INLA and other approaches to GAMs

Simon Wood

INLA: higher order marginal inference with GAMs

- So far we took a basically *empirical Bayes* approach to GAMs

$$y_i \sim \text{EF}(\mu_i, \phi) \quad g(\mu_i) = \sum_j f_j(x_{ji})$$

- Smooth functions f were represented using basis expansions of modest rank and complexity controlled by quadratic penalties induced by Gaussian smoothing priors.
- Smoothing parameters were estimated by Laplace approximate marginal likelihood, and further inference based on a Gaussian posterior approximation.
- What if the Gaussian approximation is poor? Two options
  1. Stochastic simulation (see later).
  2. Rue et al. (2009) JRSSB 71:319-392 show how to produce much more accurate approximations to marginal distributions of the model coefficients.

Gaussian posterior approximation

- $\pi(\boldsymbol{\beta}|\mathbf{y},\boldsymbol{\theta}) \propto \pi(\mathbf{y}|\boldsymbol{\beta})\pi(\boldsymbol{\beta}|\boldsymbol{\theta})$.
- Here $\pi(\boldsymbol{\beta}|\boldsymbol{\theta}) = \text{MVN}(\mathbf{0}, \mathbf{S}_\theta^-)$.
- $\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\text{argmax}}\ \pi(\boldsymbol{\beta}|\mathbf{y},\boldsymbol{\theta}) = \underset{\boldsymbol{\beta}}{\text{argmax}}\ l(\boldsymbol{\beta})^* - \frac{1}{2}\boldsymbol{\beta}^{\mathrm{T}}\mathbf{S}_\theta\boldsymbol{\beta}$.
- Define log posterior Hessian, $\mathbf{H}_\theta = -\left.\frac{\partial^2 l}{\partial\boldsymbol{\beta}\partial\boldsymbol{\beta}^{\mathrm{T}}}\right|_{\hat{\boldsymbol{\beta}}} + \mathbf{S}_\theta$.
- Second order Taylor expansion of log joint density $\Rightarrow$

$$\begin{aligned}\pi(\boldsymbol{\beta}|\mathbf{y},\boldsymbol{\theta}) &\simeq k\exp\left\{-(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})^{\mathrm{T}}\mathbf{H}_\theta(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})/2\right\}\\ &= \text{MVN}(\hat{\boldsymbol{\beta}}, \mathbf{H}_\theta^{-1})\\ &\equiv \pi_g(\boldsymbol{\beta}|\mathbf{y},\boldsymbol{\theta}),\quad \text{say.}\end{aligned}$$

$^*l(\boldsymbol{\beta}) = \log\pi(\mathbf{y}|\boldsymbol{\beta})$

Laplace approximation

Consider approximating marginal likelihood...

$$
\begin{aligned}
\pi(\mathbf{y}|\boldsymbol{\theta}) &= \int \pi(\mathbf{y}|\boldsymbol{\beta})\pi(\boldsymbol{\beta}|\boldsymbol{\theta})d\boldsymbol{\beta} \\
&\simeq \int \exp\left\{l(\hat{\boldsymbol{\beta}}) + \log \pi(\hat{\boldsymbol{\beta}}|\boldsymbol{\theta}) - (\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})^{\mathrm{T}}\mathbf{H}_\theta(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})/2\right\} d\boldsymbol{\beta} \\
&= \pi(\mathbf{y}|\hat{\boldsymbol{\beta}})\pi(\hat{\boldsymbol{\beta}}|\boldsymbol{\theta}) \int \exp\left\{-(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})^{\mathrm{T}}\mathbf{H}_\theta(\boldsymbol{\beta}-\hat{\boldsymbol{\beta}})/2\}\right\} d\boldsymbol{\beta} \\
&= \frac{\pi(\mathbf{y}|\hat{\boldsymbol{\beta}})\pi(\hat{\boldsymbol{\beta}}|\boldsymbol{\theta})(2\pi)^{p/2}}{|\mathbf{H}_\theta|^{1/2}} \quad \text{(Laplace approx.)} \\
&= \frac{\pi(\mathbf{y},\hat{\boldsymbol{\beta}}|\boldsymbol{\theta})}{\pi_g(\hat{\boldsymbol{\beta}}|\mathbf{y},\boldsymbol{\theta})}
\end{aligned}
$$

...i.e. joint density over Gaussian approx. posterior, both at $\hat{\boldsymbol{\beta}}$.

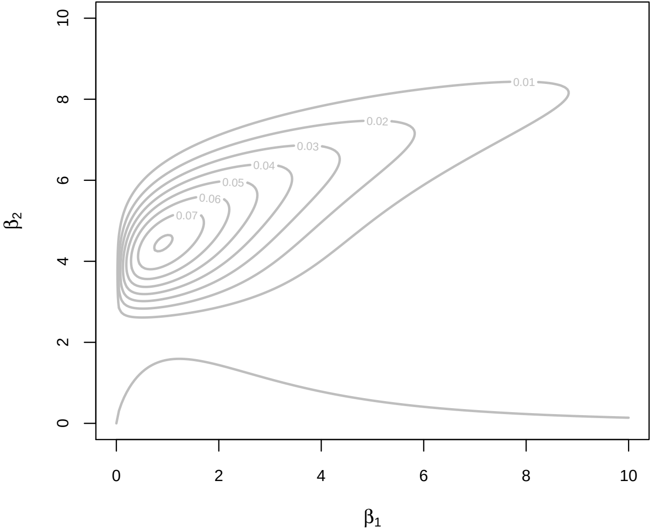
Gaussian posterior accuracy and INLA

- When $n/p \to \infty$ the approximation $\pi_g(\hat{\boldsymbol{\beta}}|\mathbf{y}, \boldsymbol{\theta})$ is usually quite accurate, at least if $p = o(n^{1/3})$.
- But not always true and anyway it deteriorates in the tails.
- Integrated Nested Laplace Approximation (INLA) makes clever use of partial Gaussian approximations to improve the approximation of marginal posteriors

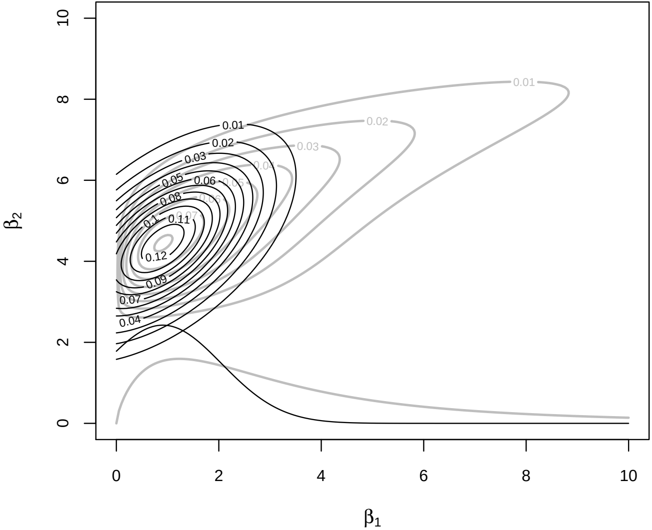
$$\pi(\beta_i|\mathbf{y}, \boldsymbol{\theta})$$

- First consider an example, illustrating how π_g performs…

Posterior $\pi(\boldsymbol{\beta}|\mathbf{y}, \boldsymbol{\theta})$ and marginal $\pi(\beta_1|\mathbf{y}, \boldsymbol{\theta})$



Basic Gaussian approximation, $\pi_g(\boldsymbol{\beta}|\mathbf{y}, \boldsymbol{\theta})$



The basic INLA idea

- The key idea in INLA is

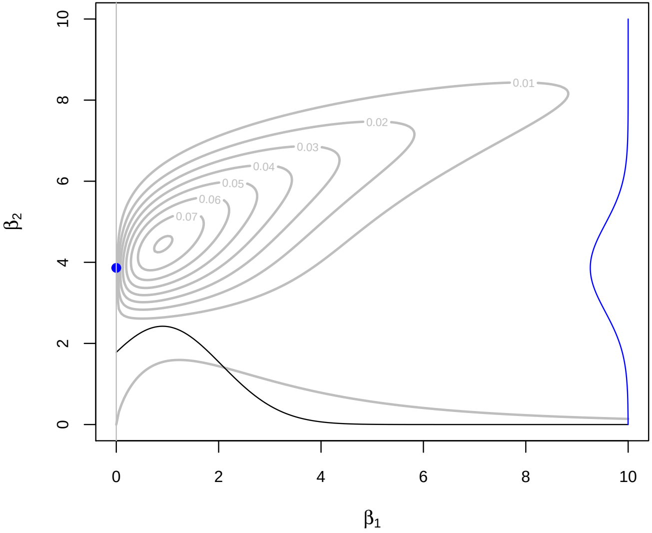
$$\pi(\beta_i|\mathbf{y},\boldsymbol{\theta}) = \frac{\pi(\tilde{\boldsymbol{\beta}},\mathbf{y},\boldsymbol{\theta})}{\pi(\tilde{\boldsymbol{\beta}}_{-i}|\beta_i,\mathbf{y},\boldsymbol{\theta})} \simeq \frac{\pi(\tilde{\boldsymbol{\beta}},\mathbf{y},\boldsymbol{\theta})}{\pi_{gg}(\tilde{\boldsymbol{\beta}}_{-i}|\beta_i,\mathbf{y},\boldsymbol{\theta})} = \tilde{\pi}(\beta_i|\mathbf{y},\boldsymbol{\theta})$$

  where π_{gg} is some Gaussian approximation to $\pi(\tilde{\boldsymbol{\beta}}_{-i}|\beta_i,\mathbf{y},\boldsymbol{\theta})$ and $\tilde{\boldsymbol{\beta}}$ maximizes the joint density subject to constraint $\tilde{\beta}_i = \beta_i$.
- For π_{gg} we could use the distribution of $\boldsymbol{\beta}_{-i}|\beta_i$ implied by $\pi_g(\boldsymbol{\beta}|\mathbf{y},\boldsymbol{\theta})$. This has a fixed covariance matrix and, writing $\boldsymbol{\Sigma} = \mathbf{H}_\theta^{-1}$, a mean $\tilde{\boldsymbol{\beta}}_{-i} = \hat{\boldsymbol{\beta}}_{-i} + \boldsymbol{\Sigma}_{-i,i}\boldsymbol{\Sigma}_{i,i}^{-1}(\beta_i - \hat{\beta}_i)$.
- Hence the simplest version of INLA could just use

$$\tilde{\pi}(\beta_i|\mathbf{y},\boldsymbol{\theta}) \propto \pi(\tilde{\boldsymbol{\beta}}(\beta_i),\mathbf{y},\boldsymbol{\theta})$$

  and renormalize.

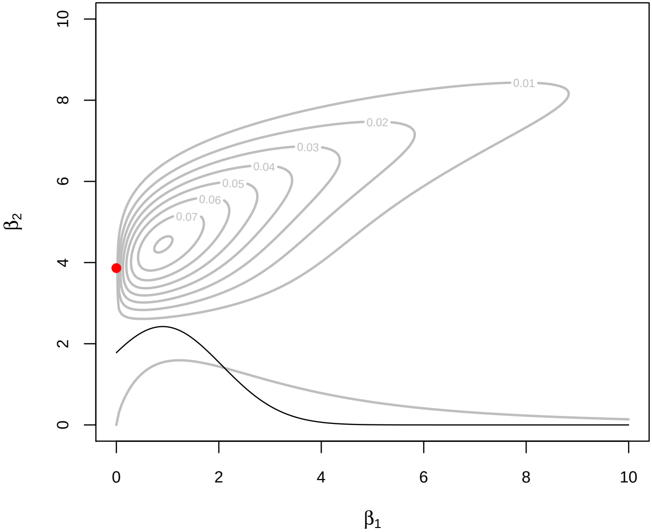
Most basic INLA $\tilde{\pi}(\beta_1|\mathbf{y}, \boldsymbol{\theta})$



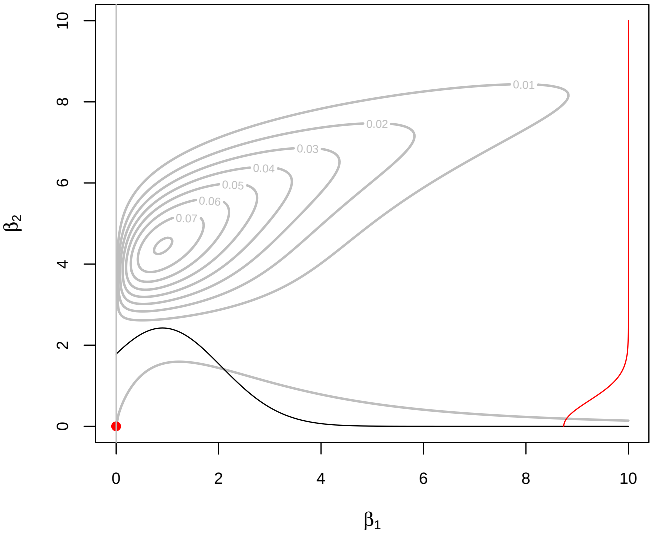
Rue et al. (2009) INLA

- If $\tilde{\boldsymbol{\beta}}$ were the actual maximiser of $\pi(\boldsymbol{\beta}, \mathbf{y}, \boldsymbol{\theta})$ given $\tilde{\beta}_i = \beta_i$ and $\mathbf{H}_{-i,-i}$ were the corresponding Hessian w.r.t. $\boldsymbol{\beta}_{-i}$, then we could set $\pi_{gg} = \text{MVN}(\tilde{\boldsymbol{\beta}}_{-i}, \mathbf{H}^{-1}_{-i,-i})$.
- Then $\tilde{\pi}(\beta_i|\mathbf{y}, \boldsymbol{\theta})$ is the Laplace approx. to $\int \pi(\boldsymbol{\beta}|\mathbf{y}, \boldsymbol{\theta}) d\boldsymbol{\beta}_{-i}$ …and INLA is rather accurate!
- But $\text{MVN}(\tilde{\boldsymbol{\beta}}_{-i}, \mathbf{H}^{-1}_{-i,-i})$ is too expensive to be practical. It has to be approximated.
- Rue et al. (2009) use $\tilde{\boldsymbol{\beta}}_{-i} = \hat{\boldsymbol{\beta}}_{-i} + \boldsymbol{\Sigma}_{-i,i}\Sigma^{-1}_{i,i}(\beta_i - \hat{\beta}_i)$ implied by π_g, and an approximation to the required $\log|\mathbf{H}_{-i,-i}|$.

Published INLA $\tilde{\pi}(\beta_1|\mathbf{y},\boldsymbol{\theta})$ — approximate $\tilde{\boldsymbol{\beta}}$



Ideal INLA $\tilde{\pi}(\beta_1|\mathbf{y}, \boldsymbol{\theta})$ — exact $\tilde{\boldsymbol{\beta}}$



The point is …

- Using easily computed Gaussian approximations we obtain marginal posterior approximations much more accurate than naive direct use of posterior Gaussian approximation.
- The improved accuracy accrues from several features of

$$\tilde{\pi}(\beta_i|\mathbf{y}, \boldsymbol{\theta}) = \frac{\pi(\tilde{\boldsymbol{\beta}}, \mathbf{y}, \boldsymbol{\theta})}{\pi_{gg}(\tilde{\boldsymbol{\beta}}_{-i}|\beta_i, \mathbf{y}, \boldsymbol{\theta})}$$

  1. we only evaluate the Gaussian approximation at its mean, not out in its inaccurate tails.
  2. the approximation error enters multiplicatively, rather than growing into the tails
  3. a univariate marginal is easy to renormalize.

- But what about $\boldsymbol{\theta}$ and where is the integration?

Uncertainty in $\boldsymbol{\theta}$

- A Laplace approximation is used for the posterior of $\boldsymbol{\theta}$

$$\tilde{\pi}(\boldsymbol{\theta} \mid \mathbf{y}) \propto \frac{\pi(\hat{\boldsymbol{\beta}}, \mathbf{y}, \boldsymbol{\theta})}{\pi_g(\hat{\boldsymbol{\beta}} \mid \mathbf{y}, \boldsymbol{\theta})}$$

- Then fairly crude quadrature[†] is used to integrate out $\boldsymbol{\theta}$

$$\tilde{\pi}(\beta_i \mid \mathbf{y}) = \int \tilde{\pi}(\beta_i \mid \boldsymbol{\theta}, \mathbf{y}) \tilde{\pi}(\boldsymbol{\theta} \mid \mathbf{y}) d\boldsymbol{\theta}$$

  and $\tilde{\pi}(\theta_i \mid \mathbf{y}) = \int \tilde{\pi}(\boldsymbol{\theta} \mid \mathbf{y}) d\boldsymbol{\theta}_{-i}$.

- Or skip integration and just use the posterior mode $\hat{\boldsymbol{\theta}}$.

[†]Numerical integration based on evaluating the integrand on some grid and forming a weighted sum of the evaluations.

Computational efficiency and approximating $\log|\mathbf{H}_{-i,-i}|$

- The *key* step in INLA is the approximation π_{gg}. It *must* be computationally efficient.
- Rue et al. (2009) use the conditional mode $\tilde{\boldsymbol{\beta}}(\beta_i)$ implied by π_g, and one of two approximations to $\log|\mathbf{H}_{-i,-i}|$:
  1. Approximate $\log|\mathbf{H}_{-i,-i}|$ by its first order Taylor expansion around $\hat{\boldsymbol{\beta}}$. Efficient default setting — properties unclear.
  2. Use the heuristic that only elements of $\boldsymbol{\beta}_{-i}$ that are highly enough correlated with β_i according to π_g need to be considered when updating from $\log|\mathbf{H}_\theta|$ to $\log|\mathbf{H}_{-i,-i}|$.

  These are efficient when $\mathbf{H}_\theta$ is a high rank sparse matrix, as it is in the INLA software, but not if $\mathbf{H}_\theta$ is dense.
- Often it makes sense to use an intermediate rank model representation and a dense $\mathbf{H}_\theta$. Then 1 and 2 impractical.

An alternative $\log|\mathbf{H}_{-i,-i}|$ approximation

1. Given a Cholesky factor $\mathbf{R}$ of $\mathbf{H}_\theta$, cheaply update it to the Cholesky factor of $\tilde{\mathbf{H}}_0 = \mathbf{H}_\theta[-i,-i]$.
2. Given this factor, cheaply run several Newton steps with fixed $\tilde{\mathbf{H}}_0$ to find the numerically exact $\tilde{\boldsymbol{\beta}}(\beta_i)$.
3. Approximate $\mathbf{H}_{-i,-i}$ at $\tilde{\boldsymbol{\beta}}(\beta_i)$ by a BFGS[‡] update of $\tilde{\mathbf{H}}_0$ using a small step from $\tilde{\boldsymbol{\beta}}(\beta_i)$ towards $\hat{\boldsymbol{\beta}}$. This allows efficient computation of the corresponding $\log|\mathbf{H}_{-i,-i}|$.

- The approach works for sparse or dense $\mathbf{H}_\theta$. An alternative version avoids the need for an explicit Cholesky update.
- As with the original method, judicious use of interpolation avoids evaluating at too many β_i values.
- The log determinant update has some theory…

[‡] An approximate Hessian update used in quasi-Newton optimization

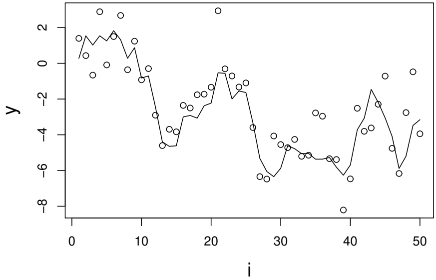
Update properties

Theorem

Let $\tilde{\mathbf{H}}_0$ *and* $\tilde{\mathbf{H}}$ *be respectively the initial Hessian and true Hessian with respect to* $\boldsymbol{\beta}_{-i}$ *at* $\tilde{\boldsymbol{\beta}}(\beta_i)$*, and assume that* $\log \pi(\boldsymbol{\beta}, \mathbf{y}, \boldsymbol{\theta})$ *is regular with bounded third derivative. Let* $\tilde{\mathbf{H}}_1$ *denote the BFGS update of* $\tilde{\mathbf{H}}_0$ *based on a step* $h\boldsymbol{\Delta}$ *from* $\tilde{\boldsymbol{\beta}}$ *where* $\|\boldsymbol{\Delta}\| = 1$*. Then* $|\tilde{\mathbf{H}}_1| \in [|\tilde{\mathbf{H}}_0| + O(h), |\tilde{\mathbf{H}}| + O(h)]$.

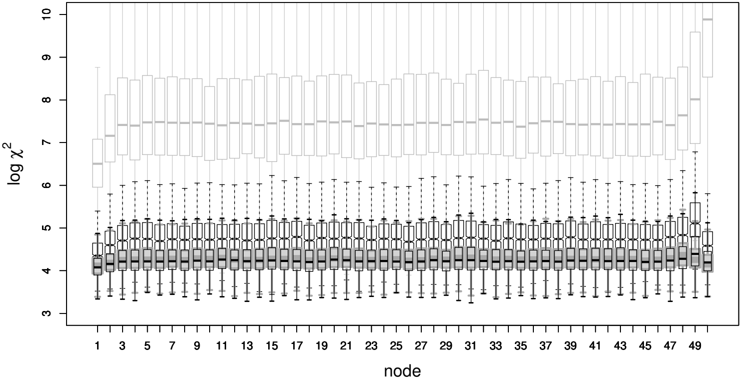
- See Wood (2019, *Biometrika*) for proof and method details
- Not all quasi-Newton updates have this property, nor does the Rue et al. (2009) default method.

Test example from Rue et al. (2009) §5.1



- $y_i - f_i \sim t_3$ where $f_i - \mu \sim N\{\phi(f_{i-1} - \mu), 1\}$ if $i = 2, \ldots, 50$,$f_1 - \mu \sim N(0, 1)$, $\phi = 0.85$ and $\mu \sim N(0, 1)$.
- Investigate goodness of fit of various INLA approximations to long Gibbs sampling runs over 1000 replicates.

Test results

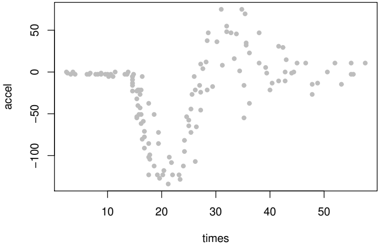


- Box-plots over 1000 reps of fit statistic - small is good.
- Black - Rue et al. expensive. Grey filled - new method. Dashed/notched Rue et al. default. Grey open - direct π_g.
- Rue et al. expensive and new method indistinguishable.

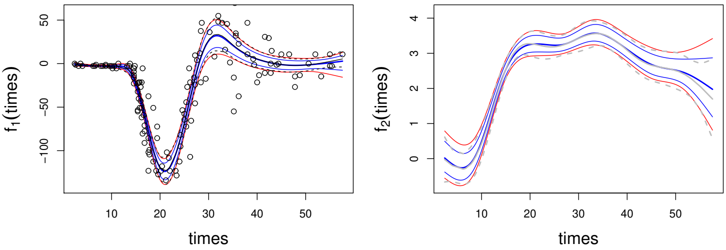
An example

- Method implemented in `mgcv::ginla` in R.
- In many real examples π_g is actually rather good, and `ginla` merely serves to confirm this!
- But it makes a difference when modelling the following over-used data with the model

$$\texttt{accel}_i \sim N\left(f_1(\texttt{times}_i), e^{2f_2(\texttt{times}_i)}\right)$$



Example results



- Solid and dashed are mean and 95% intervals from π_g.
- Blue are mean and 90% intervals, red are 95% intervals, both from `ginla`.

INLA advantages and software

- The major advantage to the INLA approach is that computation can efficiently exploit sparse matrices.
- This allows inference with large sparse *Gaussian Markov Random Fields*[§].
- Such models are especially useful in spatial settings where there is short range stochastic dependency (autocorrelation) to model.
- The INLA software is the major implementation built on sparse methods: see `www.r-inla.org`.
- `ginla` in `mgcv` offers a simple implementation for the non-sparse case.

[§] Basically a model with a Gaussian smoothing prior precision matrix that is sparse – i.e. mostly zeroes.

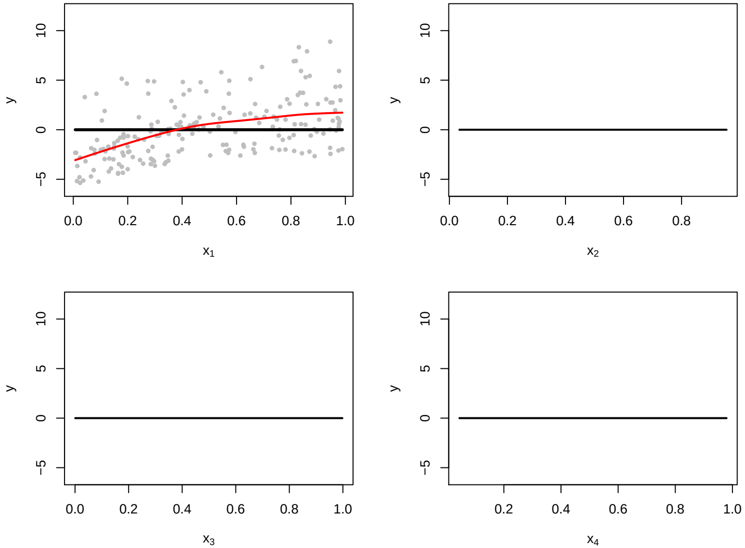
Other approaches to GAM estimation

- These slides have concentrated on quite statistical approaches to GAMs but there are other estimation methods with more of a learning algorithm feel.
- For example, *backfitting* and *boosting* both approach estimation by iterative smoothing of residuals.
- They offer advantages in terms of algorithmic modularity and efficiency, but some aspects of inference become more difficult.
- Backfitting is original method used in Hastie and Tibshirani's (1986, 1990) pioneering work on GAMs.
- Boosting is notable for providing a rather integrated method for model term selection.

Backfitting algorithm

- Estimate $y_i = \alpha + \sum_{j=1}^{m} f(x_{ji}) + \epsilon_i$. Let $\mathbf{f}_j = (f_j(x_1), f_j(x_2), \ldots)^\mathrm{T}$.
- Set $\hat{\alpha} = \bar{y}$, $\mathbf{f}_j = \mathbf{0}\ \forall\ j$ and repeat to convergence:
  For $j = 1, \ldots, m$
  1. Calculate *partial residuals* $\mathbf{e}_j = \mathbf{y} - \hat{\alpha} - \sum_{k \neq j} \mathbf{f}_k$
  2. Set $\mathbf{f}_j$ to the result of smoothing $\mathbf{e}_j$ w.r.t. $\mathbf{x}_j$.
- A weighted version can be used on the working penalized linear model when iteratively fitting a GAM to non-Gaussian data.
- Notice we could use any smoother at step 2: e.g. spline, local regression, running mean etc. although for some we might have to subtract its mean from $\mathbf{f}_j$ to ensure the smooth stays centred.
- A drawback is that it is not clear how to select smoothing parameters. See Hastie and Tibshirani (1990) *Generalized Additive Models* and R package `gam` for more.

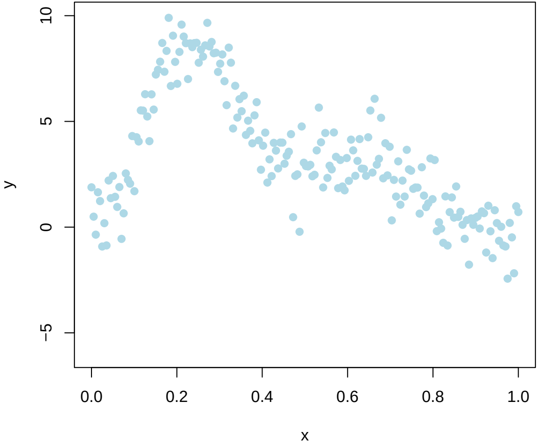
Backfitting $y_i = \alpha + \sum_{j=1}^{4} f(x_{ji}) + \epsilon_i$



Boosting

- Idea in one dimension, with least squares loss:
  1. Construct a low degree of freedom linear 'base smoother', e.g. $\hat{\boldsymbol{\mu}} = \mathbf{A}\mathbf{y}$, where $\mathbf{A} = \mathbf{X}(\mathbf{X}^{\mathrm{T}}\mathbf{X} + \lambda_{\mathrm{big}}\mathbf{S})^{-1}\mathbf{X}^{\mathrm{T}}$.
  2. Initialize $\hat{\mathbf{f}} = \mathbf{0}$ and then iterate $\hat{\mathbf{f}} \leftarrow \hat{\mathbf{f}} + \mathbf{A}(\mathbf{y} - \hat{\mathbf{f}})$.
- Note that if we iterate for ever we end up with the p degrees of freedom fit $\hat{\mathbf{f}} = \mathbf{X}(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y}$, despite the summed components each having very low EDF.
- Need a stopping rule and further inference not so easy.
- One option is the sort of bootstrap cross-validation and inference suggested for the Lasso.

Basic boosting idea



Gradient boosting with selection of multiple terms

- Consider a model with a log likelihood l and multiple smooth terms, f_j, in a linear predictor $\boldsymbol{\eta}$.
- Set up base smoothers (hat matrix $\mathbf{A}_j$) for each f_j potentially in the model. Iterate[¶] …
  1. Compute $e_i = -\mathrm{d}l/\mathrm{d}\eta_i$.
  2. For all j compute $\tilde{\mathbf{f}}_j = \mathbf{A}_j\mathbf{e}$ and find $\hat{\alpha}_j = \mathrm{argmax}_\alpha l(\boldsymbol{\eta} + \alpha\tilde{\mathbf{f}}_j)$.
  3. Find $k = \mathrm{argmax}_j l(\boldsymbol{\eta} + \hat{\alpha}_j\tilde{\mathbf{f}}_j)$.
  4. Set $\boldsymbol{\eta} \leftarrow \boldsymbol{\eta} + \hat{\alpha}_k\tilde{\mathbf{f}}_k$, and add k to set of selected terms.
- Notes:
  - This is a very efficient forward selection method, but contains no means for going backwards. Again we need a stopping rule, and have to bootstrap for further inference.
  - We have an ascent direction at step 2 because we are multiplying the gradient by a positive definite matrix.
  - Without the $\hat{\alpha}$ search, term selection is sensitive to base EDF.

[¶] Schmid and Hothorn (2008) CSDA; Mayr et al. (2012) Applied Statistics