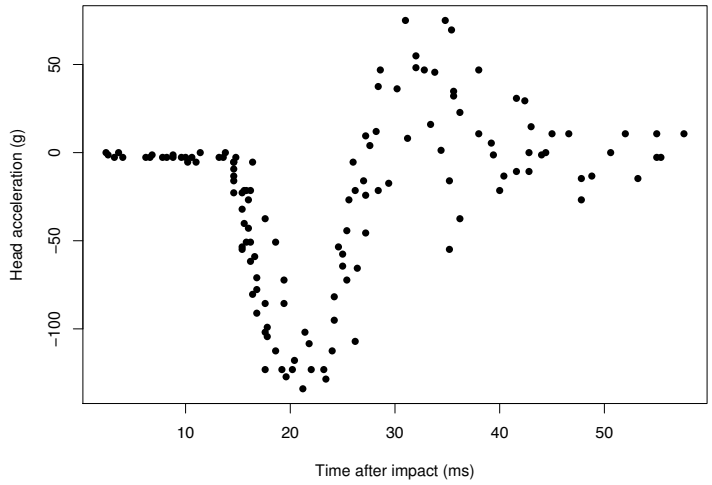
Semiparametric regression

- Normal linear model has two main aspects:
  - **systematic variation**, $\mathrm{E}(y) = \mu$, and $\mu = X\beta$ with parameters β;
  - **stochastic variation**, $y \sim \mathcal{N}_n(\mu, \sigma^2 I_n)$.
- Can relax the stochastic assumption using other distributions or second-order assumptions, but still have parametric model for the systematic part.
- Often want to relax systematic part for more flexible models, for
  - exploratory data analysis — 'will a linear model be adequate?'
  - confirmatory data analysis — 'I've fitted a linear model, is it adequate?'
  - general modelling — 'the data are too complex to expect a simple parametric model to work, so what can I do?'
  - semiparametric modelling — 'I will use a parametric model for the effects of interest, but can I model nuisance effects more flexibly?'
- Most basic tool is the **scatterplot smoother**.

Example: Motorcycle data

Measurements of head acceleration (g) at time after impact (ms) in a simulated motorcycle accident, used to test crash helmets:



Scatterplot smoothing

- Have data $(x_1, y_1), \ldots, (x_n, y_n)$, with $x_- \leq x_1 < \cdots < x_n \leq x_+$ (ahem) and we wish to estimate $\mathrm{E}(y) = \mu(x)$, for $x \in \mathcal{X} = [x_-, x_+]$.
- Suppose that $\mu \in \mathcal{M}$, a function space spanned by n linearly independent basis functions that can be identified by evaluation at $x_1, \ldots, x_n$, and let $\mu_j = \mu(x_j)$.
- Can choose a basis $\{b_1(x), \ldots, b_n(x)\}$ for $\mathcal{M}$ such that $\mu(x) = \sum_{j=1}^n \mu_j b_j(x)$ interpolates $(x_1, \mu_1), \ldots, (x_n, \mu_n)$.
- Suppose that $\mathcal{M}$ contains the linear functions on $\mathcal{X}$ and that the second derivatives of the $b_j(x)$ are not all zero, so functions in $\mathcal{M}$ may also be nonlinear in x.
- To estimate μ we minimise a **penalised sum of squares**,

$$\sum_{j=1}^{n} \{y_j - \mu(x_j)\}^2 + \lambda \int_{\mathcal{X}} \{\mu''(x)\}^2 \, \mathrm{d}x, \tag{22}$$

  where the **roughness penalty** imposes smoothness: if $\lambda \to 0$, then $\mu(x_j) \to y_j$ and $\widehat{\mu}$ interpolates, but when $\lambda \to \infty$ even tiny wiggles in μ will give a huge penalty, making $\widehat{\mu}$ linear.
- The penalty does not affect linear functions, so $\mathcal{M} = \mathcal{L} \bigoplus \mathcal{P}$, where $\mathcal{L}$ and $\mathcal{P}$ are the two-dimensional vector space of linear functions on $\mathcal{X}$ and an $(n-2)$-dimensional vector space of nonlinear functions on $\mathcal{X}$, and $\bigoplus$ denotes addition of vector spaces.

Scatterplot smoothing II

- The roughness term is

$$\int_{\mathcal{X}} \{\mu''(x)\}^2 \, \mathrm{d}x = \int_{\mathcal{X}} \left\{ \sum_{j=1}^{n} \mu_j b_j''(x) \right\}^2 \mathrm{d}x = \sum_{i,j=1}^{n} \mu_i \mu_j \int_{\mathcal{X}} b_i''(x) b_j''(x) \, \mathrm{d}x = \mu^{\mathrm{T}} S \mu,$$

  say, where $\mu^{\mathrm{T}} = (\mu_1, \ldots, \mu_n)$.
- $S_{n \times n}$ has (i, j) element $\int_{\mathcal{X}} b_i''(x) b_j''(x) \, \mathrm{d}x$ and is symmetric and positive semi-definite of rank $n - 2$, because linear functions are unpenalised, so $S 1_n = S(x_1, \ldots, x_n)^{\mathrm{T}} = 0$.
- The penalised sum of squares

$$(y - \mu)^{\mathrm{T}}(y - \mu) + \lambda \mu^{\mathrm{T}} S \mu \equiv -2\mu^{\mathrm{T}} y + \mu^{\mathrm{T}}(I_n + \lambda S)\mu,$$

  is minimised by $\widehat{\mu}_\lambda = (I_n + \lambda S)^{-1} y$.
- As λ increases from zero, the fitted value $\widehat{\mu}_\lambda$ shrinks from y towards the straight-line regression fit to y, which is unpenalised.
- The equivalent degrees of freedom are $\mathrm{edf}_\lambda = \mathrm{tr}(H_\lambda) = \sum_{j=1}^{n} (1 + \lambda \delta_j)^{-1}$, where $\delta_1 \geq \cdots \geq \delta_3 > \delta_2 = \delta_1 = 0$ are the eigenvalues of S. As λ increases edf_λ decreases monotonically from $\mathrm{edf}_0 = n$ towards $\mathrm{edf}_\infty = 2$.

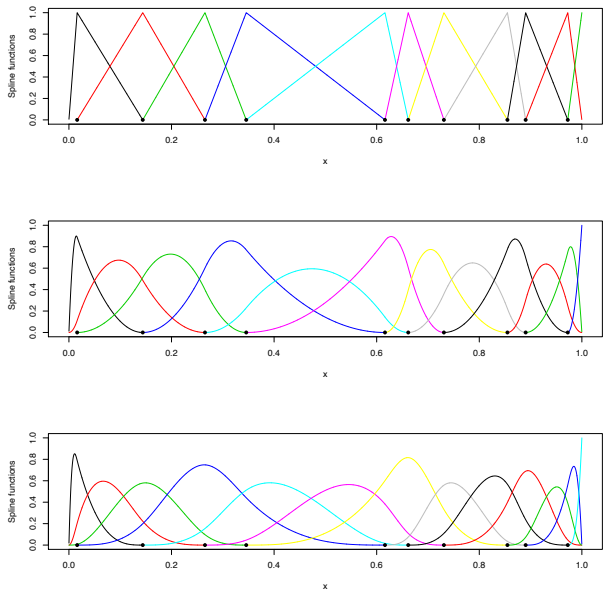
Scatterplot smoothing III

- In principle we might take any basis functions, but in practice we usually take local polynomials known as **splines** that have good approximation properties.
- There are many forms of splines, which
  - are often cubic polynomials with finite support between values of x known as **knots**, $x_1^*, \ldots, x_K^*$, and then S is tri-diagonal,
  - sometimes form a **natural cubic spline**, which has $K = n$ and certain optimality properties,
  - are discussed in more detail later.
- If there is no penalisation ($\lambda = 0$) then we have a standard linear model, and spline basis functions are called **regression splines**.
- Under second-order assumptions we choose λ by minimising $\mathrm{CV}(\lambda)$ or $\mathrm{GCV}(\lambda)$.
- Under normal-theory assumptions we can use REML to estimate σ^2 and λ.
- Obvious generalisation allows weight matrix $W = \mathrm{diag}(w_1, \ldots, w_n)$.
- If the $x_1, \ldots, x_n$ are not unique, write $\mathrm{E}(y) = N_{n \times n'} \mu_{n' \times 1}$ in terms of the means μ at the n' unique elements of x, and minimise

$$(y - N\mu)^{\mathrm{T}} W (y - N\mu) + \lambda \mu^{\mathrm{T}} S \mu.$$

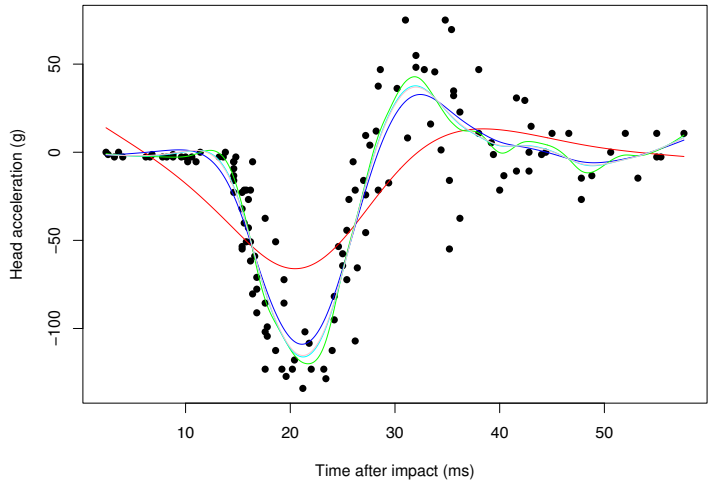
where $S_{n' \times n'}$ arises as before from the roughness penalty on $\mu(x)$.

Linear, quadratic and cubic B-splines



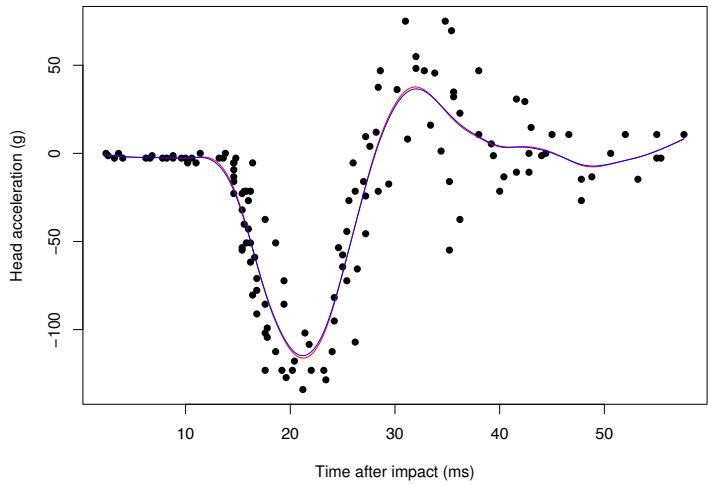
Example: Motorcycle data

Scatterplot smooths based on natural cubic splines with edf equal to 5 (red), 10 (blue), 20 (green), and chosen by CV (cyan, $\mathrm{edf} = 12.8$) and GCV (pink, $\mathrm{edf} = 12.26$):



Example: Motorcycle data

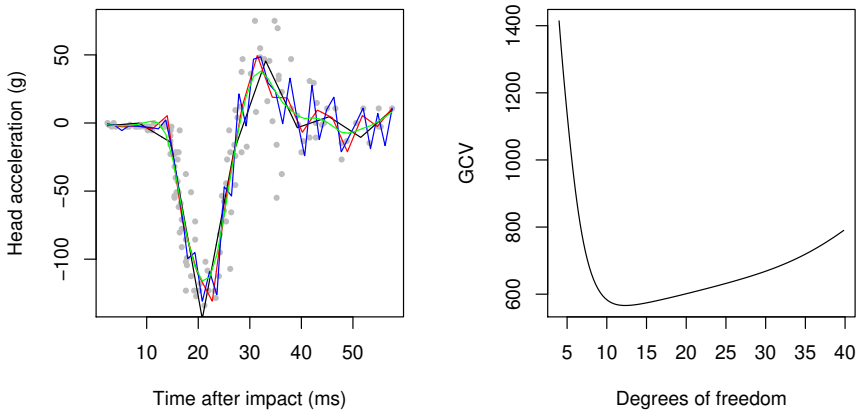
Scatterplot smooths based on natural cubic splines with weights 16 when $x \leq 12$ and 1 for $x > 12$, and edf chosen by CV (red, $\mathrm{edf} = 14.7$) and GCV (blue, $\mathrm{edf} = 13.7$):



Choosing K and λ

- Above we took $K = n$ basis functions, but for statistical purposes we seek a summary of the data, so we hope that $\mathrm{edf} \ll n$, so we hope that $K < n$, maybe even $K \ll n$.
- Theory suggests that as $n \to \infty$ we need $K = O(n^{1/5})$ or even $O(n^{1/9})$ to get near-optimal estimation of $\mu(x)$, when μ lies in reasonable function classes;
- In practice we take K (more than) large enough to give enough flexibility (increasing it if results are suspect, $K = 9$ by default in `mgcv`), and allow λ to determine the smoothness of the curve;
- Typically the knots x_k^* are placed at equally-spaced quantiles of x.

Example: Motorcycle data



- Left: linear spline fits with $\lambda = 0$ and $K = 10$ (black), 20 (red), 40 (blue), and optimal GCV choice of λ with $K = 40$ (green)
- Right: $\mathrm{GCV}(\lambda)$ as a function of df_λ for $K = 40$.

Comments

- We discuss inference (beyond 'point' estimation) and adaptive estimation of weights later ...
- Here we are producing point estimates; later we discuss the construction of confidence sets.
- An alternative local averaging approach uses locally weighted fits, such as the **Nadaraya–Watson estimator**

$$\widehat{\mu}(x) = \frac{\sum_{j=1}^{n} K\{(x - x_j)/h\} y_j}{\sum_{j=1}^{n} K\{(x - x_j)/h\}},$$

  where
  - the **kernel function** K is something like the Gaussian density, and
  - the **bandwidth** h plays a role similar to edf.

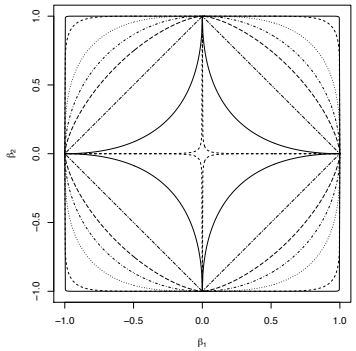
  This is also a linear smoother, and in fact the spline smoothers have representations in terms of equivalent kernels.
- Local averaging can be extended to **local likelihood** fitting of more complex models.

L_q penalties

☐ The quadratic penalty $\|\beta\|_2$ generalises to other L_q penalties

$$\|\beta\|_q = \sum_{r=1}^{p} |\beta_r|^q,$$

shown below for $p = 2$ and (working inwards) $q = 100$, 10, 3, 2, 1.5, 1, 0.5, 0.2; $\|\beta\|_0 = \#\{\beta_r \neq 0\}$ counts the number of non-zero parameters.



(Some picture credits here and later: Simon Wood)

Basic geometry

☐ If $D(\beta)$ is a sum of squares or negative log likelihood, then

$$\tilde{\beta}_\lambda = \text{argmin}_\beta \{D(\beta) + \lambda\|\beta\|_q\},$$

- satisfies $\|\tilde{\beta}_\lambda\|_q = t$ for some t, and
- minimises $D(\beta)$ on that contour, i.e.,

$$\tilde{\beta}_\lambda = \text{argmin}_\beta D(\beta) \quad \text{such that} \quad \|\tilde{\beta}_\lambda\|_q = t,$$

because otherwise we could reduce $D(\beta)$ while leaving the penalty unchanged, i.e., $\tilde{\beta}_\lambda$ would not be optimal.

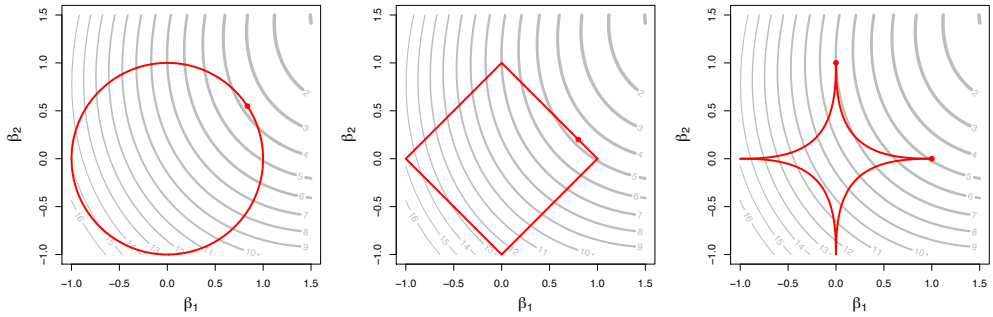
☐ The sets $\|\tilde{\beta}_\lambda\|_q = t$

- have cusps (and thus can set $\beta_r = 0$) when $q \leq 1$,
- are non-convex (and thus may give non-unique solutions) when $q < 1$,

so there is a unique solution if the contours of $D(\beta)$ and $\|\beta\|_q$ are convex, and both a unique solution and the possibility of choosing variables (sparsity) by setting $\beta_r = 0$ when $q = 1$.

Basic geometry II

Penalised solutions (red dots) for $q = 2$, 1, 0.45, with contours of $D(\beta)$ in grey and solution contour for $\|\beta\|_q$ in red.



As $\lambda \to \infty$ the constraint tightens and the red contours shrink around the origin, and as $\lambda \to 0$ the constraint relaxes and the $\tilde{\beta}_\lambda$ tends to the unconstrained estimate.

Lasso

- The **lasso (least absolute shrinkage and selection operator)** objective function can be written as

$$L = \tfrac{1}{2}\|y - X\beta\|_2 + \lambda\|\beta\|_1,$$

so suppose we have minimised this for some λ_0, giving **active set** $A = \{r : \tilde{\beta}_r \neq 0\}$ and

$$L = \tfrac{1}{2}(y - X_A\tilde{\beta}_A)^{\mathrm{T}}(y - X_A\tilde{\beta}_A) + \lambda\sum_{r\in A}|\tilde{\beta}_r|,$$

and now we aim to decrease λ (i.e., to relax the constraint).

- Now $\mathrm{d}|x|/\mathrm{d}x = \mathrm{sign}(x)$, so when

$$\frac{\mathrm{d}L}{\mathrm{d}\tilde{\beta}_A} = X_A^{\mathrm{T}}(X_A\tilde{\beta}_A - y) + \lambda\,\mathrm{sign}(\tilde{\beta}_A) = 0,$$

we have

$$\tilde{\beta}_A = (X_A^{\mathrm{T}}X_A)^{-1}X_A^{\mathrm{T}}y - \lambda(X_A^{\mathrm{T}}X_A)^{-1}\mathrm{sign}(\tilde{\beta}_A) = b - \lambda a,$$

say, i.e., $\tilde{\beta}_A$ is linear in λ until A changes.

- A changes on deleting a column X_r from X_A or on adding one from its complement X_{A^c}.
- $\mathrm{sign}(\tilde{\beta}_A)$ only changes when (say) $\tilde{\beta}_r$ passes through zero, but r leaves A when $\tilde{\beta}_r = 0$.

Lasso algorithm

- A variable in A is deleted if a component of $\tilde{\beta}_A = b - \lambda a$ hits zero as λ decreases from λ_0, which occurs at $\lambda_- = \max_{\lambda<\lambda_0} b_r/a_r$.
- If X_r is the rth column of X, then r will enter A if adding $X_r\beta_r$ decreases L, i.e., if

$$\frac{\mathrm{d}L}{\mathrm{d}\beta_r} = X_r^{\mathrm{T}}(X\beta - y) + \lambda \mathrm{sign}(\beta_r) \quad \begin{cases} < 0, & \beta_r > 0, \\ > 0, & \beta_r < 0, \end{cases}$$

  so β_r remains inactive if $|X_r^{\mathrm{T}}(y - X\beta)| \leq \lambda$.
- Thus as λ decreases, A changes when for some r in the complement A^c of A we have

$$X_r^{\mathrm{T}}(y - X_A\tilde{\beta}_A) = \pm\lambda,$$

  or, setting $\tilde{\beta}_A = b - \lambda a$,

$$X_{A^c}^{\mathrm{T}}(y - X_A b) + \lambda(X_{A^c}^{\mathrm{T}} X_A a \pm 1) = 0 \implies c + \lambda(d \pm 1) = 0,$$

  say: the next variable is added when $\lambda = \lambda_+ = \max_{\lambda<\lambda_0}\{-c_r/(d_r \pm 1)\}$.
- Hence if $s = \mathrm{sign}(\beta)$, the algorithm decreases λ from
  - the highest λ at which the a first variable is active, and defines the A and s, then
  - finds the next λ at which A changes, stores it and the corresponding $\tilde{\beta}$, updating A and s.

Practical matters and thresholding

- Usually
  - λ is chosen by dividing the data into training and testing subsets and minimising some measure of prediction error for the test subset,
  - y is centered and X has no column of ones, and
  - the columns of X are standardized to have zero mean and unit variance — what this means in terms of interpreting the components of β is then unclear!
- We can think of penalised estimators as using different sorts of **thresholding** functions, where $\widehat{\beta}$ is replaced by $\tilde{\beta} = g_\lambda(\widehat{\beta})$ and (conceptually)
  - for the **lasso** there is soft thresholding,

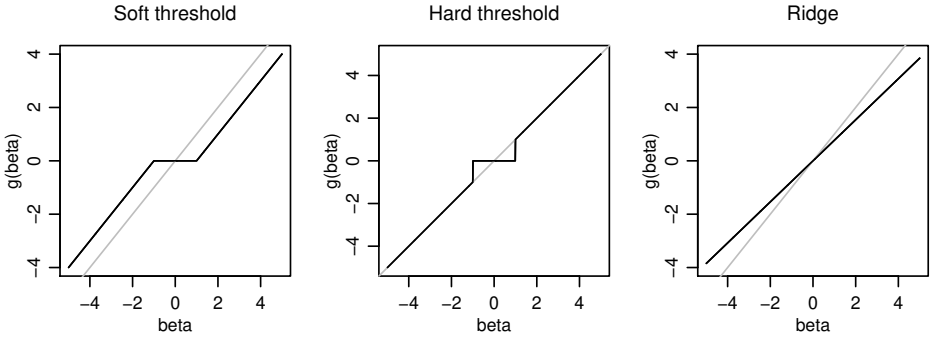
$$g_\lambda(u) = \begin{cases} 0, & |u| < \lambda, \\ \mathrm{sign}(u)(|u| - \lambda), & \text{otherwise}, \end{cases}$$

  - for **variable selection** there is hard thresholding,

$$g_\lambda(u) = \begin{cases} 0, & |u| < \lambda, \\ u, & \text{otherwise}, \end{cases}$$

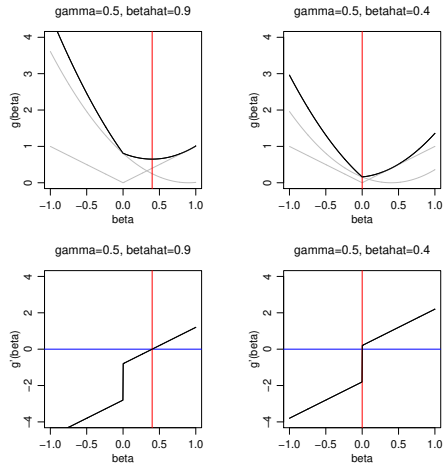
  - for **ridge regression** there is shrinkage, $g(u) = u/(1 + \lambda)$.

Threshold functions



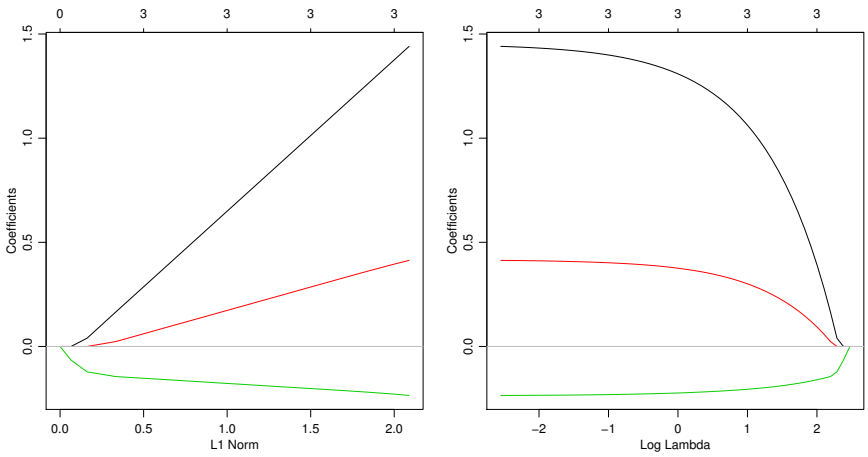
Soft thresholding

Top panels: the sum $g(\beta)$ of the L_1 penalty and the least squares function (both in grey) is the black line, which has a cusp at $\beta = 0$. If the left- and right-hand derivatives of the sum are equal at zero, then the minimiser (at the red vertical line) is non-zero, but not otherwise. Bottom panels: the derivative $g'(\beta) = 0$ when $\beta = \tilde{\beta}$.



Example: cement data

☐ Estimated coefficients for lasso fit against L_1 norm and λ:



Comments

☐ **Least angle regression (LAR)** is similar to the lasso, and can compute the lasso solution path for all λ in $O(n^3)$ operations (faster than ridge, $O(np^2)$, when $p \gg n$).

☐ **Theory**: one can ask about the properties of $\tilde{\beta}_\lambda$ in suitable settings (e.g., $n, p \to \infty$ with $p/n \to c > 0$). Then under certain conditions one can show that lasso variable is consistent (i.e., the probability that the variables with $\beta_r \neq 0$ are selected tends to 1), but that the $\tilde{\beta}_\lambda$ themselves are inconsistent (because soft thresholding implies that $|\tilde{\beta}_{\lambda,r}|$ is systematically smaller than $|\beta_r|$).

☐ Many (many!) variants and related procedures exist to overcome such problems.

☐ **Computation**: lasso and elastic net penalisations available in `R` package `glmnet` and extend to generalized linear models and more general regressions (later).

☐ For any regression model we can define the **degrees of freedom** as

$$\sigma^{-2} \sum_{j=1}^{n} \mathrm{cov}(y_j, \widehat{y}_j) = \mathrm{tr}\{\mathrm{cov}(y, \widehat{y})\}/\sigma^2;$$

this reduces to previous definitions but can be computed in more situations.

☐ When $D(\beta)$ is a general loss function (e.g., a negative log likelihood for a GLM), the exact algorithm above is replaced by a **coordinate descent algorithm** that updates each $\tilde{\beta}_r$ in turn, with the other components fixed. This too is very efficient.

3.4 Splines

Basis functions

- ☐ We seek to estimate a function $\mu(x)$ based on data $(x_1, y_1), \ldots, (x_n, y_n)$.
- ☐ There are n parameters $\mu_1 = \mu(x_1), \ldots, \mu_n = \mu(x_n)$ (plus noise, ...), so we assume that $\mu(x)$ belongs to a suitable class of functions, defined for $x \in \mathcal{X}$.
- ☐ Simple linear model is
$$\mu_{n\times 1} = B_{n\times p}\beta_{p\times 1}, \quad \operatorname{rank}(B) = p \leq n,$$
with the columns of B evaluations at $x_1, \ldots, x_n$ of **basis functions**.
- ☐ The basis functions may be
  - **global** (e.g., polynomials, trigonometric/Fourier functions),
  - **local** (e.g., splines),
  - **multiscale** (e.g., wavelets).
- ☐ We choose the basis for
  - suitability for the problem at hand (e.g., suitably smooth), and
  - computational reasons—want fast, preferably $\mathcal{O}(n)$, handling of $n \times n$ matrices.
- ☐ Focus on **spline functions**, on which there is a huge literature.

Aside: Polynomial regression

- ☐ Classical approach is to fit a polynomial of degree $p-1$, i.e.,
$$\mu(x_j) = \beta_0 + \beta_1 x_j + \cdots + \beta_{p-1} x_j^{p-1},$$
and choose $\beta_0, \ldots, \beta_{p-1}$ to minimise the sum of squares
$$\sum_{j=1}^{n} \{y_j - \mu(x_j)\}^2 = \sum_{j=1}^{n} \left\{ y_j - (\beta_0 + \beta_1 x_j + \cdots + \beta_{p-1} x_j^{p-1}) \right\}^2,$$
giving $\widehat{\beta}_{p\times 1} = (B^{\mathrm{T}}B)^{-1}B^{\mathrm{T}}y$, where (j,i) element of $n \times p$ matrix B is x_j^{i-1}.
- ☐ Comments:
  - easily copes with missing values/unequally spaced observations;
  - use orthogonal polynomials to avoid numerical problems if n, k large;
  - sensitivity to observations at extremities of series often leads to poor fit;
  - usually doesn't work well because infinite differentiability everywhere is generally unnecessarily restrictive.

Piecewise linear basis

☐ Place **knots** of a univariate x at $x_1^* < \cdots < x_K^*$, and define **tent functions**

$$
\begin{aligned}
b_1(x) &= \begin{cases} (x_2^* - x)/(x_2^* - x_1^*), & x_1^* \le x \le x_2^*, \\ 0, & \text{otherwise,} \end{cases} \\
b_k(x) &= \begin{cases} (x - x_{k-1}^*)/(x_k^* - x_{k-1}^*), & x_{k-1}^* < x \le x_k^*, \\ (x_{k+1}^* - x)/(x_{k+1}^* - x_k^*), & x_k^* < x \le x_{k+1}^*, \end{cases} \quad k = 2, \ldots, K-1, \\
b_K(x) &= \begin{cases} (x - x_{K-1}^*)/(x_K^* - x_{K-1}^*), & x_{K-1}^* \le x \le x_K^*, \\ 0, & \text{otherwise :} \end{cases}
\end{aligned}
$$

these are non-zero only in (x_{k-1}^*, x_{k+1}^*) (**compact support**) and take value 1 at x_k^*.

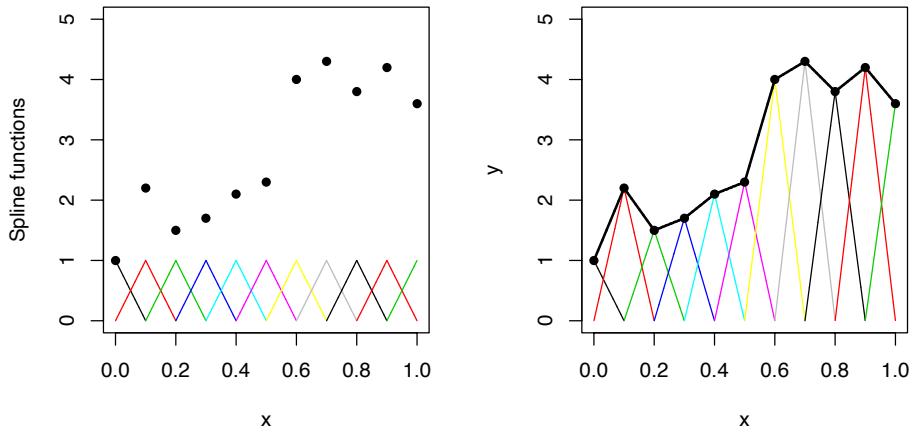
☐ An exact linear interpolant of data $y_1, \ldots, y_K$ at the knots is the function

$$\mu(x) = \sum_{k=1}^{K} b_k(x) y_k = B(x)^{\mathrm{T}} y,$$

which by construction

- passes through the points (x_k^*, y_k) and
- is linear between the knots.

Piecewise linear basis II



☐ Left: piecewise linear basis functions $b_k(x)$ and data (x_k^*, y_k).

☐ Right: functions $b_k(x)y_k$ and linear interpolant (bold).

Statistical use

- Aim for summary of the n observations, so interpolation not useful.
- Could use $K < n$ knots, but fit tends to depend heavily on their locations, so better to use high(ish) K and impose structure by penalising roughness of $\mu(x)$:

$$\widehat{\beta}_\lambda = \text{argmin}_\beta \left\{ \|y - B\beta\|^2 + \lambda \sum_{k=2}^{K-1} \left\{ \mu(x^*_{k-1}) - 2\mu(x^*_k) + \mu(x^*_{k+1}) \right\}^2 \right\}.$$

- The second term sums squared numerical second derivatives at the internal knots, and λ imposes the degree of penalisation:
  - $\lambda = 0$ (no penalty) gives the interpolant,
  - $\lambda \to \infty$ forces the second derivatives to be zero, so gives a straight-line fit.
- On setting $\beta_k = \mu(x^*_k)$ and writing

$$\begin{pmatrix} \beta_1 - 2\beta_2 + \beta_3 \\ \beta_2 - 2\beta_3 + \beta_4 \\ \beta_3 - 2\beta_4 + \beta_5 \\ \vdots \end{pmatrix} = \begin{pmatrix} 1 & -2 & 1 & 0 & 0 & 0 & \cdots \\ 0 & 1 & -2 & 1 & 0 & 0 & \cdots \\ 0 & 0 & 1 & -2 & 1 & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \vdots \end{pmatrix} = D_{(K-2)\times K}\beta_{K\times 1},$$

  the penalty is $\sum_{k=2}^{K-1}(\beta_{k-1} - 2\beta_k + \beta_{k+1})^2 = (D\beta)^{\mathrm{T}} D\beta = \beta^{\mathrm{T}} D^{\mathrm{T}} D\beta = \beta^{\mathrm{T}} S\beta$, say.

Penalized fit

- The penalty matrix S is of side $K \times K$ but of rank $K - 2$, because

$$S1_K = Sx^*_{K\times 1} = 0_K :$$

  the null space of S consists of all straight lines $\beta_0 1_K + \beta_1 x^*$, which are unpenalised.
- Hence (recalling ridge regression),

$$\widehat{\beta}_\lambda = \text{argmin}_\beta \left\{ \|y - B\beta\|^2 + \lambda\beta^{\mathrm{T}} S\beta \right\} = (B^{\mathrm{T}}B + \lambda S)^{-1} B^{\mathrm{T}} y$$

  giving

$$\begin{aligned} \textbf{fitted values} \qquad \widehat{y} &= B\widehat{\beta}_\lambda = B(B^{\mathrm{T}}B + \lambda S)^{-1}B^{\mathrm{T}}y = H_\lambda y, \\ \textbf{equivalent degrees of freedom} \quad \mathrm{df}_\lambda &= \mathrm{tr}(H_\lambda) = \sum_{k=1}^{K} \frac{1}{1 + \eta_k \lambda}, \end{aligned}$$

  where
  - $\eta_1 \le \cdots \le \eta_K \in [0, 1]$ are the eigenvalues of $(B^{\mathrm{T}}B)^{-1/2} S (B^{\mathrm{T}}B)^{-1/2}$,
  - $\eta_1 = \eta_2 = 0$, corresponding to the null space of S, so
  - df_λ is monotone decreasing in λ, with

$$(\lambda = 0) \qquad K \ge \mathrm{df}_\lambda \ge 2 \qquad (\lambda \to \infty).$$

Higher-order splines

- The p**th degree spline** basis with **knots** $x_1^* < \cdots < x_K^*$ is

$$1, x, \ldots, x^p, (x - x_1^*)_+^p, \ldots, (x - x_K^*)_+^p,$$

  where $u_+ = \max(u, 0)$ is the **positive part function**.
- The resulting basis matrix B is highly collinear and gives an implausible statistical model.
- B**-spline** bases span the same linear space, but have better numerical properties. They are defined by adding **boundary knots** x_0^* and x_{K+1}^* and setting up an **augmented knot sequence**

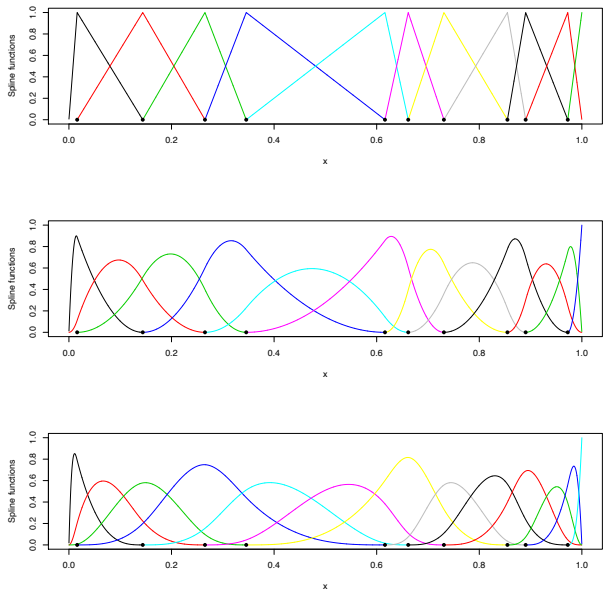
$$\tau_1 \leq \cdots \leq \tau_M \leq x_0^* \leq \tau_{M+1} = x_1^* \leq \cdots \leq \tau_{M+K} = x_K^* \leq x_{K+1}^* \leq \tau_{K+1+M} \leq \cdots \leq \tau_{K+2M};$$

  typically the τ_k outside $[x_0^*, x_{K+1}^*]$ are set to the boundary knot values. Then

$$\begin{aligned} B_{k,1}(x) &= I(\tau_k \leq x < \tau_{k+1}), \quad k = 1, \ldots, K + 2M - 1, \\ B_{k,m}(x) &= \frac{x - \tau_k}{\tau_{k+m-1} - \tau_k} B_{k,m-1}(x) + \frac{\tau_{k+m} - x}{\tau_{k+m} - \tau_{k+1}} B_{k+1,m-1}(x), \quad k = 1, \ldots, K + 2M - m, \end{aligned}$$

  where we set $B_{k,1} \equiv 0$ if $\tau_k = \tau_{k+1}$ (avoiding division by zero).
- Cubic splines ($p = 3$, $M = 4$) give visually smooth functions.
- $K = 10$ on the next slide, with $M = 2$ (linear), $M = 3$ (quadratic) and $M = 4$ (cubic), and the τ_k set to equal the boundary knots.

Linear, quadratic and cubic B-splines



Natural cubic spline

- Suppose the x_j are distinct (no loss of generality) and

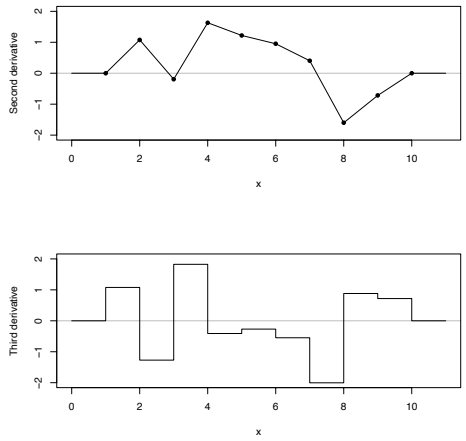
$$a < x_1 < \cdots < x_n < b, \quad \mathcal{X} = [a, b] \subset \mathbb{R}.$$

- A **natural cubic spline** adds the constraint that the function is linear outside $[x_1, x_n]$, and thus avoids high variance due to quadratic and higher terms outside this interval.
- A natural cubic spline
  - has $K = n$ knots, at $x_1 < \cdots < x_n$,
  - is a cubic polynomial on each interval between knots,
  - is continuous, with continuous first and second derivatives at each knot, and
  - is linear on $[a, x_1]$ and $[x_n, b]$, with zero second and third derivatives at x_1 and x_n,
  - has

$$2 + 4(n-1) + 2 \text{ parameters } - 3n \text{ linear constraints } = n$$

    degrees of freedom (df), which can be split into
    - ▷ 2 df for a linear fit, plus
    - ▷ $n - 2$ df for the second derivatives $\mu''(x_2), \ldots, \mu''(x_{n-1})$.

Natural cubic spline



- A natural cubic spline may be constructed by integrating a linear second derivative function $\mu''(x)$ which is determined by $\mu''(x_2), \ldots, \mu''(x_{K-1})$ and because $\mu''(x) \equiv 0$ for $x \notin (x_1, x_K)$.
- On integrating twice we gain two constants: $\mu(x) = \beta_0 + \beta_1 x + \int_0^x \int_0^{x'} \mu''(u)\, \mathrm{d}u\, \mathrm{d}x'$.
- Above $x_1 = 1, \ldots, x_{10} = 10$, so the spline is determined by $\mu''(2), \ldots, \mu''(9)$ and the line.

Optimality of natural cubic splines

- ☐ Let $\mathcal{S}_2(\mathcal{X})$ denote the set of functions μ differentiable on $\mathcal{X} = [a, b]$ with absolutely continuous first derivative μ': i.e., there exists an integrable function μ'' such that $\int_a^x \mu''(u)\mathrm{d}u = \mu'(x) - \mu'(a)$ for $x \in \mathcal{X}$.
- ☐ Clearly any μ with two continuous derivatives on $\mathcal{X}$ lies in $\mathcal{S}_2(\mathcal{X})$.

Theorem 32 *Suppose $n \geq 2$, that $a < x_1 < \cdots < x_n < b$, and that μ is the natural cubic spline interpolating $y_1, \ldots, y_n$ at $x_1, \ldots, x_n$. If $\tilde{\mu} \in \mathcal{S}_2(\mathcal{X})$ also interpolates the y_j, then*

$$\int_{\mathcal{X}} \tilde{\mu}''^2 \geq \int_{\mathcal{X}} \mu''^2,$$

with equality iff $\tilde{\mu} \equiv \mu$.

- ☐ Thus μ minimises the **roughness penalty** $\lambda \int_{\mathcal{X}} \mu''^2$ in a larger class of functions than that to which it belongs, making it a natural choice as an interpolant, because minimising

$$\sum_{j=1}^{n} \{y_j - \tilde{\mu}(x_j)\}^2 + \lambda \int_{\mathcal{X}} \tilde{\mu}''(x)^2 \, \mathrm{d}x$$

for $\tilde{\mu} \in \mathcal{S}_2(\mathcal{X})$ will automatically result in a natural cubic spline μ: if $\tilde{\mu}(x_j) = \mu(x_j)$, then the penalty is reduced by using μ.

Note to Theorem 36

Let $\nu = \tilde{\mu} - \mu \in \mathcal{S}_2(\mathcal{X})$, and note that $\nu(x_j) = 0$ for each j, since $\mu(x_j) = \tilde{\mu}(x_j) = y_j$. The natural boundary conditions imply that $\mu''(a) = \mu''(b) = 0$, so integration by parts yields

$$0 = \left[\mu''(x)\nu'(x)\right]_a^b = \int_{\mathcal{X}} (\mu''\nu')' = \int_{\mathcal{X}} \mu''\nu'' + \int_{\mathcal{X}} \mu'''\nu',$$

and hence the facts that μ''' is piecewise constant and that $\nu(x_j) = 0$ yields

$$\int_{\mathcal{X}} \mu''\nu'' = -\int_{\mathcal{X}} \mu'''\nu' = -\sum_{j=1}^{n-1} \mu'''(x_j^+) \int_{x_j}^{x_{j+1}} \nu' = -\sum_{j=1}^{n-1} \mu'''(x_j^+)\{\nu(x_{j+1}) - \nu(x_j)\} = 0.$$

Hence

$$\int_{\mathcal{X}} \tilde{\mu}''^2 = \int_{\mathcal{X}} (\mu'' + \nu'')^2 = \int_{\mathcal{X}} \mu''^2 + 2\int_{\mathcal{X}} \mu''\nu'' + \int_{\mathcal{X}} \nu''^2 = \int_{\mathcal{X}} \mu''^2 + \int_{\mathcal{X}} \nu''^2 \geq \int_{\mathcal{X}} \mu''^2,$$

wth equality iff $\nu''(x) \equiv 0$. This occurs iff $\nu(x)$ is linear, but since $\nu(x_j) = 0$ at at least two points, $\nu(x) = 0$ for all $x \in \mathcal{X}$.

More splines

- Sometimes cyclic effects (e.g., seasonality, diurnal variation) must be modelled smoothly, so (e.g.) December joins smoothly onto January. Then the penalty and spline basis must be modified accordingly, to give a **cyclic (cubic) spline**.
- **P-splines** are a version of B-splines (usually with equally-spaced knots) in which a difference penalty is applied to the parameters to control the wiggliness of μ, e.g.,

$$\sum_{k=1}^{K-1} w_k(\beta_{k+1}-\beta_k)^2 = \beta^{\mathrm{T}} D^{\mathrm{T}} W D \beta, \quad \text{with} \quad D = \begin{pmatrix} -1 & 1 & 0 & 0\cdots & \\ 0 & -1 & 1 & 0 & \cdots \\ \cdot & \cdot & \cdot & \cdot & \cdots \end{pmatrix},$$

  and $W = \mathrm{diag}(w_1, \ldots, w_{K-1})$. These are easy to set up and flexible, but messy if the knots are not equi-spaced, and the penalty is less readily interpreted.
- For an **adaptive spline** we can let $w_k \equiv w_k(x)$ vary with x, for example setting $w(x) = B(x)\lambda_{L\times 1}$ and thus having $D^{\mathrm{T}}WD = \sum_l \lambda_l D^{\mathrm{T}} \mathrm{diag}\{B_l(x)\} D$, where $B_l(x)$ is the lth column of $B(x)$, then estimating the vector λ.
- Other possibilities include (Wood, 2017, Chapter 5)
  - **shape-constrained splines** to impose, e.g., monotonicity on the fit;
  - **thin-plate**, **Duchon** and **tensor product** splines used in spatial problems; and
  - **soap film splines** used when smoothing over complex domains.

Motorcycle data: adaptive fit

Standard (left) and adaptive (right) spline fits, the latter with $K = 40$ and $L = 5$: