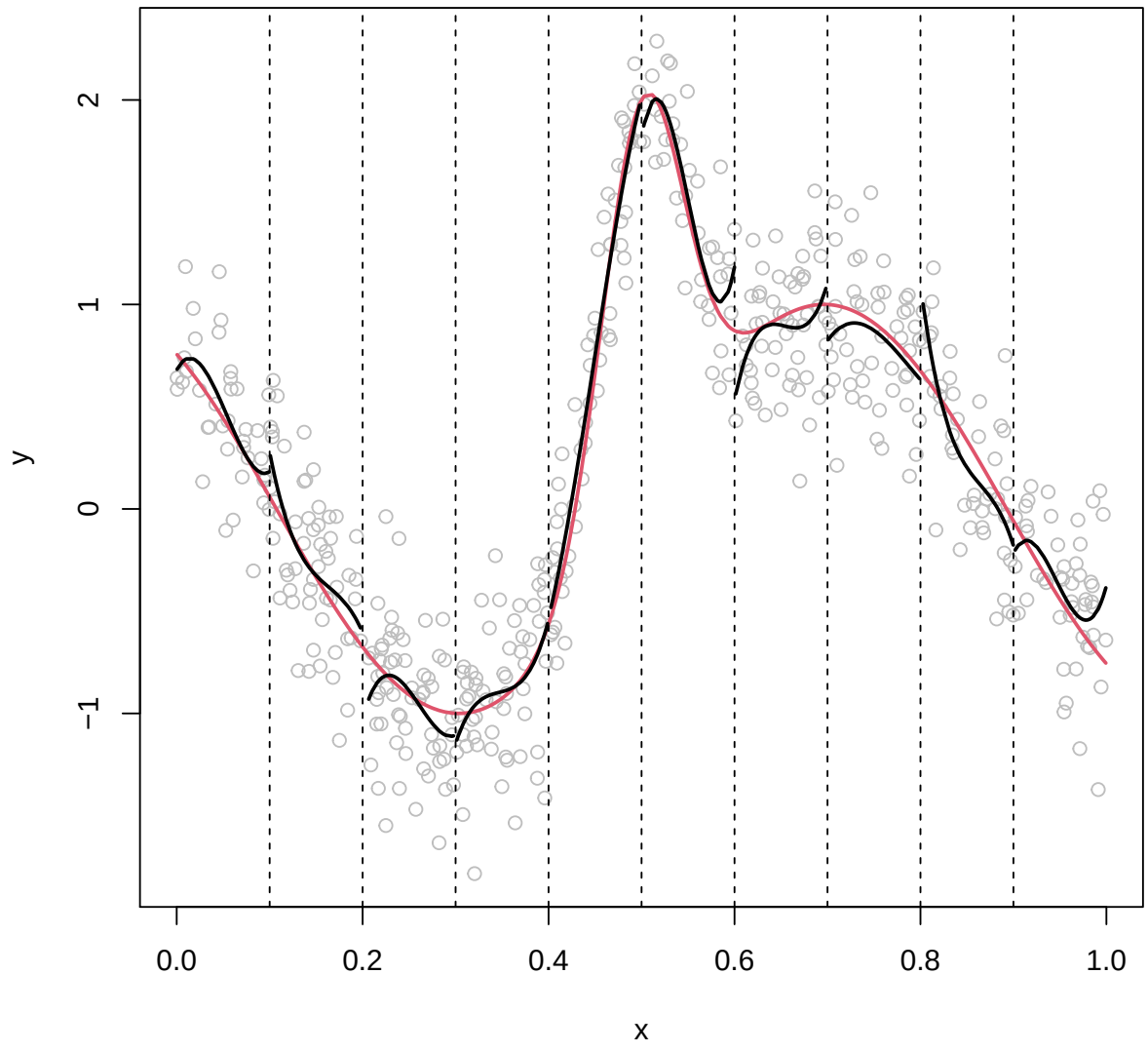
- Divide the data into chunks at various points (*knots*), and fit a polynomial model within that subset of data
- Specify K *internal knots* on the range of x:

$$\min(x) < \xi_1 < \ldots < \xi_K < \max(x)$$

  which define $K + 1$ intervals
- Fit a polynomial model of degree M on each of the $K + 1$ intervals

$$(-\infty, \xi_1], (\xi_1, \xi_2], \ldots, (\xi_{K-1}, \xi_K], (\xi_K, +\infty)$$

- A shortcoming is that at each knot the predicted values will generally not be continuous



Piecewise cubic regression

Basis expansion

$$f(x) = \sum_{j=1}^{p} \beta_j B_j(x)$$

where $B_j(\cdot)$ are known functions called *basis functions*. For example

- 3rd degree polynomial:

$$B_1(x) = 1, B_2(x) = x, B_3(x) = x^2, B_4(x) = x^3$$

- Step function with K knots:

$$B_1(X) = \mathbb{1}\{x < \xi_1\}, B_2(x) = \mathbb{1}\{\xi_1 \leq x < \xi_2\}, \ldots,$$
$$B_K(x) = \mathbb{1}\{\xi_{K-1} \leq x < \xi_K\}, B_{K+1}(x) = \mathbb{1}\{x \geq \xi_K\}$$

Regression splines

Regression splines

A *spline* of degree M with knots $\xi_1, \ldots, \xi_K$

- is a polynomial of degree M on each of the intervals

$$(-\infty, \xi_1], [\xi_1, \xi_2], [\xi_2, \xi_3], \ldots, [\xi_{K-1}, \xi_K], [\xi_K, \infty)$$

- it has continuous derivatives of orders $0, \ldots, M-1$ at each knot

$$f(\xi_k^-) = f(\xi_k^+), \quad \ldots, \quad f^{(M-1)}(\xi_k^-) = f^{(M-1)}(\xi_k^+), \quad k = 1, \ldots, K$$

where ξ_k^+ and ξ_k^- indicate indicate the left and right limits of the function at ξ_k

Truncated power basis

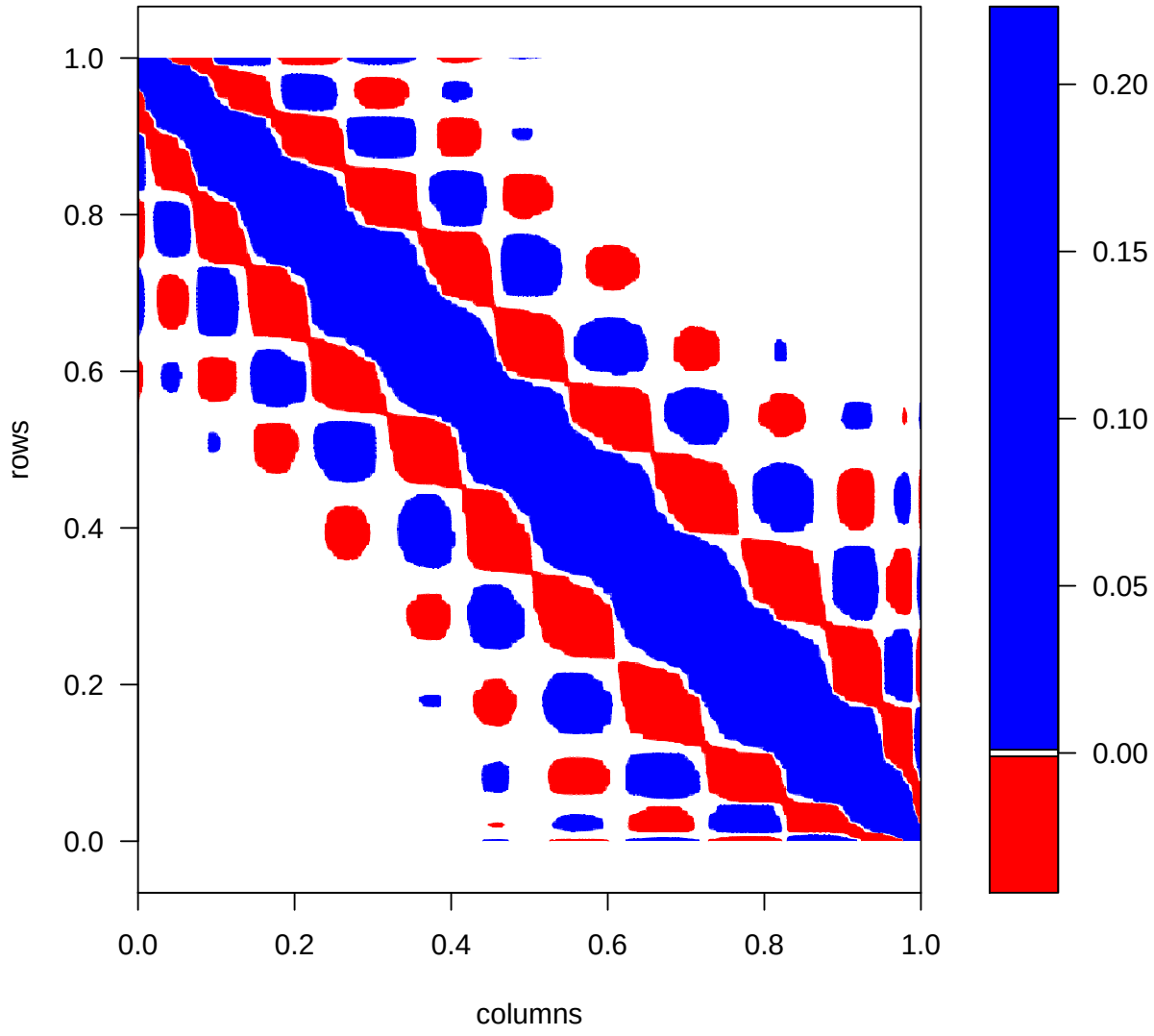
- A spline of degree M with knots $\xi_1, \ldots, \xi_K$ can be defined by the *truncated power basis*

$$\begin{aligned} B_1(x) &= 1 \\ B_{j+1}(x) &= x^j, \quad j = 1, \ldots, M \\ B_{M+k+1}(x) &= (x - \xi_k)_+^M, \quad k = 1, \ldots, K \end{aligned}$$

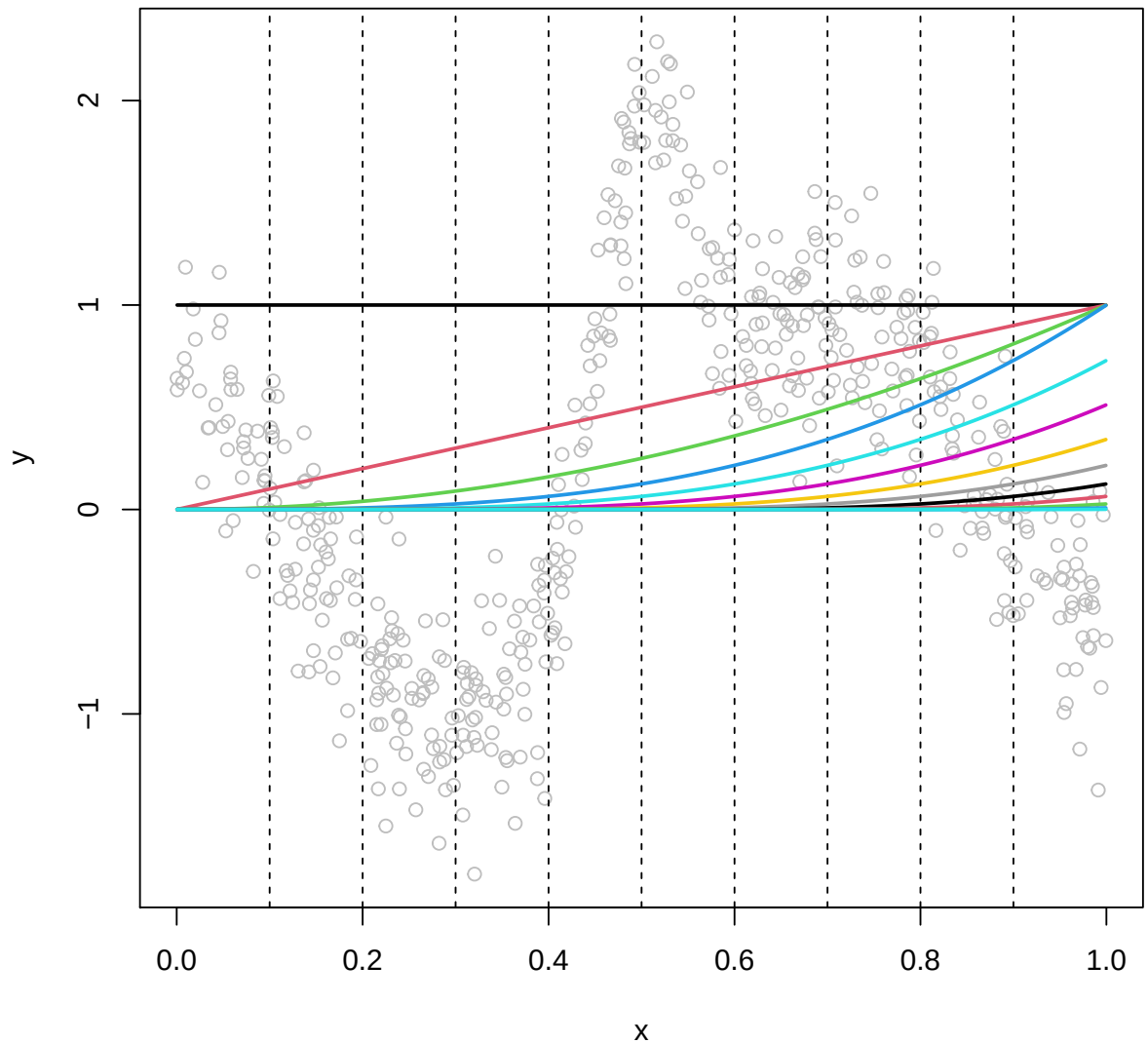
  where $(\cdot)_+$ defines the positive portion of its argument, i.e.

$$(x - \xi_k)_+^M = \begin{cases} (x - \xi_k)^M & x \geq \xi_k \\ 0 & \text{otherwise} \end{cases}$$

- There are $K + 1$ polynomials of order M and K sets of M constraints; the truncated power basis has $(K + 1)(M + 1) - KM$, or $1 + M + K$ free parameters.



Smoothing matrix $B(B^tB)^{-1}B^t$ for polynomial regression of degree 15



Truncated power basis $B_j(x)$

- We compute the solution by explicitly constructing a $n \times (1 + M + K)$ design matrix B with

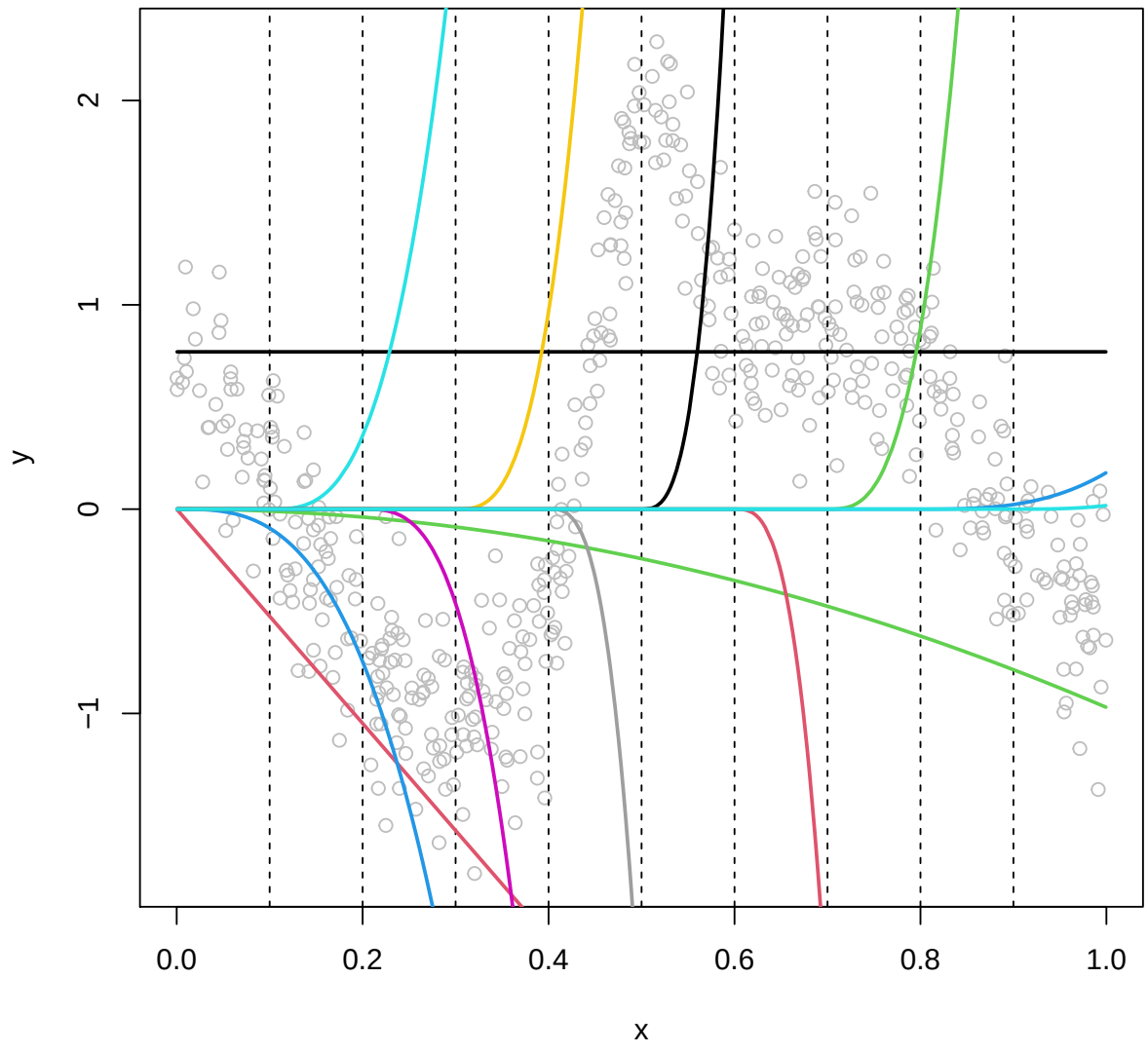
$$B_{i,j} = B_j(x_i) \qquad i = 1, \ldots, n, \quad j = 1, \ldots, 1 + M + K$$

- The regression spline can be written as

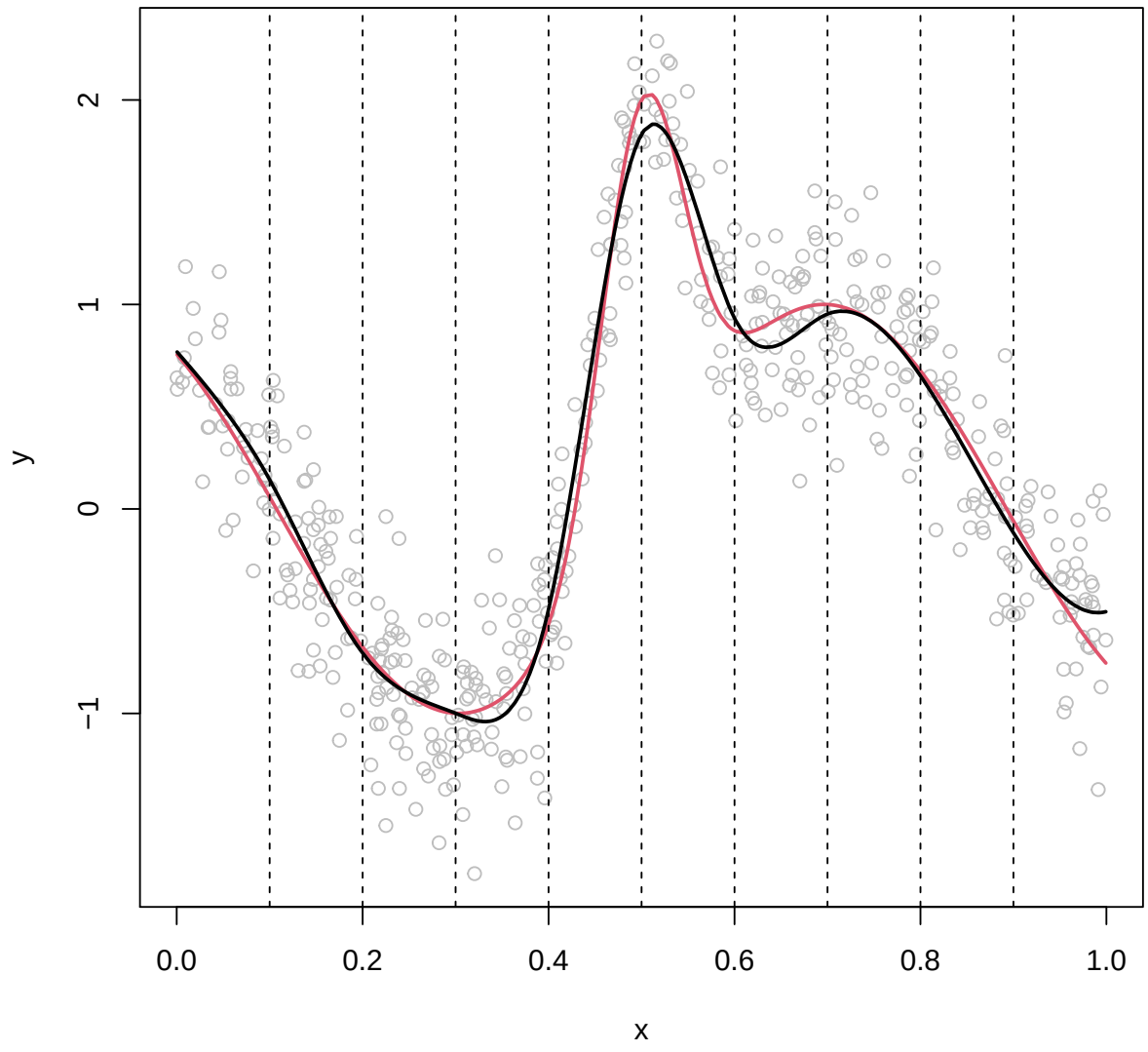
$$\hat{f}(x) = \sum_{j=1}^{1+M+K} \hat{\beta}_j B_j(x)$$

where $\hat{\beta}$ is given by

$$\hat{\beta} = (B^t B)^{-1} B^t y$$



Scaled truncated power basis $\hat{\beta}_j B_j(x)$

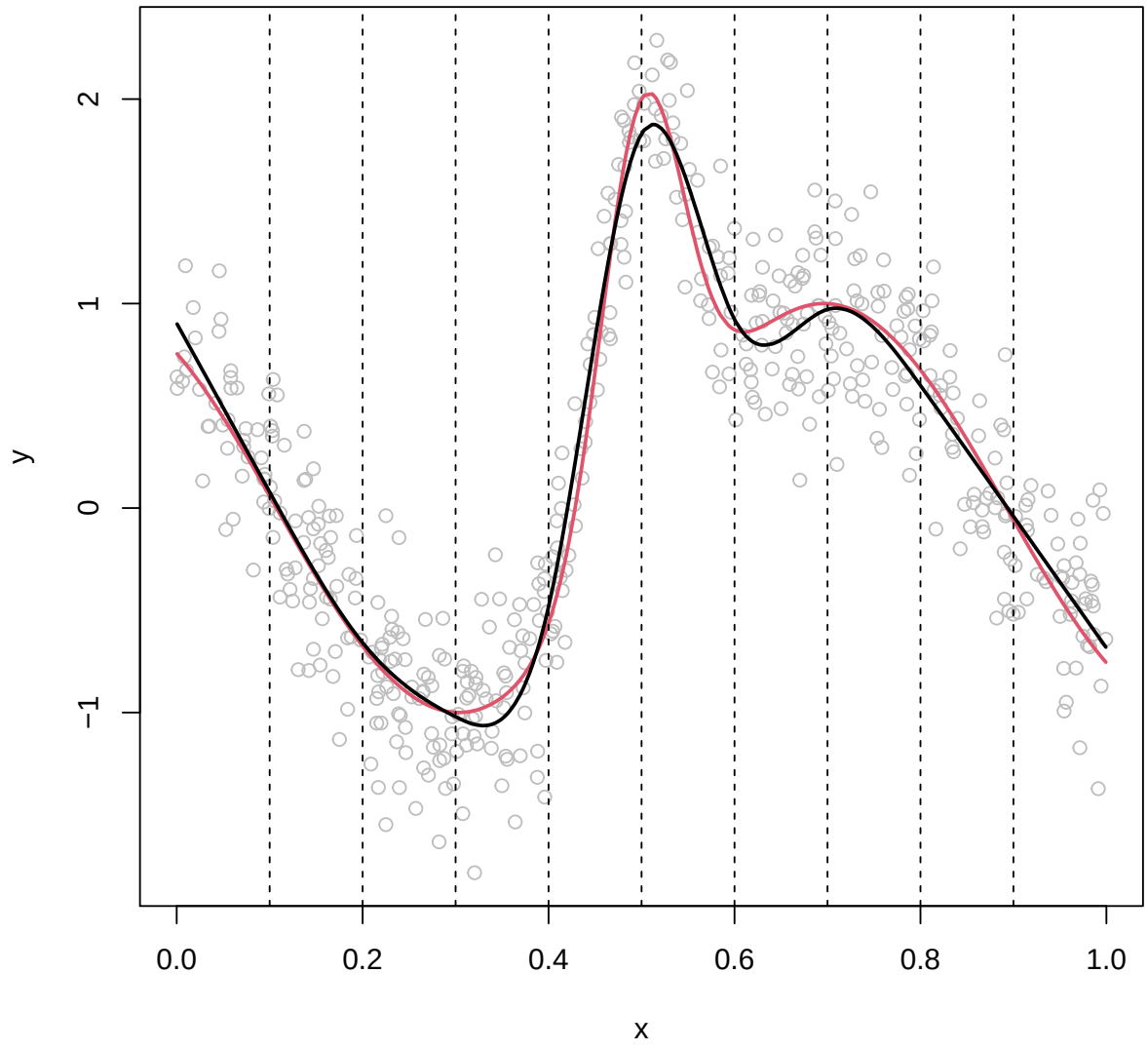


Regression cubic spline

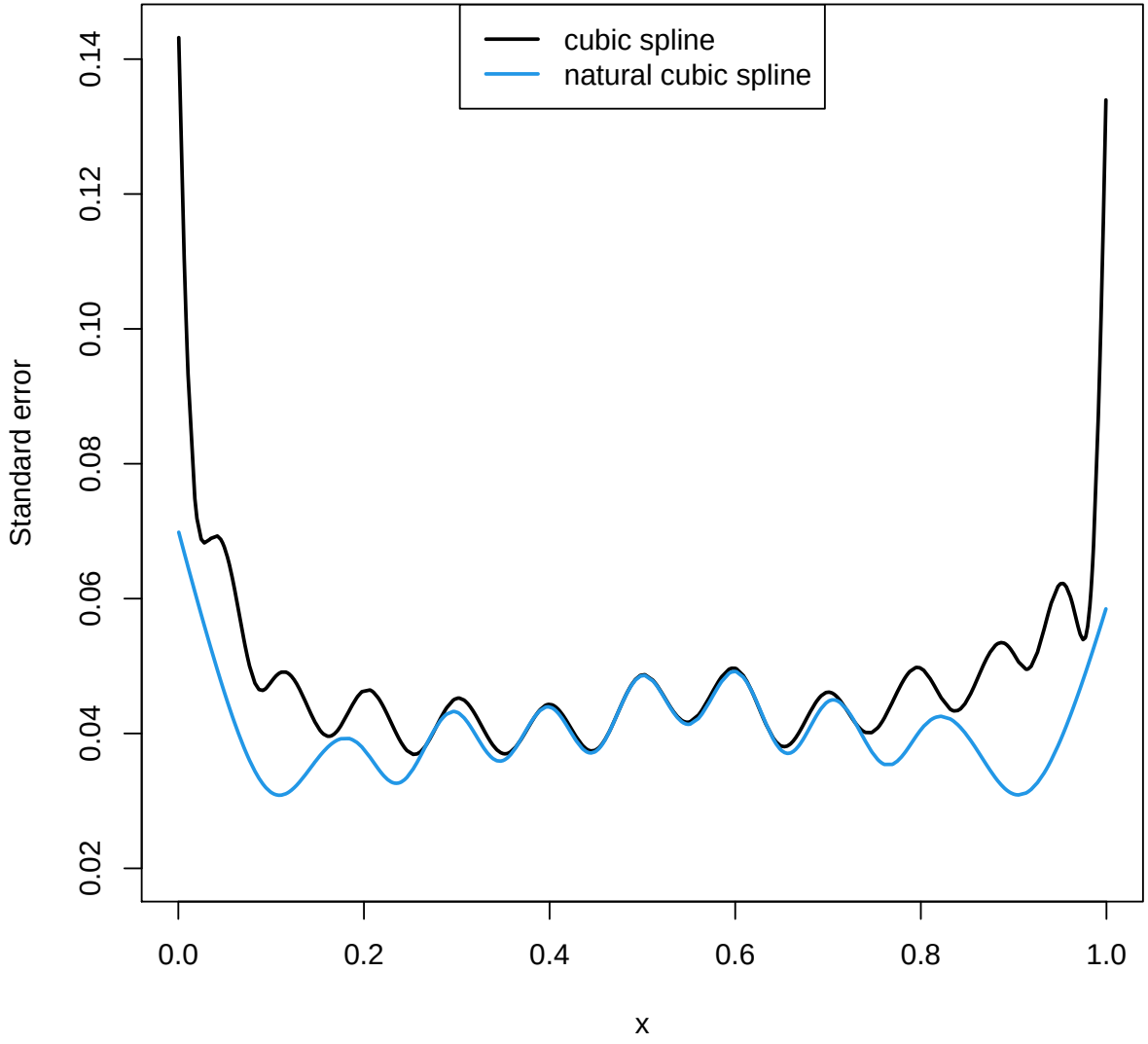
Natural cubic splines

- The most commonly used are cubic splines. Splines exhibit erratic behaviour for values less than ξ_1 and larger than ξ_K
- A natural cubic spline adds additional constraints, namely that the function is linear beyond the boundary knots.
- This frees up 4 degrees of freedom: a natural cubic spline with K knots is represented by K basis functions:

$$
\begin{aligned}
B_1(x) &= 1 \\
B_2(x) &= x \\
B_{j+2}(x) &= \frac{(x-\xi_j)_+^3 - (x-\xi_K)_+^3}{\xi_K - \xi_j} - \frac{(x-\xi_{K-1})_+^3 - (x-\xi_K)_+^3}{\xi_K - \xi_{K-1}} \\
& \qquad j = 1, \ldots, K-2
\end{aligned}
$$



Natural cubic spline



Standard error of the fit

Smoothing splines

Cubic smoothing spline

- Smoothing splines circumvent the problem of knot selection by performing regularized regression over the natural spline basis, placing knots at all inputs $x_1, \ldots, x_n$
- With inputs $x_1 < \ldots < x_n$ contained in an interval $[a, b]$, the cubic smoothing spline estimate is defined as

$$\hat{f} = \arg\min_{f \in C_2} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \int_a^b (f''(x))^2 dx$$

- $f''(x)$ is the second derivative of f with respect to x - it would be zero if f were linear, so this measures the curvature of f at x.
- Remarkably, the minimizer is unique, and is a natural cubic spline with knots at all $x_1, \ldots, x_n$

- The previous result tells us that we can choose natural cubic spline basis $B_1, \ldots, B_n$ with knots $\xi_1 = x_1, \ldots, \xi_n = x_n$ and solve

$$\hat{\beta}_\lambda = \arg\min_\beta \sum_{i=1}^{n} (y_i - \sum_{j=1}^{n} \beta_j B_j(x_i))^2 + \lambda \int_a^b \Big(\sum_{j=1}^{n} \beta_j B_j''(x) \Big)^2 dx$$

  to obtain the smoothing spline estimate $\hat{f}(x) = \sum_{i=1}^{n} \hat{\beta}_j B_j(x)$
- Rewriting

$$\hat{\beta}_\lambda = \arg\min_\beta \|y - B\beta\|^2 + \lambda \beta^t \Omega \beta$$

  where $B_{ij} = B_j(x_i)$ and $\Omega_{jk} = \int B_j''(x) B_k''(x) dx$, shows the smoothing spline problem to be a type of generalized ridge regression problem with solution

$$\hat{\beta}_\lambda = (B^t B + \lambda \Omega)^{-1} B^t y$$

- Fitted values in Reinsch form

$$\begin{aligned}\hat{y} &= B(B^tB + \lambda\Omega)^{-1}B^ty \\ &= (I_n + \lambda K)^{-1}y\end{aligned}$$

  where $K = (B^t)^{-1}\Omega B^{-1}$ does not depend on λ, and $S = (I_n + \lambda K)^{-1}$ is the $n \times n$ *smoothing matrix*

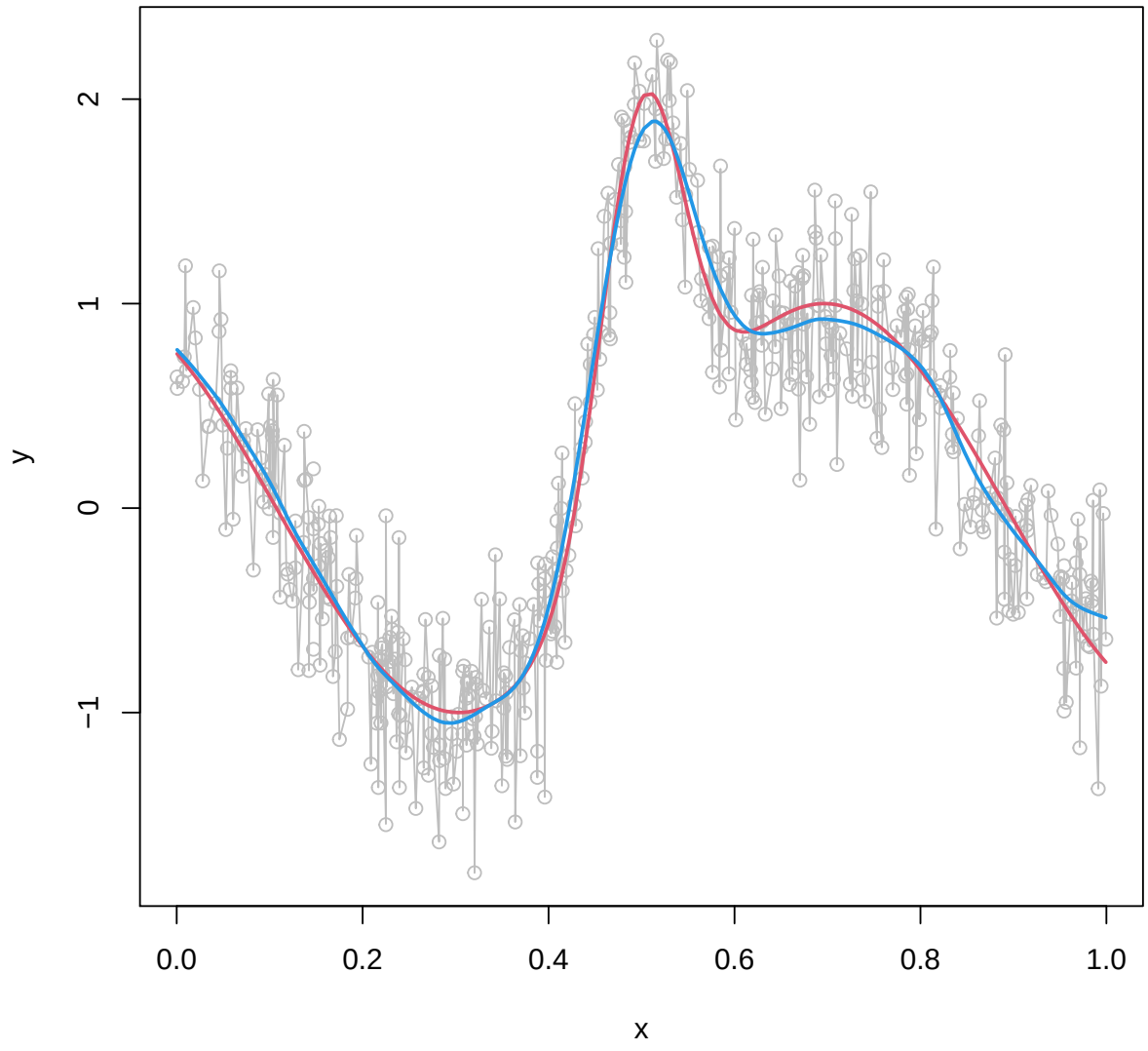
- Leave-one-out cross validation

$$\text{LOO} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1 - S_{ii}}\right)^2$$

- Generalized cross validation

$$\text{GCV} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1 - \text{tr}(S)/n}\right)^2$$

  where $\text{tr}(S)$ is the effective degrees of freedom



`smooth.spline` result with $\lambda = 0$ and 6.9e-15 by LOO

Reinsch original solution

- The original Reinsch (1967) algorithm solves the constrained optimization problem

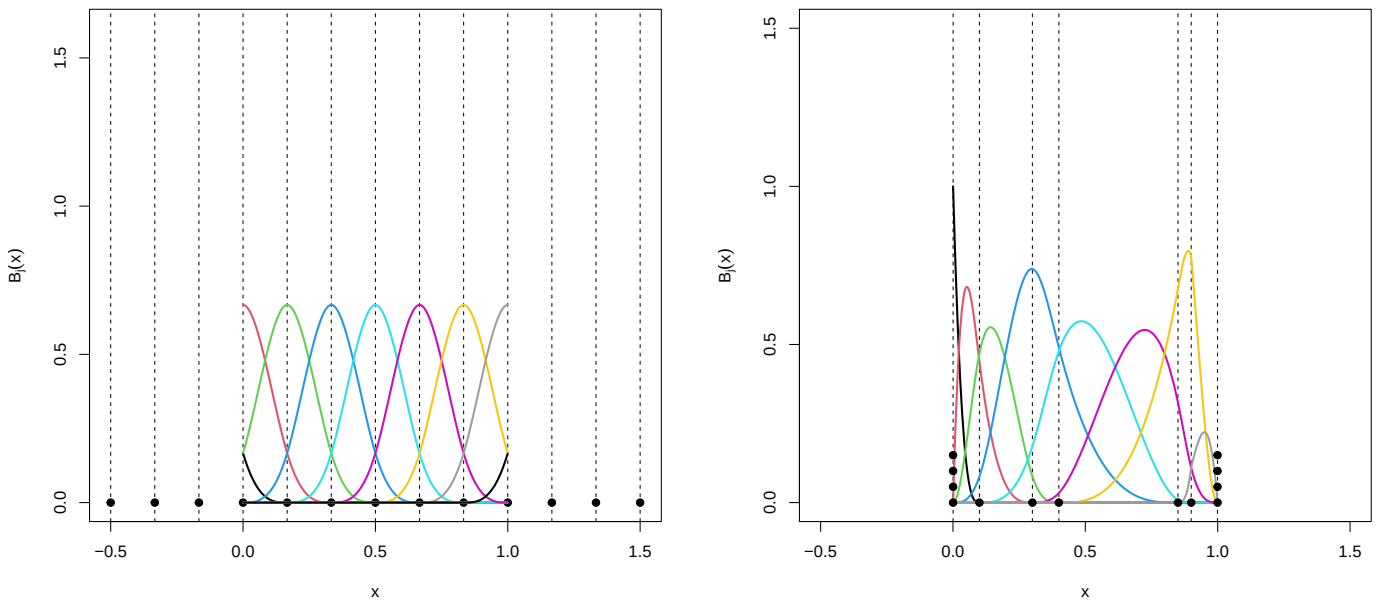
$$\hat{f} = \arg\min_{f \in C_2} \int_a^b (f''(x))^2 dx \text{ such that } \sum_{i=1}^{n} (y_i - f(x_i))^2 \leq c$$

- The previous formulation with a Lagrange parameter on the integral smoothing term instead of the least squares term is equivalent
- See `casl_smspline` implementation in Section 2.6 of CASL

P-splines

B-spline basis

- The truncated power basis suffers from computational issues. The *B*-spline basis is a re-parametrization of the truncated power basis spanning an equivalent space
- The appearance of *B*-splines depends on their knot spacing, e.g.
  - uniform *B*-splines on equidistant knots;
  - non-uniform *B*-splines on unevenly spaced knots and repeated boundary;



Left plot: uniform cubic B-splines with equidistant knots

Right plot: non-uniform cubic B-splines with unevenly spaced knots and duplicated boundary knots

B-spline basis

- B-splines can be computed as differences of truncated power functions
- The general formula for equally-spaced knots is

$$B_j(x) = \frac{(-1)^{M+1}\Delta^{M+1} f_j(x, M)}{h^M M!}$$

satisfying

$$\sum_j B_j(x) = 1$$

where $f_j(x, M) = (x - \xi_j)_+^M$, h is is the distance between knots and Δ^O is the Oth order difference with
$\Delta f_j(x, M) = f_j(x, M) - f_{j-1}(x, M)$,
$\Delta^2 f_j(x, M) = \Delta(\Delta f_j(x, M)) = f_j(x, M) - 2f_{j-1}(x, M) + f_{j-2}(x, M)$

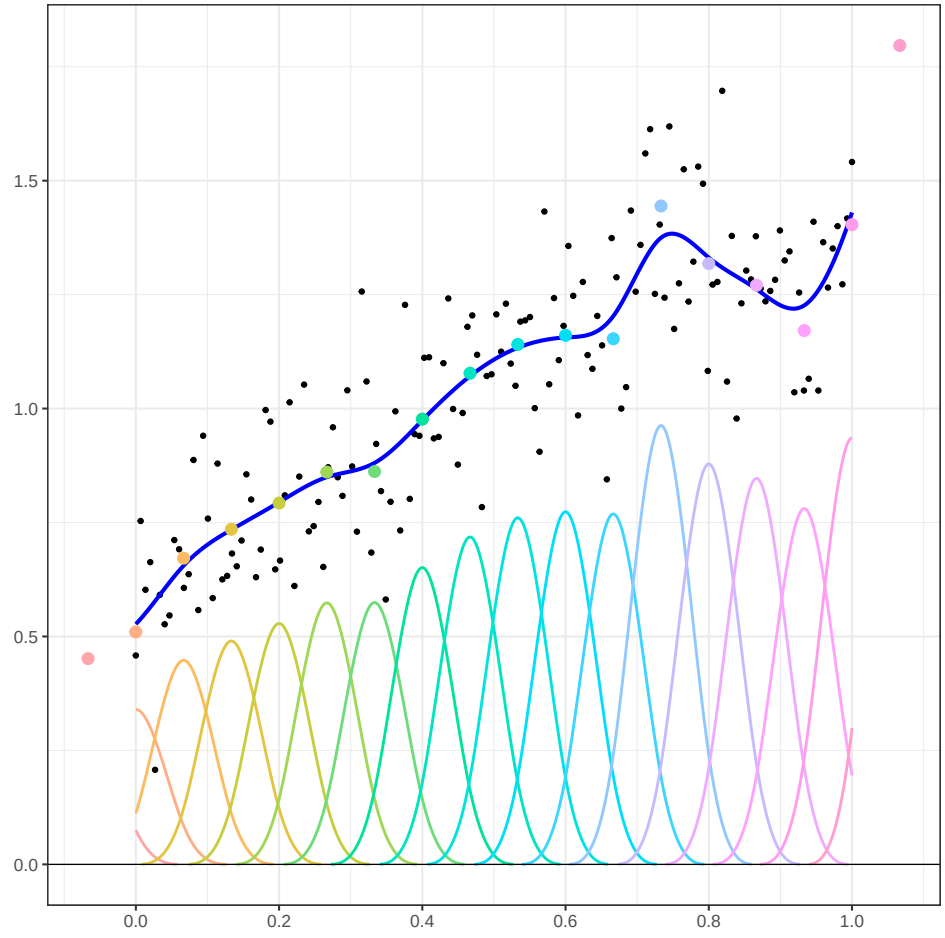
P-splines

- There is an intermediate solution between regression and smoothing splines, proposed more recently by Eilers and Marx (1996)
- P-splines use a basis of (quadratic or cubic) B-splines, B, computed on x and using equally-spaced knots. Minimize

$$\|y - B\beta\|^2 + \lambda\|D\beta\|^2$$

  where $D = \Delta^O$ is the matrix of Oth order differences, with $\Delta\beta_j = \beta_j - \beta_{j-1}$, $\Delta^2\beta_j = \Delta(\Delta\beta_j) = \beta_j - 2\beta_{j-1} + \beta_{j-2}$ and so on for higher O. Mostly $O = 2$ or $O = 3$ is used.
- Minimization leads to the system of equations

$$(B^tB + \lambda D^tD)\hat{\beta} = B^ty$$



The core idea of *P*-splines: a sum of B-spline basis functions, with gradually changing heights. The blue curve shows the *P*-spline fit, and the large dots the *B*-spline coefficients. R code in `f-ps-show.R`

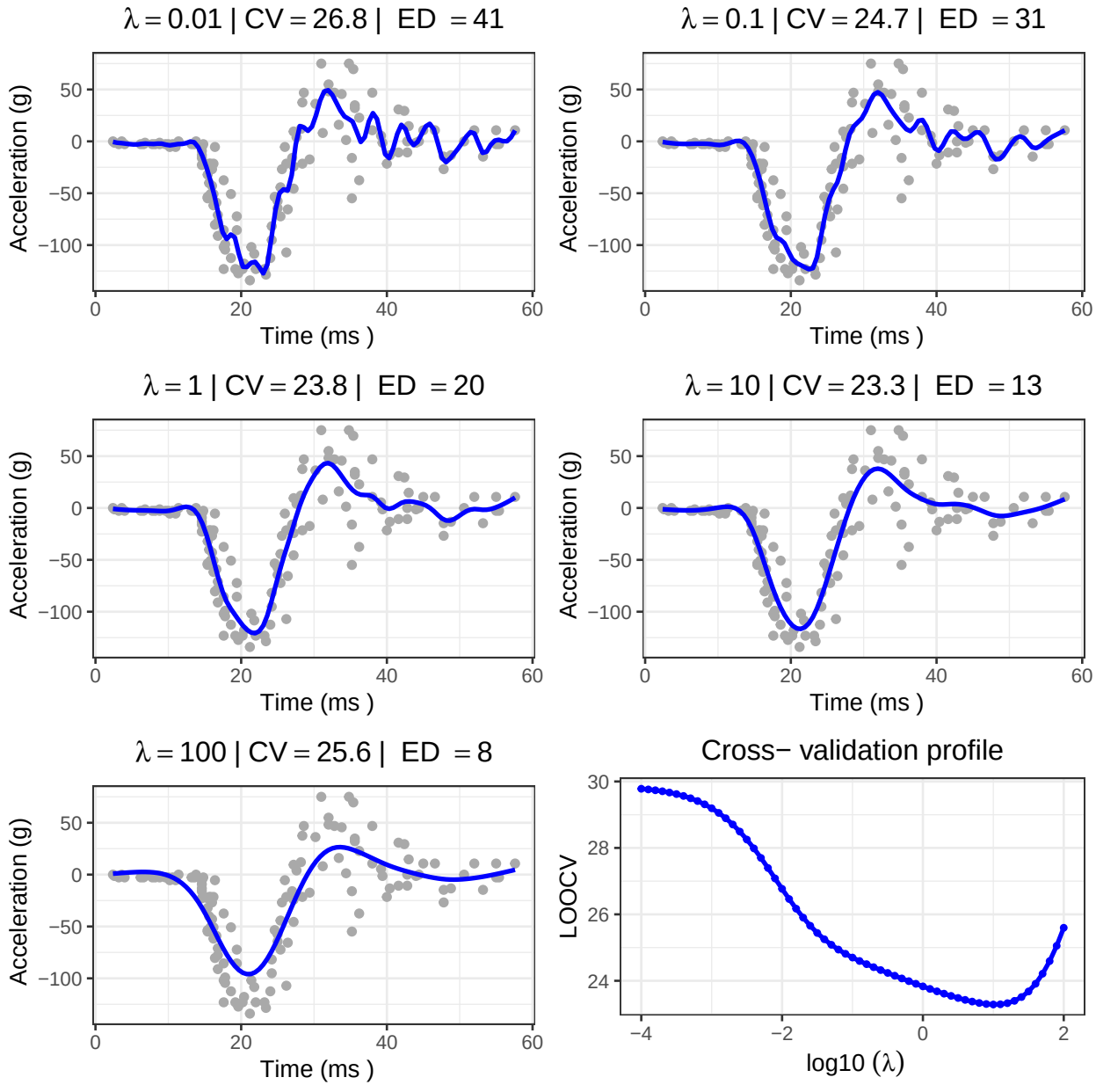
Cross-validation

- We have that $\hat{y} = B(B^tB + \lambda D^tD)^{-1}B^ty = Sy$
- $$\text{LOO} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i^{(-i)})^2 = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{1 - S_{ii}}\right)^2$$
- $$\text{GCV} = \frac{1}{n}\sum_{i=1}^{n}\frac{(y_i - \hat{y}_i)^2}{(1 - \text{tr}(S)/n)^2}$$
- We can compute the trace of R without actually computing its diagonal, using

  $$\text{tr}(S) = \text{tr}((B^tB + P)^{-1}B^tB) = \text{tr}(I_n - (B^tB + P)^{-1}P)$$

  where $P = \lambda D^tD$

mcycle


λ = 0.01 | CV = 26.8 | ED = 41
λ = 0.1 | CV = 24.7 | ED = 31
λ = 1 | CV = 23.8 | ED = 20
λ = 10 | CV = 23.3 | ED = 13
λ = 100 | CV = 25.6 | ED = 8
Cross− validation profile
Acceleration (g)
Time (ms)
LOOCV
log10 (λ)