

Smooth modelling of (very) large datasets

Simon N. Wood

University of Bath

Generalized additive models

- ⑥ Consider a univariate response y and corresponding predictors $x_1, x_2, x_3 \dots$ (possibly vectors).
- ⑥ A GAM models the dependence of y on the x_j via, $y_i \sim \text{some exponential family distribution}$

$$g\{\mathbb{E}(y_i)\} = \mathbf{X}_i^* \boldsymbol{\theta} + f_1(x_{1i}) + f_2(x_{2i}) + \dots$$

- ⑥ The f_j are smooth functions, to be estimated; g is a known monotonic function; $\mathbf{X}_i^* \boldsymbol{\theta}$ represents parametric model components.
- ⑥ \dots offers a nice balance of flexibility and structure.

GAM fitting

- ⑥ If f_j 's are completely free, then Maximum Likelihood Estimation leads to overfitting.
- ⑥ Solution is to penalize the likelihood using a penalty dependent on the wiggleness of component functions. i.e. to maximize

$$\log \text{likelihood} - \sum_j \lambda_j \times [\text{wiggleness of } f_j]$$

smoothing params, λ_j , control fit-smoothness tradeoff.

- ⑥ Actually penalized likelihood maximized by penalized iterative least squares.

Estimation by backfitting

- At simplest, estimation of smooth functions is performed by iteratively smoothing *partial residuals*

$$\hat{\epsilon}_i^j = y_i - \mathbf{X}_i^* \hat{\boldsymbol{\theta}} - \sum_{k \neq j} \hat{f}_k(x_{ki})$$

w.r.t. x_j to obtain \hat{f}_j (at the observed x_{ji} values).
Hastie and Tibshirani (1986,1990)

- + elegant and efficient if smoothing efficient.
- Automatic smoothness selection very costly [$O(n^2)$?].

Estimation by generalized splines

- ⑥ Find *functions* maximizing the penalized likelihood, using RKHS methods. (Wahba, 1990, Gu, 2002).
- ⑥ + can't find a better model, given objective!
- ⑥ + Automatic smoothness selection same cost as fitting.
- ⑥ - cost of fitting $O(n^3)$.
- ⑥ - More efficient approximations don't allow huge data set tricks later!

Penalized regression splines

- ⑥ Represent each smooth using a low rank spline like basis...

$$f_j(x) = \sum_k \beta_{jk} b_{jk}(x)$$

where $b_{jk}(x)$ are known basis functions and the β_{jk} s are coefficients to be estimated.

- ⑥ Associate a wiggleness penalty with each smooth. e.g.

$$\int f_j''(x)^2 dx = \beta_j^T \mathbf{S}_j \beta_j$$

\mathbf{S}_j is a matrix of known coefficients derived from $b_{jk}(x)$.

GAMs using PRS

- Given a basis it's easy to re-write the model as $g\{\mathbb{E}(y_i)\} = \mathbf{X}\beta$, where \mathbf{X} is a model matrix defined by the basis functions and \mathbf{X}^* .
- Fitting then involves minimizing

$$\mathcal{S} = \|\mathbf{y} - \mathbf{X}\beta\|^2 + \sum_j \lambda_j \beta^\top \mathbf{S}_j \beta$$

w.r.t. β (iteratively using a weighted RSS term for MLE). Note that the \mathbf{S}_j 's have been padded with zeros.

- Cost is linear in number of data, n . **Note:** unlike full smoothing splines, \mathbf{X} does not involve λ_j 's.

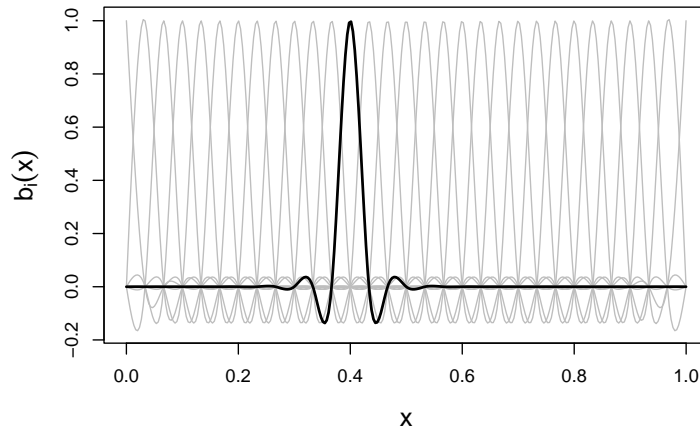
Simple regression splines

- ⑥ Get a spline basis and penalty for a 'representative' subset of the real data.
- ⑥ Use this basis to model the real data.
- ⑥ This is very cheap, but somewhat ad hoc.

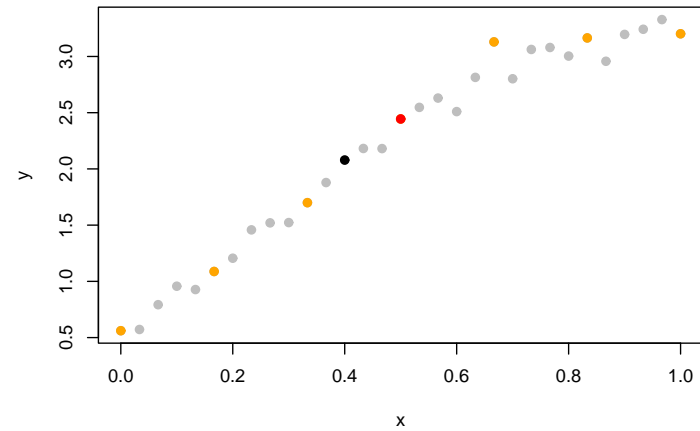
Simple regression splines



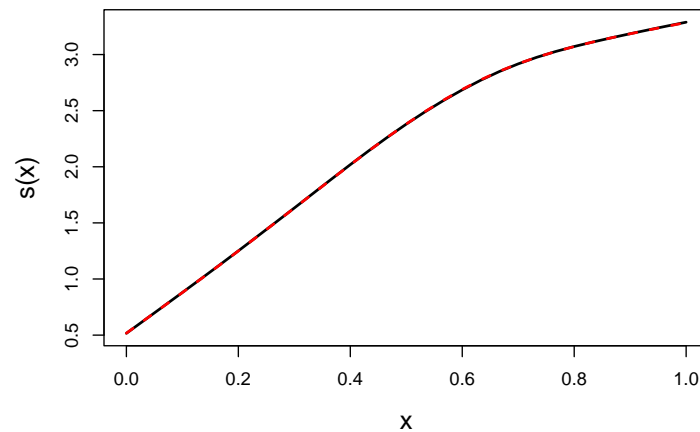
full spline basis



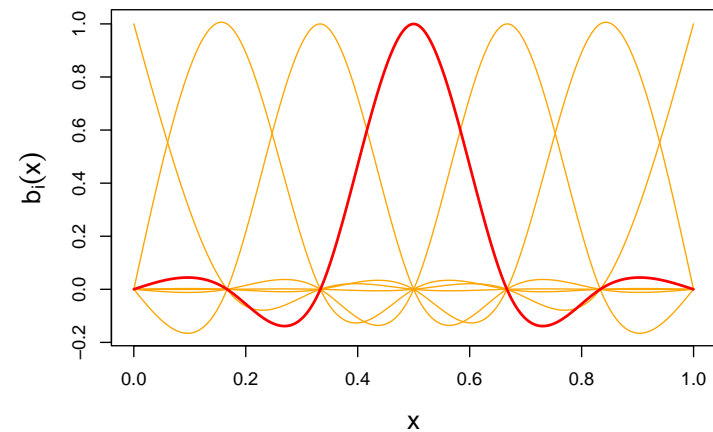
data to smooth



function estimate: full black, regression red



simple regression spline basis



Thin plate regression splines

- ⑥ A thin plate spline is the function (of any number of covariates) maximizing a likelihood penalized using a particular isotropic wiggleness penalty.
- ⑥ It's coefficients are found by minimizing (LS case):

$$\|\mathbf{y} - \mathbf{E}\boldsymbol{\delta} - \mathbf{T}\boldsymbol{\alpha}\|^2 + \lambda\boldsymbol{\delta}^T\mathbf{E}\boldsymbol{\delta}$$

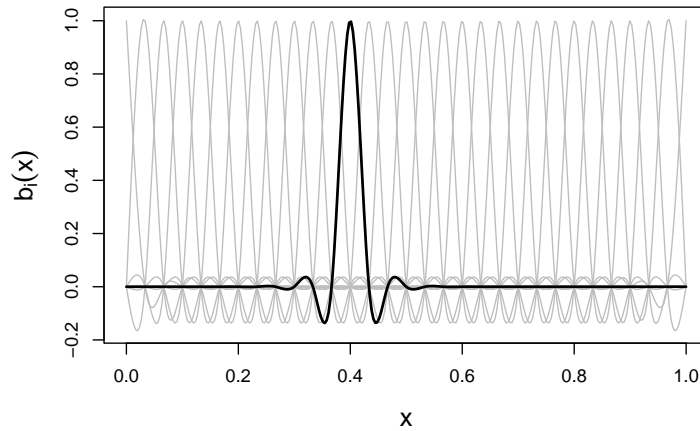
wrt $\boldsymbol{\delta}$, $\boldsymbol{\alpha}$... an $O(n^3)$ problem.

- ⑥ Replacing \mathbf{E} by its rank k truncated eigen-decomposition [$O(n^2k)$] yields the optimal rank k approximation to the TPS, and $O(k^3)$ cost to find the coefficients. (Wood, 2003, JRSSB)

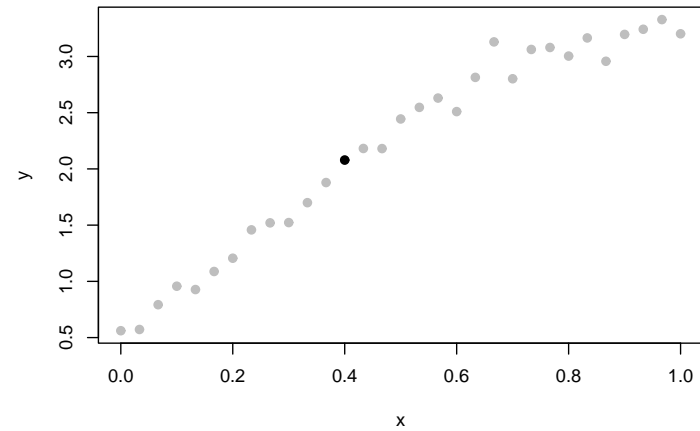
Thin plate regression splines



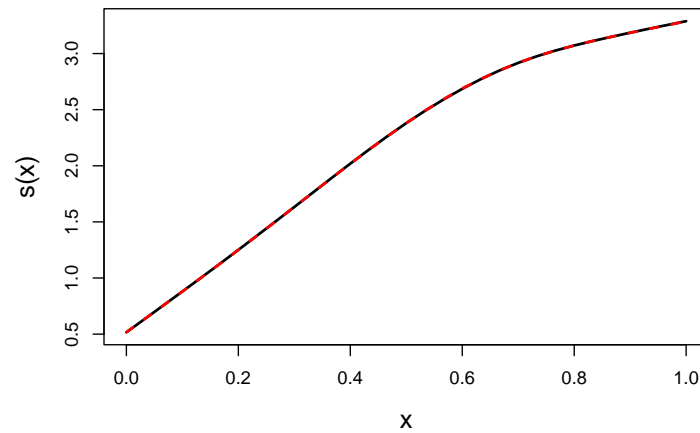
full spline basis



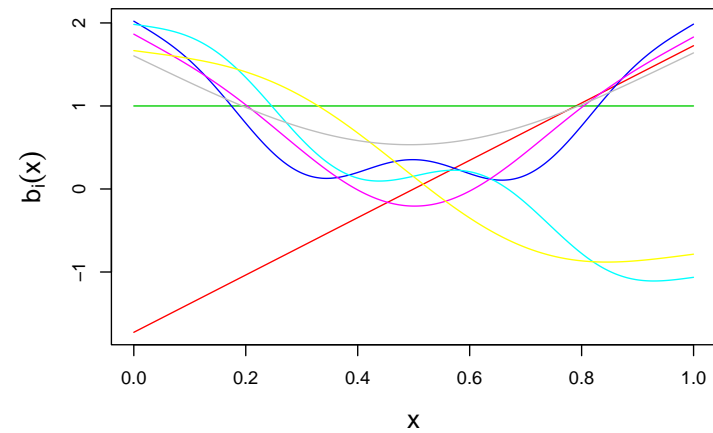
data to smooth



function estimate: full black, regression red

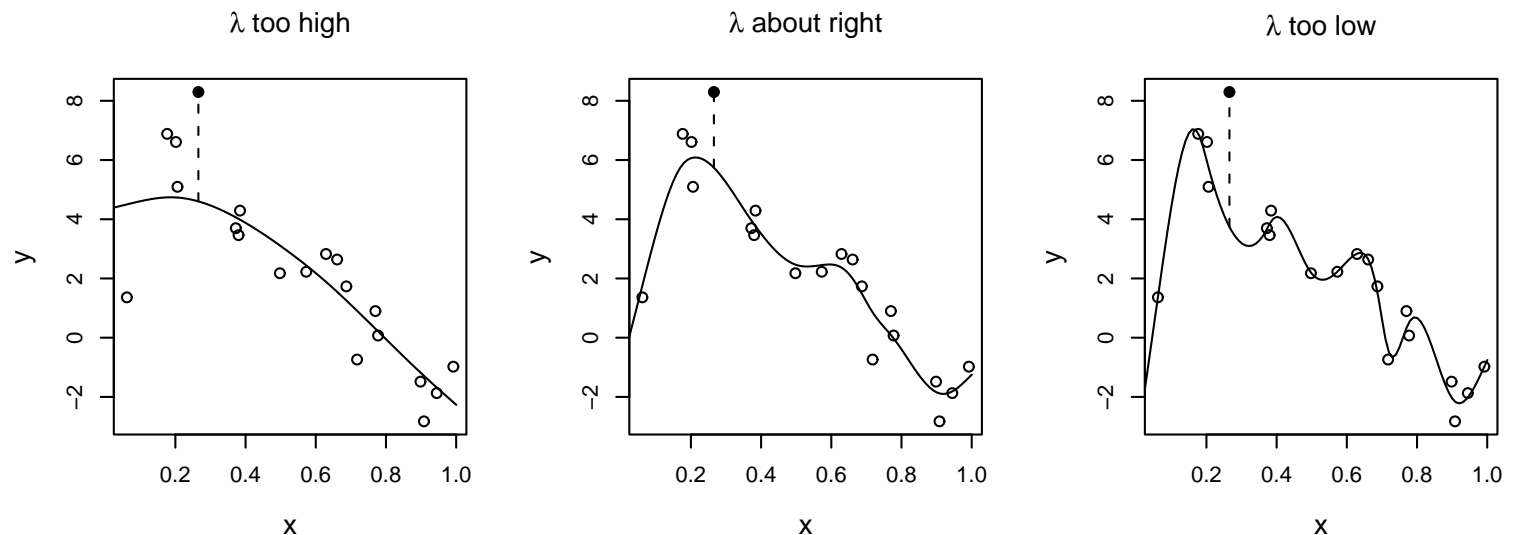


thin plate regression spline basis



Choosing smoothing parameters

- So far, we've seen how to reduce the cost of GAM modelling using low rank bases.
- It is also necessary to choose the degree of smoothing. Cross validation is one method, here's an illustration ...



Cross Validation

- ⑥ Ordinary cross validation minimises the mean square error in predicting each datum in the data set, using the model fitted to the data excluding this datum.
- ⑥ OCV is not invariant in a rather odd way. GCV is an invariant modification. Choose λ to minimize:

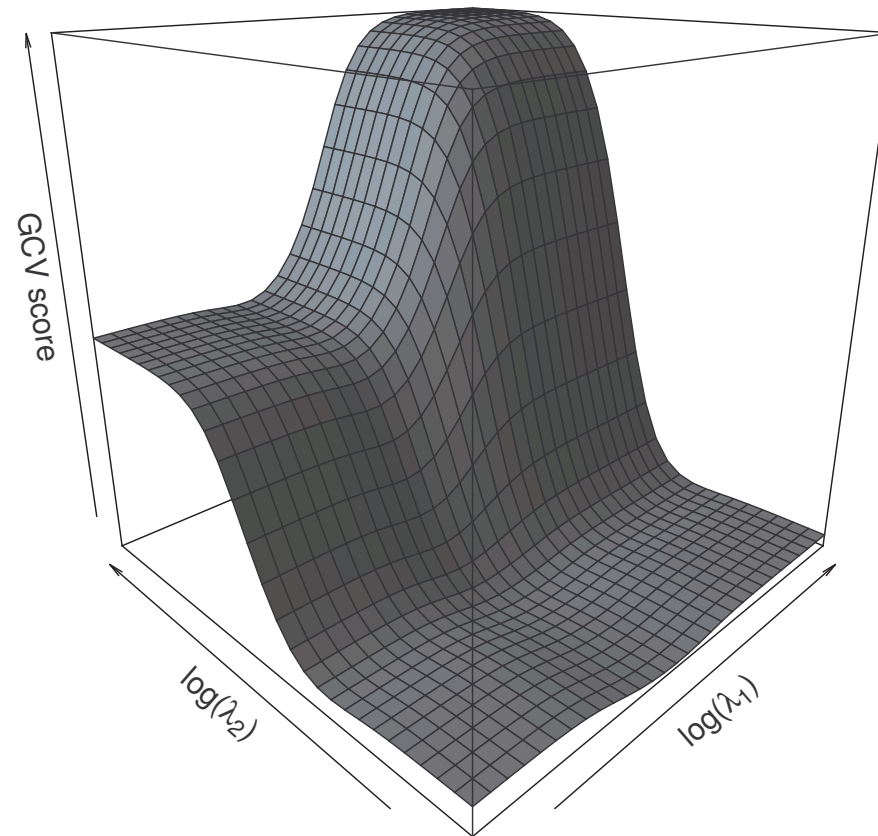
$$V(\boldsymbol{\lambda}) = \frac{n \|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}_{\lambda}\|^2}{[n - \text{tr}(\mathbf{A})]^2}$$

- ⑥ $\mathbf{A} = \mathbf{X}(\mathbf{X}^T\mathbf{X} + \sum_j \lambda_j \mathbf{S}_j)^{-1}\mathbf{X}^T$, the *influence matrix*.
 $\text{tr}(\mathbf{A}) = \text{tr}(\mathbf{F})$ where $\mathbf{F} = (\mathbf{X}^T\mathbf{X} + \sum_j \lambda_j \mathbf{S}_j)^{-1}\mathbf{X}^T\mathbf{X}$, the *degrees of freedom matrix*.

Example GCV function



GCV score for a 2 term GAM



GCV optimal model computation

- ⑥ $\mathcal{S} = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \sum_j \lambda_j \boldsymbol{\beta}^\top \mathbf{S}_j \boldsymbol{\beta}$, where \mathbf{X} is $n \times p$.
- ⑥ Form $\mathbf{X} = \mathbf{Q}\mathbf{R}$, where \mathbf{R} is $p \times p$; cols of $\mathbf{Q} \perp$.
- ⑥ Let $\mathbf{f} = \mathbf{Q}^\top \mathbf{y}$ and $\|\mathbf{r}\|^2 = \|\mathbf{y}\|^2 - \|\mathbf{f}\|^2$.
- ⑥ Then $\mathcal{S} = \|\mathbf{f} - \mathbf{R}\boldsymbol{\beta}\|^2 + \|\mathbf{r}\|^2 + \sum_j \lambda_j \boldsymbol{\beta}^\top \mathbf{S}_j \boldsymbol{\beta}$,

$$V(\boldsymbol{\lambda}) = \frac{n\|\mathbf{f} - \mathbf{R}\boldsymbol{\beta}\|^2 + n\|\mathbf{r}\|^2}{[n - \text{tr}(\mathbf{F})]^2},$$

where $\mathbf{F} = (\mathbf{R}^\top \mathbf{R} + \sum_j \lambda_j \mathbf{S}_j)^{-1} \mathbf{R}\mathbf{R}^\top$.

- ⑥ Cost after QR is down to $O(p^3)$.

Computation details

- ⑥ Posterior cov matrix is $\Sigma_{\beta} \propto (\mathbf{R}^T \mathbf{R} + \sum_j \lambda_j \mathbf{S}_j)^{-1}$.
- ⑥ $V(\lambda)$ optimization performed by Newton method.
- ⑥ Model flexibility enhances chance of co-linearity.
- ⑥ $\|\mathbf{f} - \mathbf{R}\boldsymbol{\beta}\|^2 + \sum_j \lambda_j \boldsymbol{\beta}^T \mathbf{S}_j \boldsymbol{\beta} = \left\| \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} \mathbf{R} \\ \mathbf{B} \end{bmatrix} \boldsymbol{\beta} \right\|^2$ where $\mathbf{B}^T \mathbf{B} = \sum_j \lambda_j \mathbf{S}_j$, so estimation and derivatives of $V(\boldsymbol{\lambda})$ are based on a (truncated?) SVD of $\begin{bmatrix} \mathbf{R}^T & \mathbf{B}^T \end{bmatrix}^T$.
- ⑥ Use of $\text{tr}(\mathbf{A}\mathbf{B}) = \text{tr}(\mathbf{B}\mathbf{A})$, minimum column $\sqrt{\mathbf{S}_j}$ etc conserves flops. See Wood (2004) JASA.

Chicago air pollution example

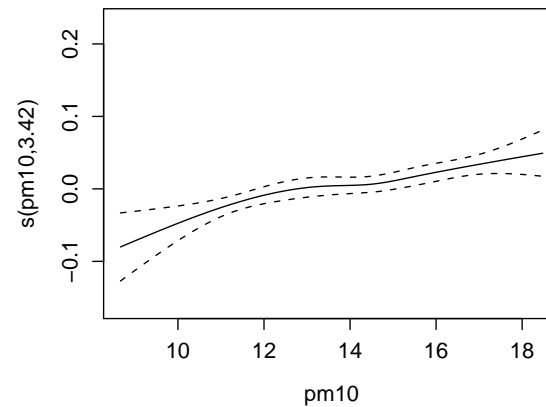
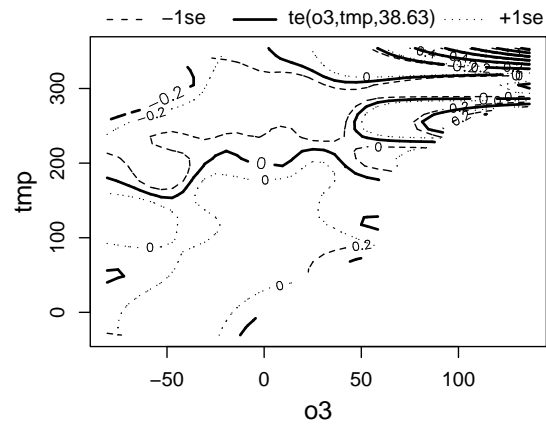
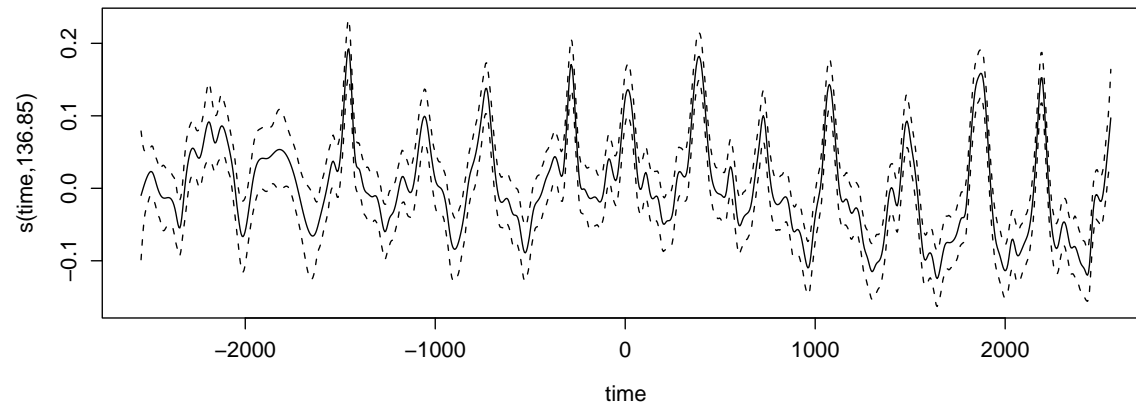
- ⑥ Around 5000 daily death rates, for Chicago, along with `time`, `ozone`, `pm10`, `tmp` (last 3 averaged over preceding 3 days). Peng and Welty (2004).

- ⑥ Appropriate GAM is: $\text{death}_i \sim \text{Poi}$,

$$\log\{\mathbb{E}(\text{death}_i)\} = f_1(\text{time}_i) + f_2(\text{ozone}_i, \text{tmp}_i) + f_3(\text{pm10}_i).$$

- ⑥ f_1 and f_3 penalized cubic regression splines, f_2 tensor product spline.
- ⑥ Estimation is by P-IRLS. Smoothing parameters can be selected by AIC for each working linear model in this iteration. 10s of minutes to fit on pentium 4.

Chicago air pollution fit



Huge data-sets

- ⑥ For fitting/GCV we need only have \mathbf{R} , \mathbf{f} and $\|\mathbf{r}\|^2$, which cost little storage. These can be found without ever forming the whole \mathbf{X} , which may be very large...
- ⑥ Suppose $\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \end{bmatrix}$, $\mathbf{X}_1 = \mathbf{Q}_1\mathbf{R}_1$ and $\mathbf{X}_2 = \mathbf{Q}_2\mathbf{R}_2$.
- ⑥ Form $\begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \end{bmatrix} = \mathbf{Q}\mathbf{R}$, and let $\mathbf{Q}^* = \begin{bmatrix} \mathbf{Q}_1^T & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_2^T \end{bmatrix} \mathbf{Q}$: then $\mathbf{X} = \mathbf{Q}^*\mathbf{R}$ is a QR decomposition!
- ⑥ \Rightarrow we can update \mathbf{R} , \mathbf{f} and $\|\mathbf{r}\|^2$ as new rows are added to \mathbf{X} without forming \mathbf{X} explicitly, or storing the intermediate QR factors.

Huge data example

- ⑥ 5 million data were simulated from

$$y_i = \mu_i(\mathbf{x}) + \epsilon_i, \quad \epsilon_i \text{ i.i.d. } N(0, \sigma^2)$$

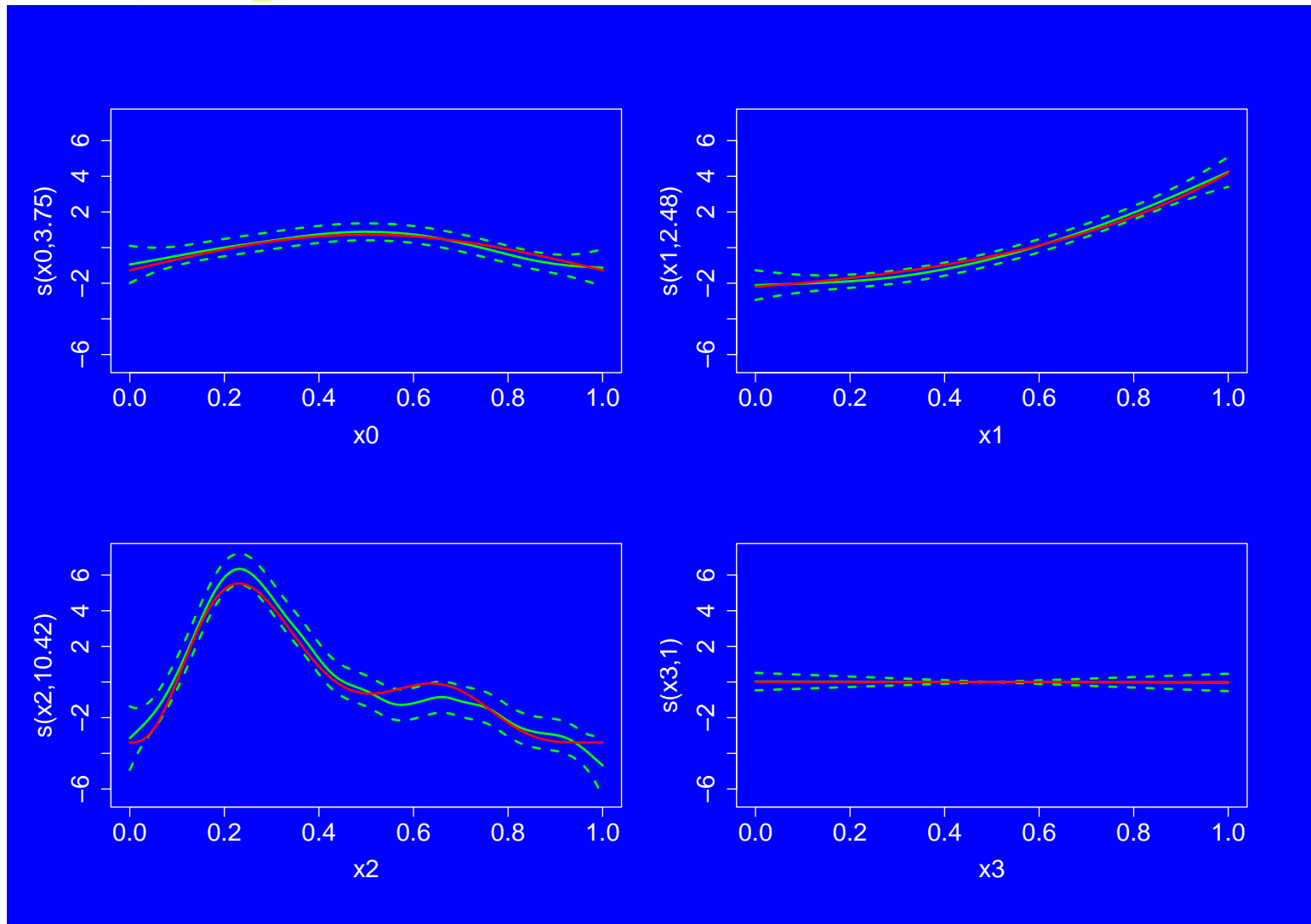
where $\mu_i(\mathbf{x}) = f_0(x_0) + f_2(x_2) + f_3(x_3) + f_4(x_4)$; the f_j are smooth and the x_{ij} are i.i.d. $U(0, 1)$.

- ⑥ The variance of the μ_i was around 10. $\sigma^2 = 10^5$. i.e. the data are rather noisy.
- ⑥ A 4 term additive model was fitted, with basis dimension 40 for each smooth.

Huge data example II

- ⑥ The full model matrix for this would require more than 6 Gb of storage.
- ⑥ Working in blocks of 10^5 observations, the QR update approach was used to accumulate the required fit information without needing the full model matrix.
- ⑥ Model setup (i.e QR accumulation) took 27 minutes (Pentium Xeon 1.6Ghz).
- ⑥ Model estimation (including GCV smoothing parameter selection) took 4.4 seconds.

Results: *Truth*, *Fit*



The End

1. GAMs offer a flexible approach to modelling very large complex datasets!
2. Most of the methods here can be found in **R** package **mgcv** (cran.r-project.org).
3. For more information and references see `mgcv` help files and/or
Wood SN, (2006) Generalized Additive Models: An Introduction with R. Chapman & Hall/ CRC press