Regression Methods

Myrto Limnios

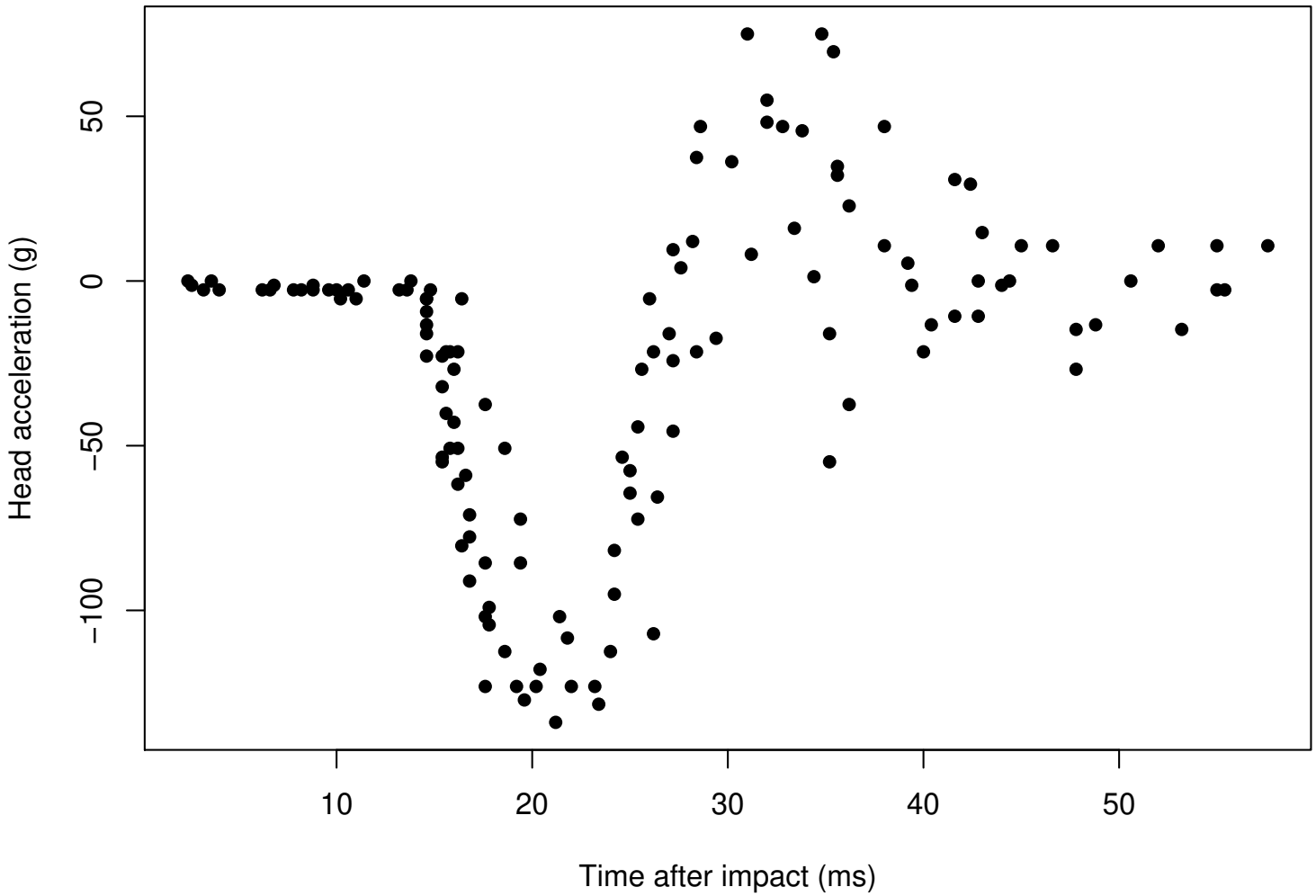
Autumn 2025 - Week 11

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 $x_- \leq x_1 < \cdots < x_n \leq x_+$ 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**,

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

  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.

$\mu(x) = \sum_{j=1}^{n} \mu_j b_j(x)$ with $b_j \in C^2$ diff.

- 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 $S 1_n = S(x_1, \ldots, x_n)^{\mathrm{T}} = 0$.

$b_i(x) = a_1 x + a_0$ has dimension = 2

- The penalised sum of squares

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

  keep terms depending on μ — so we can differentiate w.r.t μ and obtain

  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 $\mathbf{edf}_\lambda = \mathrm{tr}(H_\lambda) = \sum_{j=1}^{n} (1 + \lambda \delta_j)^{-1}$, where $\delta_1 \geq \cdots \geq \delta_3 > \delta_2 = \delta_1 = 0$ are the eigenvalues of S. As λ increases $\mathbf{edf}_\lambda$ decreases monotonically from $\mathbf{edf}_0 = n$ towards $\mathbf{edf}_\infty = 2$.

- In principle we might take any basis functions, but in practice we usually take local polynomials known as **splines** that have good approximation properties.

$P = \alpha_0 + \alpha_1 X + \alpha_2 X^2 + \alpha_3 X^3$

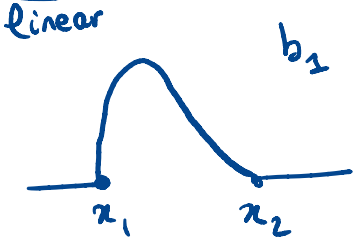
- There are many forms of splines, which
  - are often cubic polynomials with finite support between values of x known as **knots**, $x_1^*, \ldots, x_K^*$, and then S is tri-diagonal,
  - sometimes form a **natural cubic spline**, which has $K = n$ and certain optimality properties,
  - are discussed 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

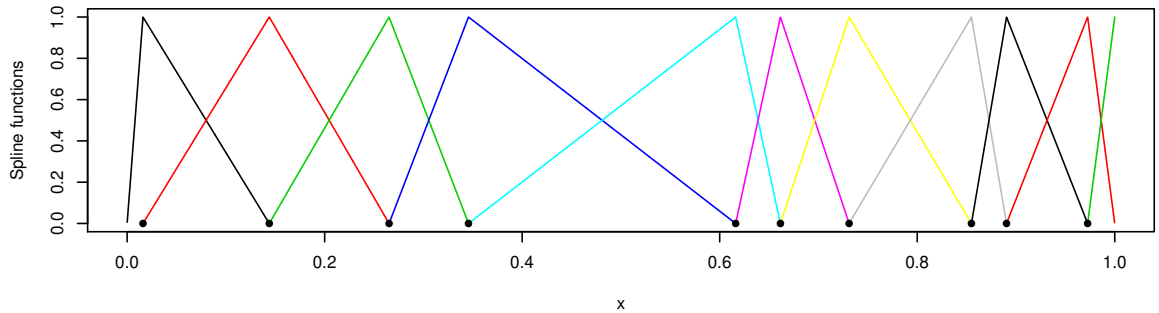
$$(y - N\mu)^{\mathrm{T}} W (y - N\mu) + \lambda \mu^{\mathrm{T}} S \mu.$$

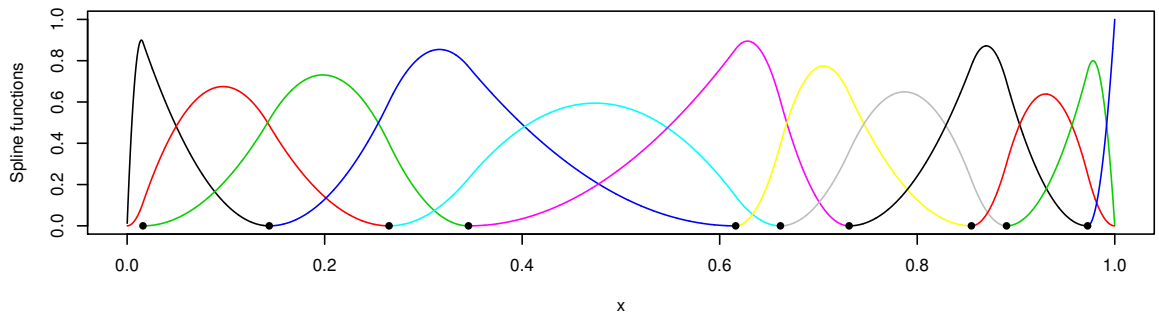
where $S_{n' \times n'}$ arises as before from the roughness penalty on $\mu(x)$.

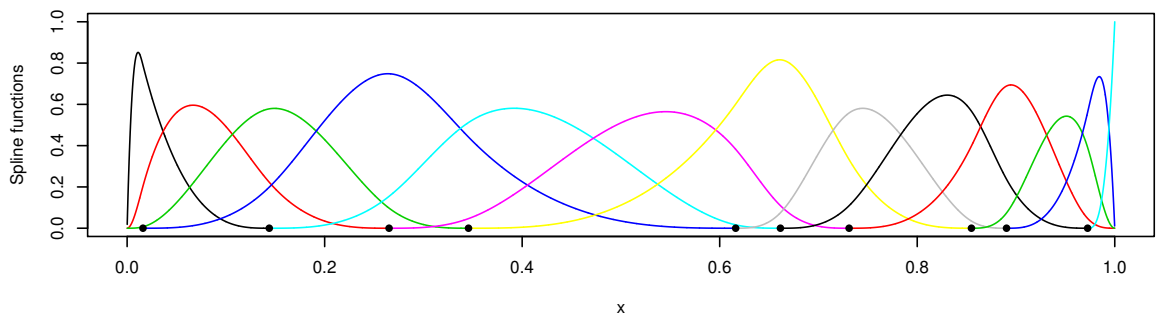
Linear, quadratic and cubic B-splines

$$P = \underbrace{\alpha_0 + \alpha_1 X}_{\text{linear}} + \alpha_2 X^2 + \alpha_3 X^3$$



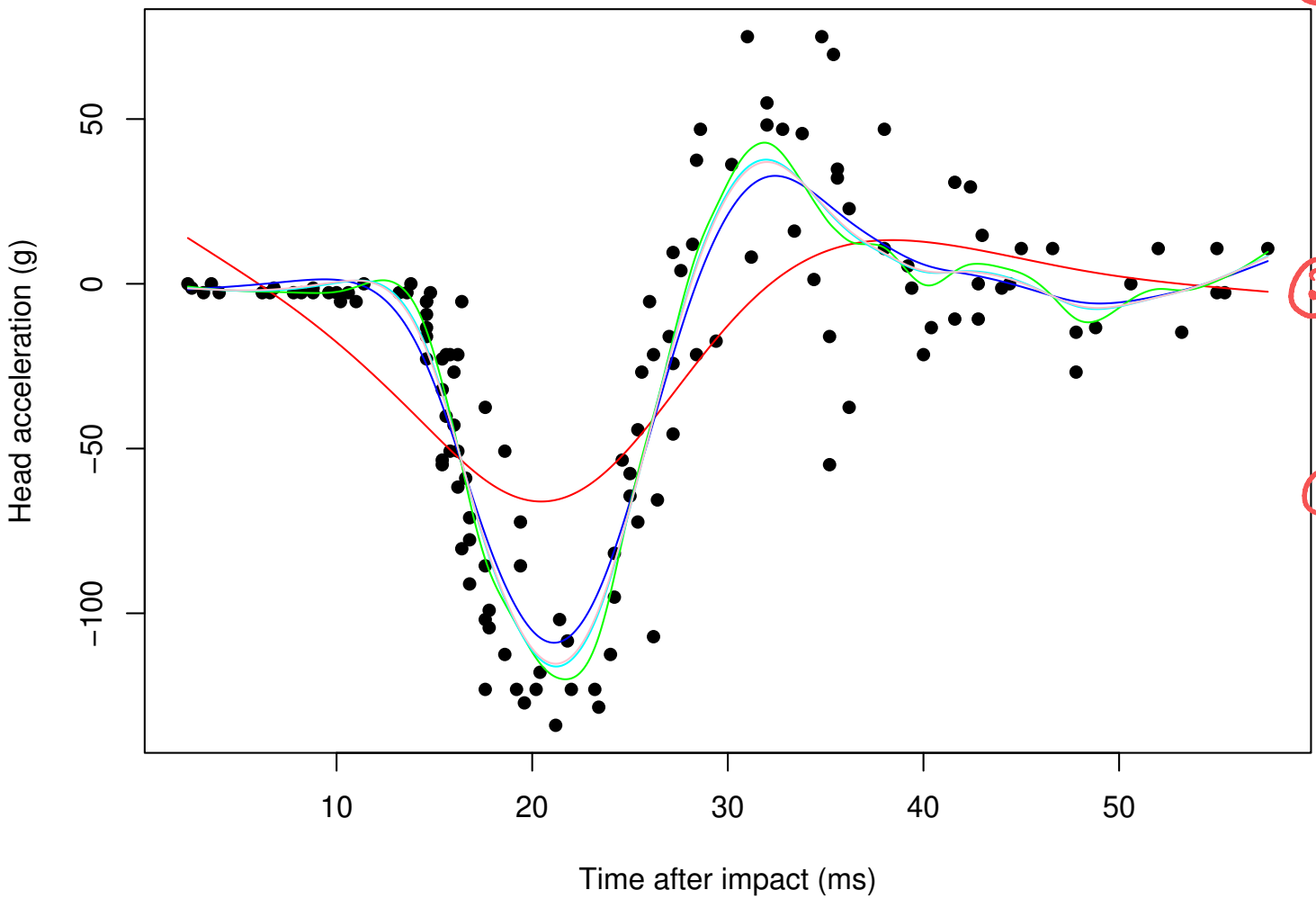






Example: Motorcycle data

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



1. $f_M(x) = \sum_{j \in M} \mu_j b_j(x)$

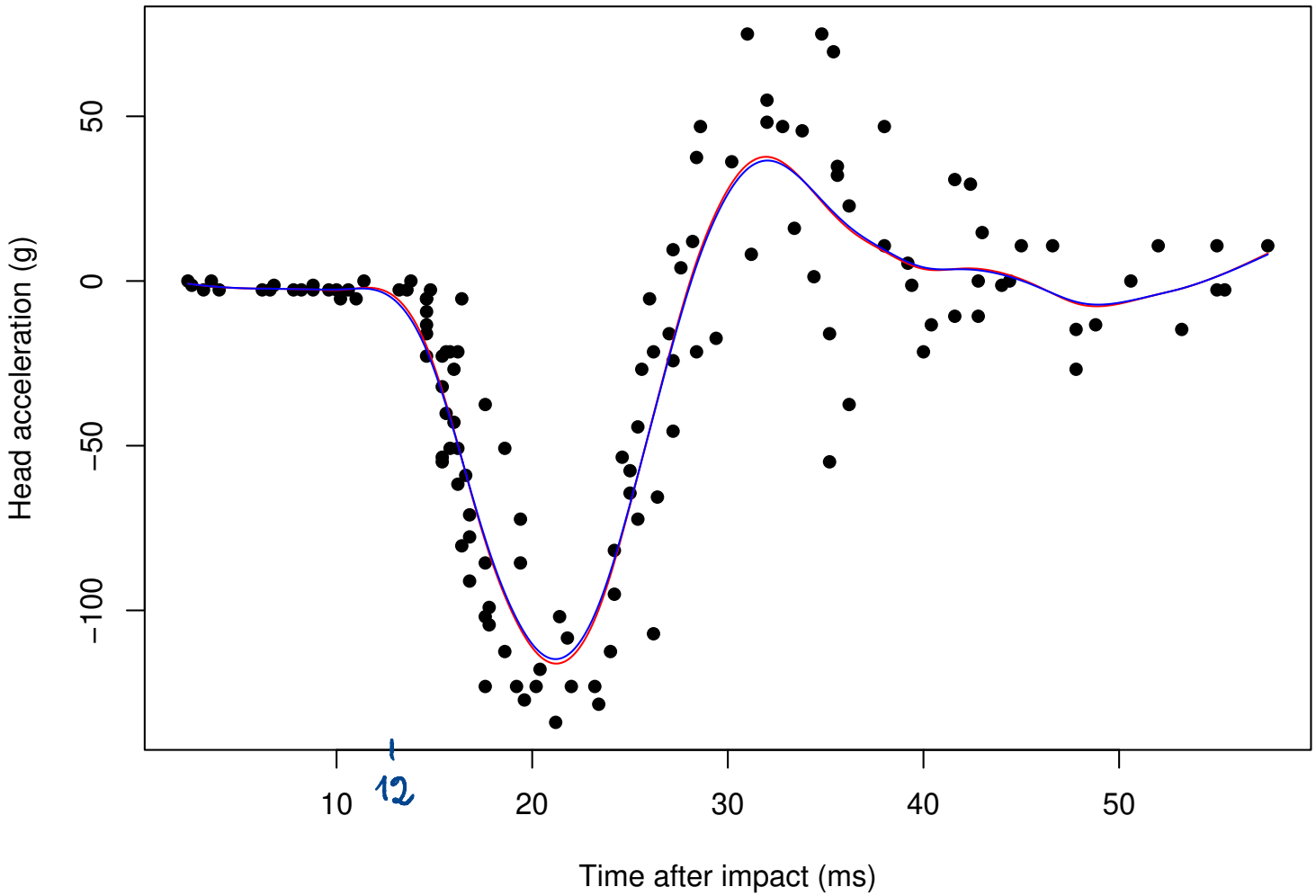
$\hat{f} \in \operatorname{argmin} \|y - f_M(x)\| + \lambda \int (f''(x))^2 dx$

2. Choose (b_j) to be natural cubic splines (up to 3rd degree)

3. Model selection
   - CV criterion
   - GCV "

Example: Motorcycle data

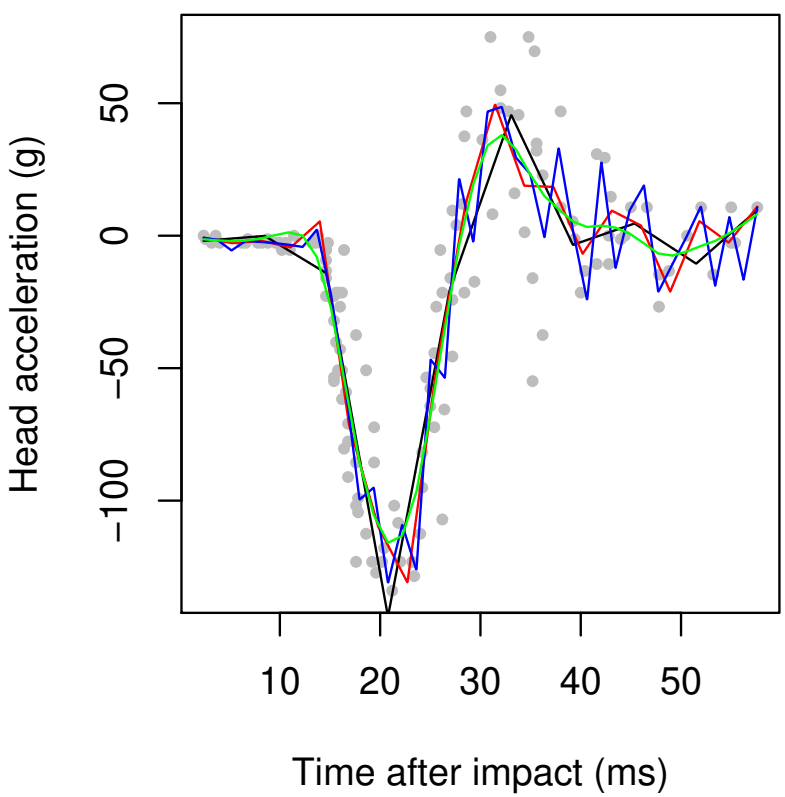
Scatterplot smooths based on natural cubic splines with weights 16 when $x \leq 12$ and 1 for $x > 12$, and **edf** chosen by CV (red, **edf** $= 14.7$) and GCV (blue, **edf** $= 13.7$):

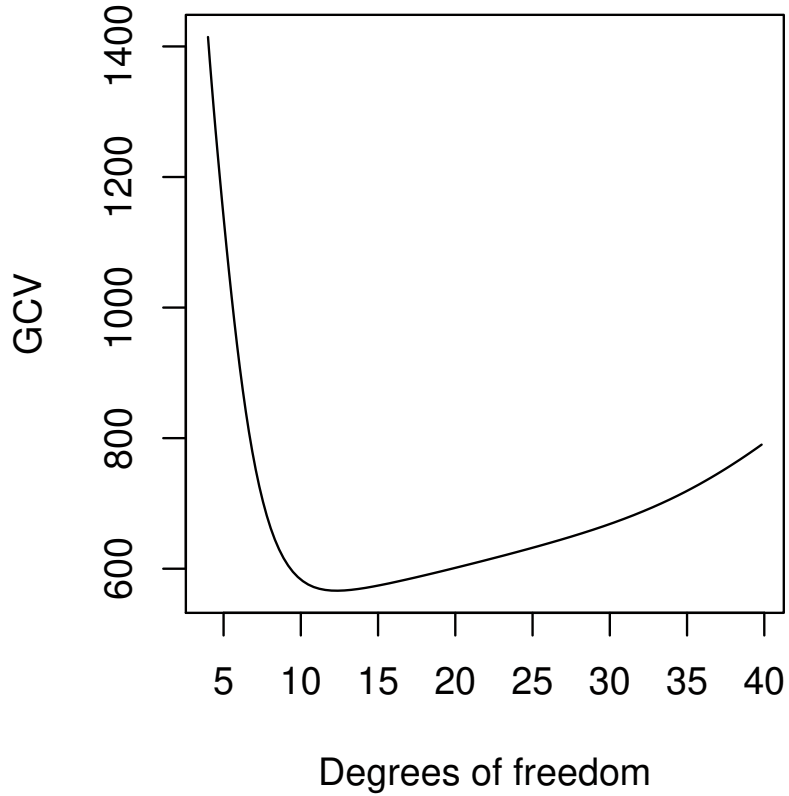


Choosing K and λ

- Above we took $K = n$ basis functions, but for statistical purposes we seek a summary of the data, so we hope that **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



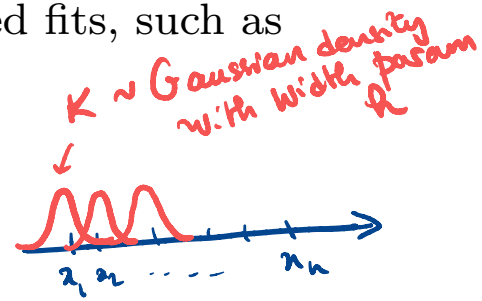


- Left: linear spline fits with $\lambda = 0$ and $K = 10$ (black), 20 (red), 40 (blue), and optimal GCV choice of λ with $K = 40$ (green)
- Right: GCV(λ) as a function of $\mathbf{df}_\lambda$ for $K = 40$.

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.

Regularisation - Lasso

$\|y - X\beta\| + \lambda \|\beta\|_1 \qquad \|\beta\|_1 = \sum_{i \le p} |\beta_i|$

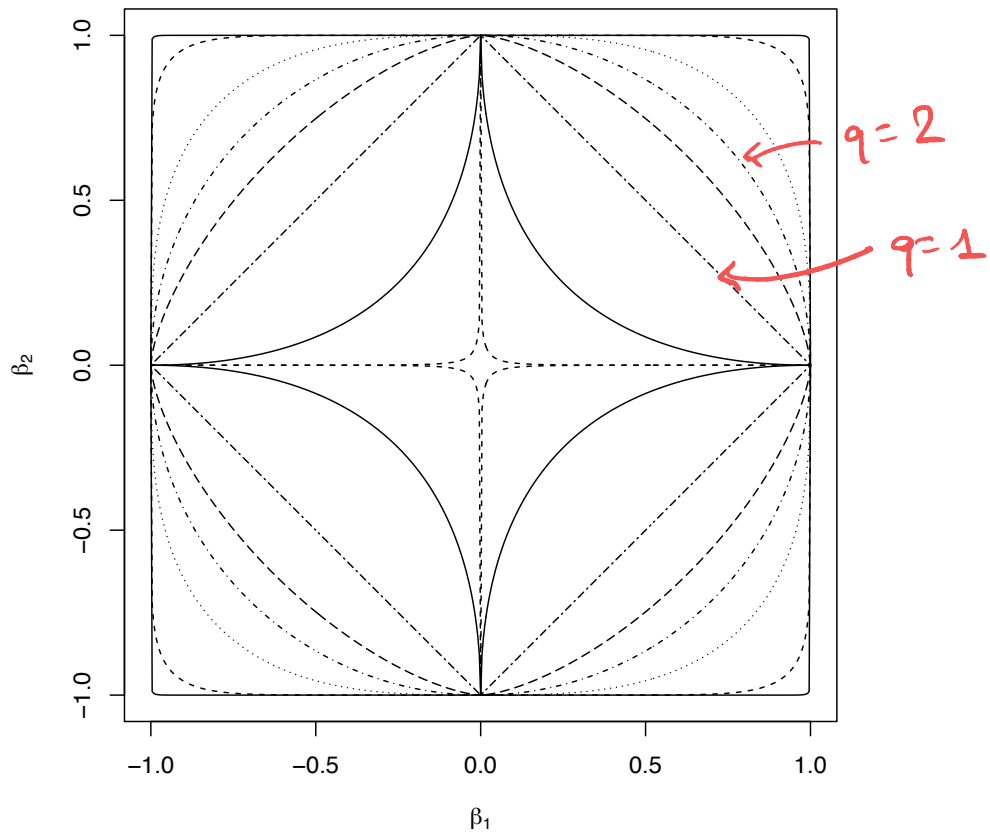
↳ feature selection because of the geometry of unit ℓ_1-ball

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

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

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

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

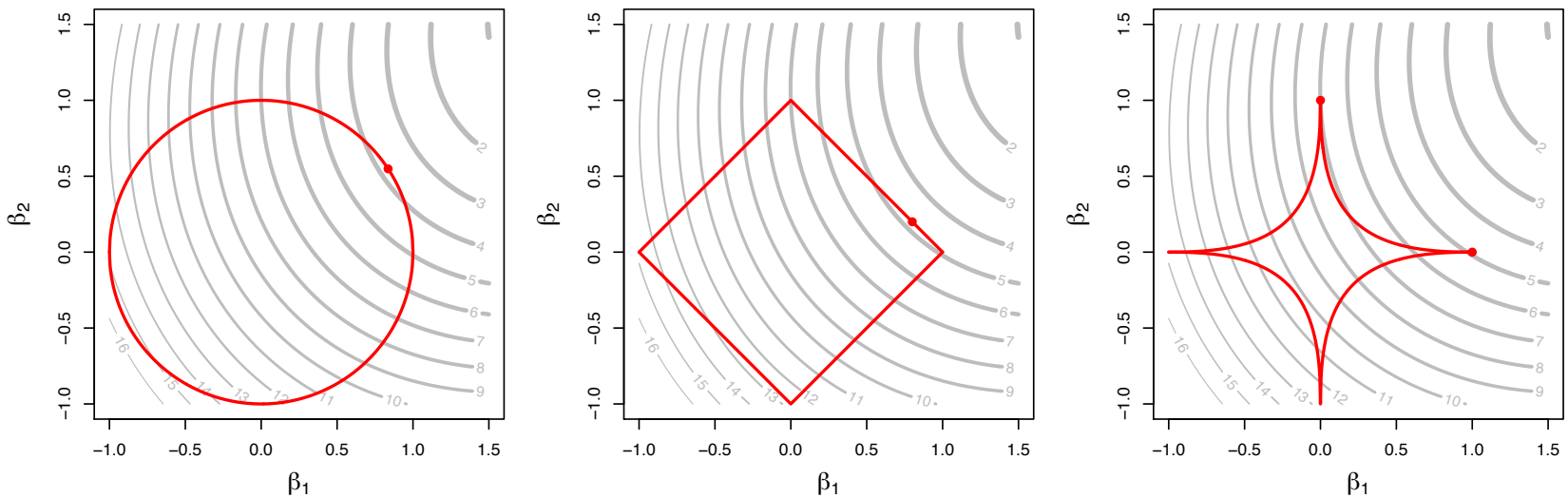
    because otherwise we could reduce $D(\beta)$ while leaving the penalty unchanged, i.e., $\tilde{\beta}_\lambda$ would not be optimal.

- The sets $\|\tilde{\beta}_\lambda\|_q = t$
  - have corners (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.

$\|y - X\beta\|_2 + P_\lambda(\beta)$ $\Leftrightarrow$ $\operatorname{argmin} \|y - X\beta\|_2$ s.t. $P_\lambda(\beta) \le t$

$t \to 0$

$\Rightarrow$ red geometry $\to$ origin

if $\lambda \to 0$ $\Rightarrow$ unconstrained problem $\Rightarrow t \to \infty$

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,

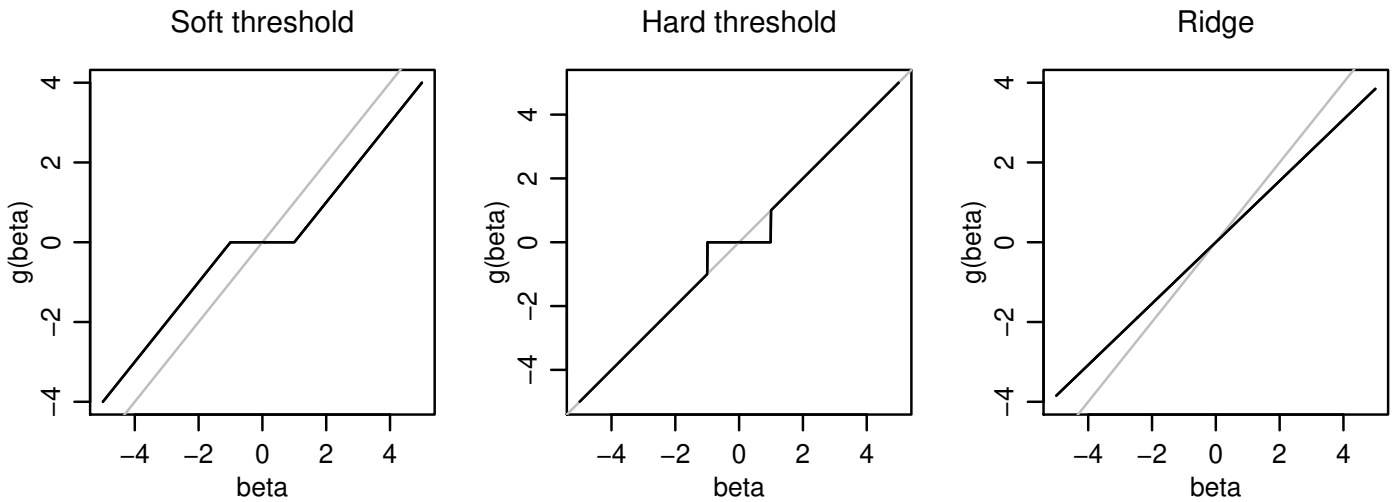
$$g_\lambda(u) = \begin{cases} 0, & |u| < \lambda, \\ \text{sign}(u)(|u| - \lambda), & \text{otherwise}, \end{cases}$$

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

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

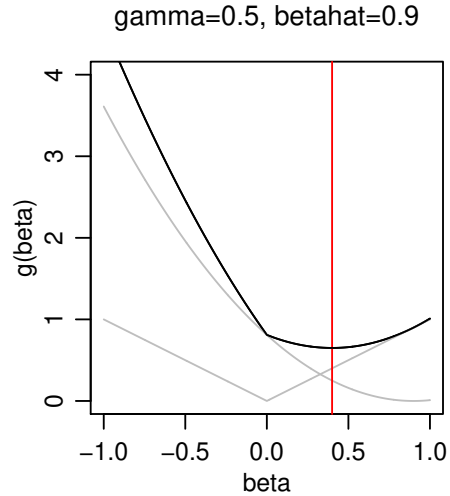
  - for **ridge regression** there is shrinkage, $g(u) = u/(1+\lambda)$.

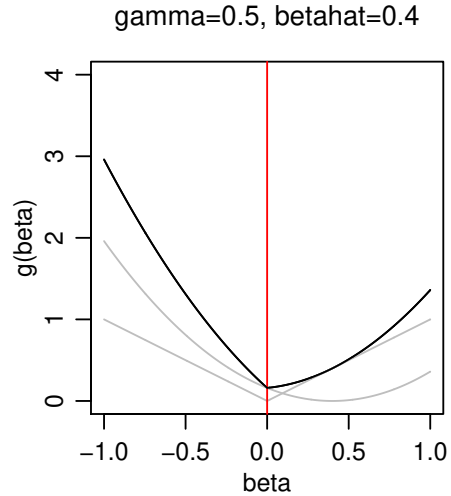
Threshold functions

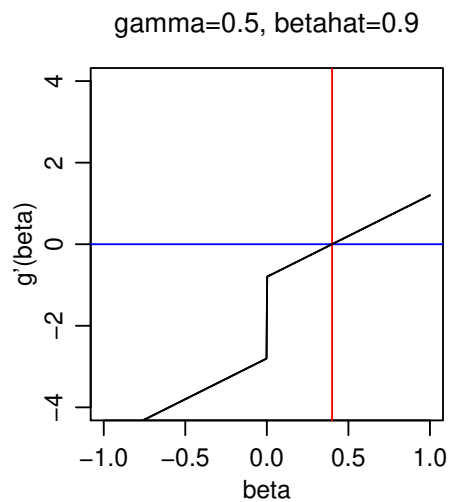


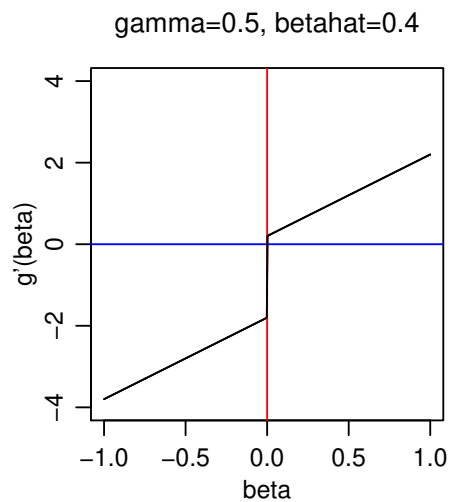
Soft thresholding

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 $\beta = \tilde{\beta}$.



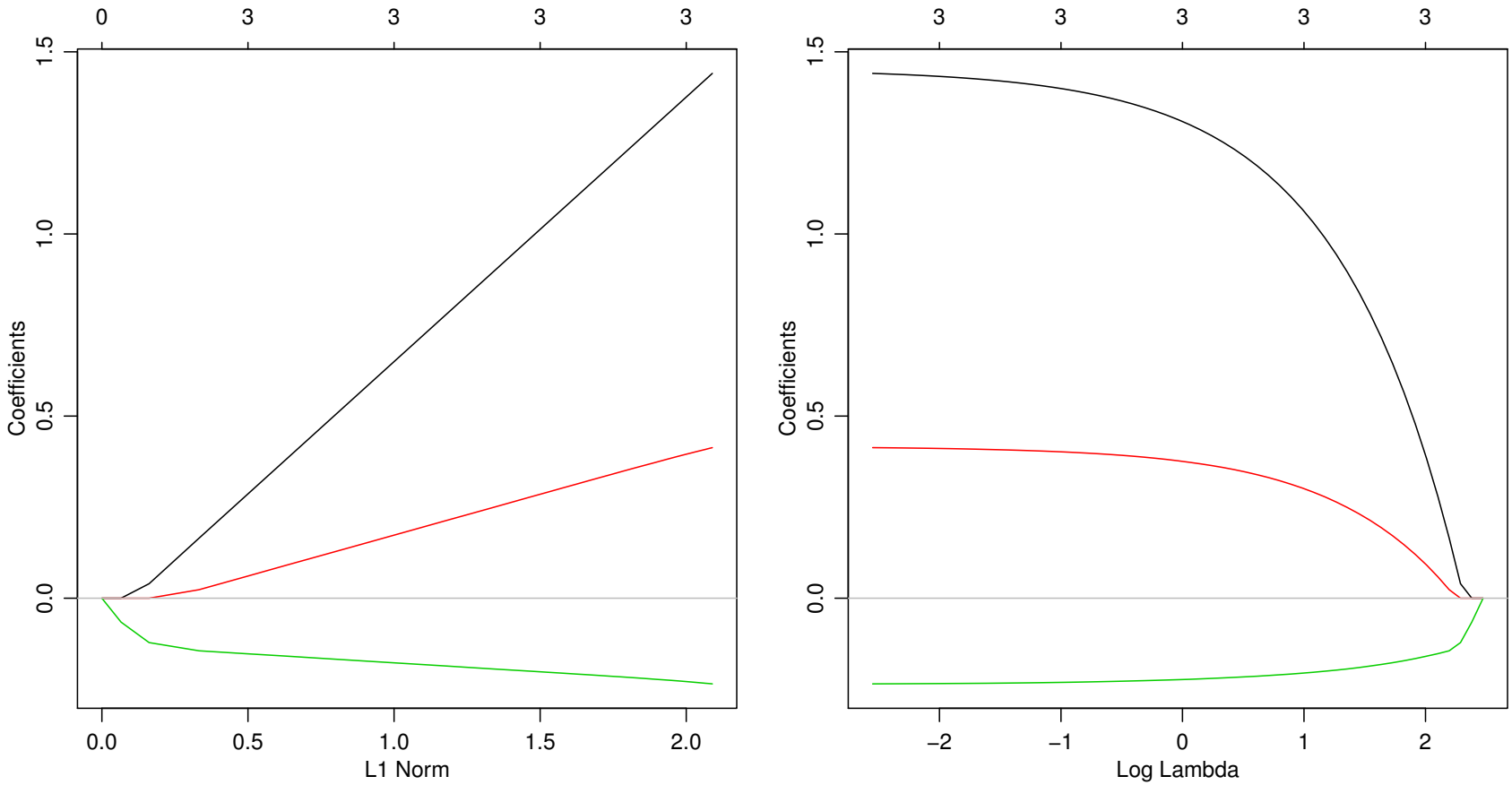






Example: cement data

- 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.