LMU

LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

INSTITUT FÜR STATISTIK

Fabian Scheipl & Thomas Kneib

Locally Adaptive Bayesian P-Splines with a Normal-Exponential-Gamma Prior

Technical Report Number 022, 2008
Department of Statistics
University of Munich

http://www.stat.uni-muenchen.de

Locally Adaptive Bayesian P-Splines with a Normal-Exponential-Gamma Prior

Fabian Scheipl, Thomas Kneib

March 19, 2008

The necessity to replace smoothing approaches with a global amount of smoothing arises in a variety of situations such as effects with highly varying curvature or effects with discontinuities. We present an implementation of locally adaptive spline smoothing using a class of heavy-tailed shrinkage priors. These priors utilize scale mixtures of normals with locally varying exponential-gamma distributed variances for the differences of the P-spline coefficients. A fully Bayesian hierarchical structure is derived with inference about the posterior being based on Markov Chain Monte Carlo techniques. Three increasingly flexible and automatic approaches are introduced to estimate the spatially varying structure of the variances. In an extensive simulation study, the performance of our approach on a number of benchmark functions is shown to be at least equivalent, but mostly better than previous approaches and fits both functions of smoothly varying complexity and discontinuous functions well. Results from two applications also reflecting these two situations support the simulation results.

1. Introduction

In many regression applications, the assumption of a linear dependence of the response on predictor variables is inappropriate. One appealing solution to the problem of modeling smooth functions of an unknown shape, that is, fitting models of the form

$$\boldsymbol{y} = f(\boldsymbol{x}) + \boldsymbol{\varepsilon}; \quad \boldsymbol{\varepsilon} \sim \mathcal{N}_n(0, \sigma_\varepsilon^2 \boldsymbol{I}),$$

where $f(\cdot)$ is a smooth function of a covariate x, is P-spline smoothing [Eilers and Marx, 1996]. The idea behind this approach is conceptually simple: The unknown function is approximated by a piecewise polynomial function subject to some differentiability constraints at the interval boundaries. The resulting function can be represented as a linear combination of B-spline basis functions, i.e. basis functions with local support. The number of basis functions must be large enough to allow for sufficient flexibility in the shape of the estimated function. However, due to the high dimension of the basis, an unregularized fit would result in a very variable estimate. In order to avoid this overfitting problem, the basis coefficients are penalized to enforce smoothness of the resulting fit. Let $\boldsymbol{X}$ denote the matrix of the j basis functions, evaluated at $\boldsymbol{x}$. The objective function for the P-spline fit is then the penalized least squares criterion

$$||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}||^2 + \frac{1}{\tau^2}||\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}||^2 \rightarrow \min_{\boldsymbol{\beta}},$$

where $\boldsymbol{\Delta}^{(d)}$ of dimension $(j-d) \times j$ is the d^{th}-degree difference operator matrix and τ^2 is the smoothing parameter controlling the amount of penalization. In effect, this form of penalization penalizes deviations of the fitted curve from a $(d-1)$-degree polynomial [Eilers and Marx, 1996] since the d^{th} order derivative of B-splines essentially depends on d^{th} order differences. From a Bayesian perspective, d^{th} order differences correspond to a Gaussian random walk prior of order d for the vector $\boldsymbol{\beta}$ [Lang and Brezger, 2004].

For functions with locally varying complexity (e.g. oscillations with varying frequency and/or amplitude, or functions with discontinuities), a global penalty with constant smoothing parameter over the range of $\boldsymbol{x}$ is inappropriate, as it would lead to overfitting in the smooth parts of the function and underfitting at the more wiggly or discontinuous parts of the function. This problem can be tackled by introducing a penalty that varies spatially in order to reflect the spatial heterogeneity of the function [Lang et al., 2002]. Previous suggestions for locally adaptive smoothing include Bayesian and frequentist approaches that allow for (smoothly varying) spatial heterogeneity by fitting a smooth penalty function $\tau(x)$ represented as a second P-spline [Ruppert and Carroll, 2000, Baladandayuthapani et al., 2005, Krivobokova et al., 2007], or reweighting the individual penalty terms so that
$\left(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}\right)_i \sim \mathcal{N}(0, \frac{\tau^2}{\delta_i})$ [Lang and Brezger, 2004], with $\delta_i \sim \Gamma(\frac{\nu}{2}, \frac{\nu}{2})$ leading to a marginally t-distributed random walk prior, as well as knot-selection based approaches [Denison et al., 1998, Biller, 2000].

Similar to Lang and Brezger [2004], the main idea of our fully Bayesian approach is to replace the homoskedastic Gaussian random walk prior for $\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}$ with a heteroscedastic heavy-tailed random walk prior. Unlike Lang and Brezger [2004], we assume piecewise constant variances, however, and, unlike the solutions based on the original idea by Ruppert and Carroll [2000], we make no smoothness assumptions about the shape of the resulting variance step function. The prior we use is a scale mixture of normals introduced by Griffin and Brown [2007]. The variance of the normal follows an exponential distribution with a gamma-distributed rate resulting in a Normal-Exponential-Gamma (NEG) prior. This mixture distribution is strongly peaked in the origin and has heavy tails leading to advantageous adaptivity properties.

We propose a hierarchy of estimation schemes based on Markov chain Monte Carlo simulation (MCMC) techniques that introduce increasing flexibility in estimating the variance step function. Starting with fixed number and locations of the changepoints, we then introduce a more flexible alternative in which the locations of the changepoints are estimated as well. In a final step, the number of steps is included as a further unknown parameter, leading to a reversible jump MCMC algorithm. All the NEG-based algorithms are implemented in R [R Development Core Team, 2007], the code is available from the first author.

Results from an extensive simulation study show that these approaches, unlike previous suggestions, can deal equally well with both smoothly varying local complexity and functions with discontinuities and usually converge very fast due to the excellent mixing properties of the sampling steps we use. For all practical purposes, the reversible jump algorithm is fully automatic in the sense that results are very robust against changes in the only two hyperparameters supplied by the user. The applicability of the proposed approach in both situations with discontinuous functions and functions with varying curvature is demonstrated in applications on the estimation of fractionation curves in quality control of cDNA microarray experiments and the analysis of CP6 sales data.

The rest of the paper is structured as follows: Section 2 describes the hierarchy of our model and discusses the three implementations of our approach. Results of a fairly extensive simulation study are given in section 3, followed by two exemplary applications to real data in section 4.

2. Models and Algorithms

Conventional Bayesian P-spline smoothing [Lang and Brezger, 2004] is based on a homoscedastic Gaussian prior for the d^{th} differences $\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}$ of the j P-spline coefficients $\boldsymbol{\beta}$: $\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta} \sim N(\mathbf{0}, \tau^2 \boldsymbol{I}_{j-d})$. This corresponds to a ridge-type regularization of the fitted function, leading to a proportional shrinkage of the unregularized random walk. An improved prior distribution, however, should be designed to allow for high penalisation in areas with low variability and, vice versa, low penalisation in areas with high curvature or discontinuities. Translated to the form of the prior distribution this means that a prior with a peak at zero on the one hand but heavy tails on the other hand should be considered. Such types of priors have received considerable attention in recent years in applications on variable selection and regularisation in high-dimensional regression models [Griffin and Brown, 2005, Park and Casella, 2006]. One particularly promising candidate is the Normal-Exponential-Gamma (NEG) prior by Griffin and Brown [2007] that combines the desired properties with computational convenience in a hierarchical Bayesian updating scheme.

The prior $\pi_\tau(\tau^2|z)$ for $\tau_b^2,\ b = 1, \ldots, B$ is assumed to follow an exponential distribution with rate $z_b,\ b = 1, \ldots, B$. This rate, in turn, is assigned a $\Gamma(a_z, b_z)$-prior. Following Griffin and Brown [2007], we set $a_z = 0.5$, since a sufficiently flexible family of distributions is obtained by letting b_z vary all by itself. The prior for b_z is a discrete uniform distribution on a $\log_{10}$-regular grid with 550 values between 10^{-3} and 10^5. The resulting scale mixture NEG-prior for $\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}$ has the desired properties: Its mass is concentrated around zero, with a finite spike in the origin, leading to the desired regularization properties, and yet has heavy tails which allow for the possibility of large jumps in the random walk and therefore sudden jumps or curvature changes of the fitted function.

Griffin and Brown [2007] show that the maximum a posteriori (MAP) estimate for $\boldsymbol{\beta}$ based on this hierarchy also fulfills the so-called oracle property since the derivative of the scale-mixture prior tends to zero for increasing $|\beta|$. It is reasonable to assume that posterior means based on a prior with this desirable property also benefit from this fact and our simulation results (Section 3) confirm this intuition: Using this hierarchy, we obtain a strong shrinkage of $\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}$ where differences are small, increasing the smoothness of the fitted curve, while simultaneously allowing faithful modeling of jumps or sudden curvature changes.

The assumption of a homogeneous random walk on the coefficients is obviously problematic for functions with locally varying complexity. To further increase adaptivity, we replace the conventional homoscedastic prior for the d^{th} differences $\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}$ of the j P-spline coefficients $\boldsymbol{\beta}$ with a heteroscedastic prior. Specifically, we replace the sequence of identical variances for the random walk increments in $\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}$ with a piecewise constant variance list consisting of B different values, i.e. $\tau_b^2,\ b = 1, \ldots, B$. To characterize the piecewise constant variance step function, we can either consider the changepoints of the step function or the lengths of the constant pieces. Let $\boldsymbol{s} = (s_1, \ldots, s_{B-1})$ denote the vector of interior changepoints and set $s_0 = 1$ and $s_B = j - d$. From the changepoints we can derive the lengths of the intervals via $\boldsymbol{l} = (l_1, \ldots, l_B) = \boldsymbol{\Delta}^{(1)}(1, \boldsymbol{s}, j - d)'$ and vice versa $s_b = l_1 + \ldots + l_b$. The variance for the Gaussian random walk on $\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}$ at indices $i \in \{s_{b-1}, \ldots, s_b - 1\}$ is then given by τ_b^2. By designating a random walk prior with piecewise constant variances, we reduce the numbers of parameters to be sampled. Furthermore, this allows us to take into account local information about the variability of the function to be fitted and thereby increases robustness of the fitted function to outliers compared to using individual variances $\tau_i^2, i = 1, \dots, j$ for the random walk increments.

The following directed acyclic graph (DAG) gives the basic hierarchy for the proposed model specification:

$$\begin{array}{l} z_b \sim \Gamma(a_z, b_z); \quad b = 1, \dots, B \\ \quad \downarrow \\ \tau_b^2 \sim \mathrm{Exp}(z_b); \quad b = 1, \dots, B \\ \quad \downarrow \\ \boldsymbol{\Delta}^{(d)}\boldsymbol{\beta} \sim \mathcal{N}_{j-d}(\mathbf{0}, \boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l})); \quad \boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l}) = \mathrm{blockdiag}\left(\tau_1^2 \boldsymbol{I}_{l_1}, \dots, \tau_B^2 \boldsymbol{I}_{l_B}\right) \\ \quad \downarrow \\ \boldsymbol{y} \sim \mathcal{N}_n(\boldsymbol{X}\boldsymbol{\beta}, \sigma_\varepsilon^2) \end{array} \tag{1}$$

and the corresponding posterior $p(\boldsymbol{y}, \sigma_\varepsilon^2, \boldsymbol{\beta}, \boldsymbol{\tau}^2, \boldsymbol{z}, b_z | \boldsymbol{x})$ is given by

$$p(\boldsymbol{y}, \sigma_\varepsilon^2, \boldsymbol{\beta}, \boldsymbol{\tau}^2, \boldsymbol{z}, b_z | x) = \\ \pi_y(\boldsymbol{y} | \boldsymbol{X}\boldsymbol{\beta}, \sigma_\varepsilon^2) \pi_\beta(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta} | \boldsymbol{\tau}^2) \pi_{\tau^2}(\boldsymbol{\tau}^2 | \boldsymbol{z}) \pi_z(\boldsymbol{z} | a_z, b_z) \pi_{b_z} b_z.$$

Despite the appealing theoretical properties of the MAP estimates we use a full MCMC approach instead of, say, an EM-type algorithm, because of the importance of reliable variability measures for function estimation and because an implementation based on a full MCMC approach will facilitate inclusion into the general structured additive regression context f.e. as part of a Bayesian backfitting algorithm for (G)AMs. The rate b_z is sampled with a Metropolis-Hastings-Step. The remaining parameters $\boldsymbol{z}, \boldsymbol{\tau}^2, \boldsymbol{\beta}$ and σ_ε^2 are updated from their full conditionals (see App. A) via Gibbs-Sampling. We use a weakly informative inverse gamma prior, $\mathrm{IG}(a_\sigma = 10^{-5}; b_\sigma = 10^{-5})$, for the variance σ_ε^2 of the errors.

In the following we describe and compare three approaches with increasing flexibility for the variance function given by $\boldsymbol{\tau}^2$: The first approach uses a piecewise constant variance function with fixed number and positioning of changepoints as described in this section. In the second approach, we sample the locations of the changepoints while leaving their number fixed. In the third approach we use reversible jump MCMC methodology to sample the number of changepoints B as well.

2.1. Blockwise NEG P-spline

In this formulation, the number of blocks B as well as the positions and lengths of the blocks in the variance function are fixed. Simulation results were obtained using blocks of (approximately) equal length. The resulting posterior variance function is piecewise constant. We investigate the robustness with respect to the number of blocks in section 3.6.2. The hierarchy for this model is given in (1). In the following, this algorithm will be referred to as NEG.

2.2. Flexible Blockwise NEG P-spline

In this model, we let B remain fixed and sample the locations of the steps $s_1, \dots, s_{B-1}$ at which the variance of $\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}$ changes. The prior for the vector of interior changepoints $\boldsymbol{s} = (s_1, \dots, s_{B-1})$ is assumed to be the distribution of the order statistic of a discrete uniform distribution on $\{1, \dots, j-d-1\}$. The rest of the hierarchy and the sampler remains unchanged.

$$\begin{tikzcd}
z_b \sim \Gamma(a_z, b_z); \quad b = 1, \dots, B \arrow[dd] & & \boldsymbol{s} \sim \dfrac{(B-1)!}{(j-d-2)^{B-1}}; \\
 & & s_i \in \{2, \dots, j-d-1\}; \\
\tau_b^2 \sim \mathrm{Exp}(z_b); \quad b = 1, \dots, B \arrow[dr] & & (l_1, \dots, l_B) = \boldsymbol{\Delta}^{(1)}(1, \boldsymbol{s}, j-d)' \arrow[dl] \\
 & \boldsymbol{\Delta}^{(d)}\boldsymbol{\beta} \sim \mathcal{N}_{j-d}(\mathbf{0}, \boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l})); \quad \boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l}) = \mathrm{blockdiag}\left(\tau_1^2 \boldsymbol{I}_{l_1}, \dots, \tau_B^2 \boldsymbol{I}_{l_B}\right) \arrow[d] & \\
 & \boldsymbol{y} \sim \mathcal{N}_n(\boldsymbol{X\beta}, \sigma_\varepsilon^2) &
\end{tikzcd}$$

We use the following Metropolis-Hastings step to update the vector of changepoints $\boldsymbol{s}$:

Updating $\boldsymbol{s}$:

- Define the tuning parameter m_s, which is the maximal number of indices that the new proposal can move the selected changepoint. In our implementation, m_s defaults to $\lceil (j-d)/B \rceil$, the length of the random walk divided by the number of blocks and rounded to the next highest integer.
- Draw $b^\star$ uniformly from the set of indices of movable changepoints

  $$\mathcal{B}_s = \{1, \dots, B-1\} \setminus \{b : l_b = 1 \text{ and } l_{b+1} = 1\}.$$

  Indices b where $l_b = 1$ and $l_{b+1} = 1$ are not permitted, because both neighboring intervals only span a single index so that the changepoint in the middle cannot move. Let $B_m = |\mathcal{B}_s|$ denote the number of movable indices.

- Determine the minimal index $i_- = \max(s_{b^\star-1}+1, s_{b^\star} - m_s)$ and maximal index $i_+ = \min(s_{b^\star+1}-1, s_{b^\star}+m_s)$ and draw the proposal $s^\star_{b^\star}$ to replace $s_{b^\star}$ uniformly from $\{i_-, \ldots, i_+\}$.
- Update $\boldsymbol{s}^\star$, $\boldsymbol{l}^\star$, $i^\star_-$, $i^\star_+$ and $\boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l}^\star)$ accordingly.
- Accept the new vector of change points $\boldsymbol{s}^\star$ with probability $\alpha(\boldsymbol{s}^\star)$:

$$\begin{aligned}\log\alpha(\boldsymbol{s}^\star) &= \log\left((i_+ - i_-)\,B_m\right) - \log\left((i^\star_+ - i^\star_-)\,B^\star_m\right)\\ &\quad +0.5\left(\frac{\operatorname{diag}(\boldsymbol{T}(\boldsymbol{\tau}^2,\boldsymbol{l}^\star)) - \operatorname{diag}(\boldsymbol{T}(\boldsymbol{\tau}^2,\boldsymbol{l}))}{\operatorname{diag}(\boldsymbol{T}(\boldsymbol{\tau}^2,\boldsymbol{l}^\star))\cdot\operatorname{diag}(\boldsymbol{T}(\boldsymbol{\tau}^2,\boldsymbol{l}))}\right)'(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})^2\\ &\quad +0.5(\boldsymbol{l}-\boldsymbol{l}^\star)'\log(\boldsymbol{\tau}^2),\end{aligned}$$

  where the expression in the first line is the proposal ratio for $\boldsymbol{s}^\star$, and the second and third line come from the prior ratio for the random walk.

This model gives substantially more flexibility with regard to the estimated variance function $\boldsymbol{\tau}$ by averaging over the step functions drawn in each iteration. In effect, we use Bayesian model averaging to arrive at posterior estimates for $f(x)$ and $\tau^2_i,\ i = 1, \ldots, j-d$. In the following, this algorithm will be referred to as FlexNEG.

We also experimented with another, more elaborate proposal scheme for $\boldsymbol{s}^\star$: In this scheme we drew the proposed new changepoint $s^\star$ uniformly from $\{1, \ldots, j-d\} \setminus \{\boldsymbol{s}\}$ and let the probability with which to propose replacing the changepoint to the left (s_-) or to the right (s_+) of $s^\star$ with $s^\star$ depend on the distances from $s^\star$ to the neighboring changepoints s_- and s_+ and on the change in the logarithm of the variance function at the changepoints neighboring $s^\star$. Specifically, we set the probability to propose replacing the neighboring changepoint s_- on the left with $s^\star$ to

$$\frac{(s_+ - s^\star)|\log(\tau^2_+) - \log(\tau^2_\star)|}{(s^\star - s_-)|\log(\tau^2_-) - \log(\tau^2_\star)| + (s_+ - s^\star)|\log(\tau^2_+) - \log(\tau^2_\star)|},$$

where $\tau^2_\star$ denotes the current value of the variance function at $s^\star$. This probability is large if $s^\star$ is further removed from s_+ than from s_- and if the value of the variance function at $s^\star$ is more different from the variance in the neighboring block to the right (τ^2_+) than from the variance in the neighboring block to the left (τ^2_-). However, this proposal usually performed less well than the simpler one described above.

2.3. Flexible Blockwise NEG P-spline with variable number of blocks

As the most flexible alternative, we also implemented a reversible jump-type algorithm [Green, 1995] to determine the number of changepoints B automatically in a data-driven way. We used a truncated Poisson distribution on $\{1,\dots,s_{\max}\}$ with rate s_{mean} as prior $\pi_B(B)$ for the number of blocks B. The rest of the hierarchy remains unchanged:

$$\begin{tikzcd}
& B \sim Po_{\text{trunc}}(\lambda = s_{\text{mean}}, \max = s_{\max}) \arrow[dl] \arrow[dr] & \\
z_b \sim \Gamma(a_z, b_z); \quad b = 1,\dots,B \arrow[dd] & & \boldsymbol{s} = (s_1,\dots,s_{B-1}) \sim \frac{(B-1)!}{(j-d-2)^{B-1}}; \\
& & s_i \in \{2,\dots,j-d-1\}; \\
\tau_b^2 \sim \text{Exp}(z_b); \quad b = 1,\dots,B \arrow[dr] & & (l_1,\dots,l_B) = \boldsymbol{\Delta}^{(1)}(1,\boldsymbol{s},j-d)' \arrow[dl] \\
& \boldsymbol{\Delta}^{(d)}\boldsymbol{\beta} \sim \mathcal{N}_{j-d}(\boldsymbol{0},\boldsymbol{T}(\boldsymbol{\tau}^2,\boldsymbol{l})); \quad \boldsymbol{T}(\boldsymbol{\tau}^2,\boldsymbol{l}) = \text{blockdiag}\left(\tau_1^2\boldsymbol{I}_{l_1},\dots,\tau_B^2\boldsymbol{I}_{l_B}\right) \arrow[d] & \\
& \boldsymbol{y} \sim \mathcal{N}_n(\boldsymbol{X\beta},\sigma_\varepsilon^2) &
\end{tikzcd}$$

The reversible jump algorithm has three move types: birth (adding a changepoint), death (removing a changepoint), and position change. The latter is identical to the update procedure for $\boldsymbol{s}$ described in the previous section. Let $p_b(B)$ and $p_d(B)$ denote the probability for a birth and death step, respectively, given the number of blocks B. To satisfy detailed balance, we set $p_b(B) = c\min\left(1,\pi_B(B+1)/\pi_B(B)\right)$; $p_d(B) = c\min\left(1,\pi_B(B-1)/\pi_B(B)\right)$, where c is chosen so that $p_b(B) + p_d(B) < 0.8 \;\forall\; B$ [Green, 1995]. The birth and death moves to increase or decrease B are as follows:

Birth Move: $B \longrightarrow (B+1)$

- Draw the proposed new changepoint $s^\star$ uniformly from $\{2,\dots,j-d-1\} \setminus \{\boldsymbol{s}\}$.
- Determine the affected block $b^\star : s_{b^\star-1} < s^\star < s_{b^\star}$ and the (expanded) proposal vectors $\boldsymbol{s}^\star$ and $\boldsymbol{l}^\star$.
- Draw proposals $z^\star_{b^\star}, z^\star_{b^\star+1} \sim \Gamma(a_z+1, b_z+\tau^2_{b^\star})$ from the full conditional based on $\boldsymbol{\tau}^2$ from the *previous* iteration.

- Draw proposals $\tau^{2\star}_{b^\star}, \tau^{2\star}_{b^\star+1}$ from their full conditionals (see app. A) based on the *updated* vector $\boldsymbol{z}^\star$.
- Accept $B^\star = B + 1$, $\boldsymbol{s}^\star$, $\boldsymbol{z}^\star$ and $\boldsymbol{\tau}^{2\star}$ with probability $\alpha_b = \mathcal{A}_b \mathcal{P}_b$, where $\mathcal{A}_b$ is the prior ratio and $\mathcal{P}_b$ is the proposal ratio for the birth move.

The acceptance probability has this simple form because the likelihood ratio for dimension changing moves is 1. In our context, the changed parameters do not occur in the likelihood but only in a higher stage of the hierarchy. The Jacobian is 1 as well since the mapping function between the parameter spaces is the identity. The prior ratio is given by

$$\begin{aligned}\mathcal{A}_b =& \frac{\pi_B(B^\star|s_{\text{mean}}, s_{\text{max}})}{\pi_B(B|s_{\text{mean}}, s_{\text{max}})} \frac{\pi_s(\boldsymbol{s}^\star|B^\star)}{\pi_s(\boldsymbol{s}|B)} \frac{\pi_z\left((z^\star_{b^\star}, z^\star_{b^\star+1})|b_z\right)}{\pi_z(z_{b^\star}|b_z)} \\ & \frac{\pi_\tau\left((\tau^{2\star}_{b^\star}, \tau^{2\star}_{b^\star+1})|\boldsymbol{z}^\star\right)}{\pi_\tau(\tau^2_{b^\star}|\boldsymbol{z})} \frac{\pi_{\Delta\beta}\left(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}|\boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l}^\star)\right)}{\pi_{\Delta\beta}\left(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}|\boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l})\right)},\end{aligned}$$

and the proposal ratio for the birth step is

$$\begin{aligned}\mathcal{P}_b =& \frac{p_d(B^\star)}{p_b(B)} \frac{|\{2, \ldots, j-d-1\} \setminus \{\boldsymbol{s}\}|}{B} \frac{\pi\left(z_{b^\star}|a_z, b_z, \tau^2_{b^\star}\right)}{\pi\left((z^\star_{b^\star}, z^\star_{b^\star+1})|a_z, b_z, \tau^2_{b^\star}\right)} \\ & \frac{\pi\left(\tau^2_{b^\star}|\boldsymbol{s}, \boldsymbol{l}, \tilde{z}_{b^\star} = 0.5(z^\star_{b^\star} + z^\star_{b^\star+1}), \boldsymbol{\beta}\right)}{\pi\left((\tau^{2\star}_{b^\star}, \tau^{2\star}_{b^\star+1})|\boldsymbol{s}^\star, \boldsymbol{l}^\star, \boldsymbol{z}^\star, \boldsymbol{\beta}\right)},\end{aligned}$$

see App. B for detailed expressions. The required dimension matching [Green, 1995] is fulfilled since the dimension changes (in the notation for the birth step) proceed from parameter vector $\left(\boldsymbol{z}, \boldsymbol{\tau}^2, z^\star_{b^\star}, z^\star_{b^\star+1}, \tau^{2\star}_{b^\star}, \tau^{2\star}_{b^\star+1}\right)$ with dimension $2B + 4$ to parameter vector $\left(\boldsymbol{z}^\star, \boldsymbol{\tau}^{2\star}, z_{b^\star}, \tau^2_{b^\star}\right)$ with dimension $2(B + 1) + 2$ and vice versa for the death step. We also experimented with more complex proposal schemes for the birth and death moves and with replacing the truncated Poisson prior for B with a discrete uniform prior. Neither of these changes affected the performance of the algorithm decisively. We document the alternative proposal schemes in appendix B. Acceptance probabilities for the dimension changing moves were in the range of 0.3 to 0.6 and usually around 0.4.
We also experimented with alternating between the dimension-changing transition kernel implied by the update procedure above and the fixed-dimension kernel described in section 2.2. This did not influence results in a systematic way and usually increased the necessary burn-in period.

Death Move: $(B+1) \rightarrow B$

- Draw index $b^\star$ of the changepoint that is to be deleted uniformly from $\{1, \dots, B-1\}$ and determine the reduced proposal vectors $\boldsymbol{s}^\star$, $\boldsymbol{l}^\star$.
- Draw the new proposal $\tau^{2\star}_{b^\star}$ to replace $(\tau^2_{b^\star}, \tau^2_{b^\star+1})$ from $\pi(\tau^{2\star}_{b^\star}|\boldsymbol{s}^\star, \tilde{z}_{b^\star} = 0.5(z_{b^\star} + z_{b^\star+1}))$
- Draw the new proposal $z^\star_{b^\star}$ to replace $(z_{b^\star}, z_{b^\star+1})$ from $\pi(z^\star_{b^\star}|a_z, b_z, \boldsymbol{\tau}^{2\star})$
- Accept $B^\star = B-1$, $\boldsymbol{s}^\star$, $\boldsymbol{z}^\star$ and $\boldsymbol{\tau}^{2\star}$ with probability $\alpha_d = \log(\mathcal{A}_d\mathcal{P}_d)$: where $\mathcal{A}_d$, the prior ratio, and $\mathcal{P}_d$, the proposal ratio, are simply $\mathcal{A}_b^{-1}$ and $\mathcal{P}_b^{-1}$ with indices appropriately changed.

In the following, this algorithm will be referred to as RJNEG.

3. Simulation Results

This section presents the results of a simulation study we did to evaluate the performance of our method and to compare it with other approaches for spatially adaptive spline estimation. We compared the performance of our methods to the performance of the spatially adaptive Bayesian P-Splines suggested by Baladandayuthapani et al. [2005] and the R [R Development Core Team, 2007] implementation AdaptFit [Krivobokova, 2007] of the frequentist equivalent of their model described in Krivobokova et al. [2007]. In the following, these approaches will be referred to as BMC and AdaptFit, respectively. For both algorithms we used the published hyperparameter settings, number of knots etc. Both approaches are based on a representation of the logarithm of the variance function $\log(\tau^2(x))$ as a second P-spline. We use these two methods for benchmarking since their performances are reportedly superior – or at least equivalent – to those of the competing wavelet approach of Donoho and Johnstone [1994], to the knot-selection based approach by Denison et al. [1998] and the approach based on a heteroscedastic heavy-tailed random walk priors for $\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}$ by Lang and Brezger [2004]. We also compared the performance of our approach to the performance of the latter. Average MSEs were consistently larger by an order of magnitude for the latter and we omit a detailed analysis for these results in the following.

We consider 4 widely used benchmarking functions that, together, represent a cross section of challenging functional forms encountered in real-world data. We generated 100 datasets for every function and obtained the fits of the considered methods. Pointwise coverage values (calculated for a nominal level of 90%) should therefore be treated with caution, since the sample sizes are not really large enough for reliable estimation. Graphical panels include boxplots of $\log_{10}(\sqrt{MSE})$, benchmarkplots (see App. C) based on MSE, as well as the achieved coverage over the 100 replications for a nominal level of 90% for pointwise intervals and a plot of the point-wise bias.

It should be noted that published results for AdaptFit are based on the S-PLUS version of the algorithm while our results are based on the R implementation, which seems to suffer from a less stable optimizer in the underlying mixed effects model software (`nlme::lme`, Pinheiro et al. [2006]). In order to avoid large proportions of non-convergence errors, we relaxed the convergence criterion for the estimated variance function, which may explain the discrepancies between the results in Krivobokova et al. [2007] and ours. Discrepancies between our results for BMC in section 3.1 and published results are due to a faulty simulation design in Baladandayuthapani et al. [2005][1].

3.1. Oscillating function

As an example for a function with smoothly varying curvature, we used the doppler-like function

$$m_1(x) = \sqrt{x(1-x)}\sin\frac{18\pi}{x+8}$$

with $n = 400$ observations and $\sigma_\varepsilon^2 = 0.04$ (SNR ≈ 2.1) in accordance with the set-up in Baladandayuthapani et al. [2005] and Krivobokova et al. [2007]. Results are based on cubic P-splines ($j = 90; d = 2$) with $B = 15$ for NEG, $B = 10$ for FlexNEG and $s_{\text{mean}} = 5, s_{\text{max}} = 40$ for RJNEG. Although differences in MSE between the two top competitors FlexNEG (average MSE (AMSE): 0.0034) and AdaptFit (AMSE: 0.0035) are negligible, FlexNEG has a better coverage and considerably smaller bias in the difficult region of the third to sixth oscillations from the left. Average coverage for FlexNEG is slightly conservative (.93), the average coverage of the other methods is between .895 and .905. We were unable to reproduce the results in Baladandayuthapani et al. [2005] which report an AMSE of 0.00028. This is due to a mistake in their simulation design, our results give an AMSE of 0.0044. The mean posterior median of B over the 100 simulations for RJNEG is 5. It should be noted that, in the case of FlexNEG, convergence of b_z for this function can be fairly slow if the chain is started with smallish (< 10) values of b_z. For most datasets, there seem to be multiple

[1] Because we could not find an explanation for the discrepancies between between our results for BMC in section 3.1 and published results we requested the original simulation files used by V. Baladandayuthapani, who was kind enough to oblige us. We found out that published results are based on repeated fitting of identical data, because the random generator for the errors was set to the same seed in each iteration of the data generating simulation. The dataset they fit 100 times had a fairly small error variance (.0355 compared to the nominal .04) and consequently produced fits with a fairly good MSE. The (small) variability in their results was entirely due to the MC error of the MCMC chains.

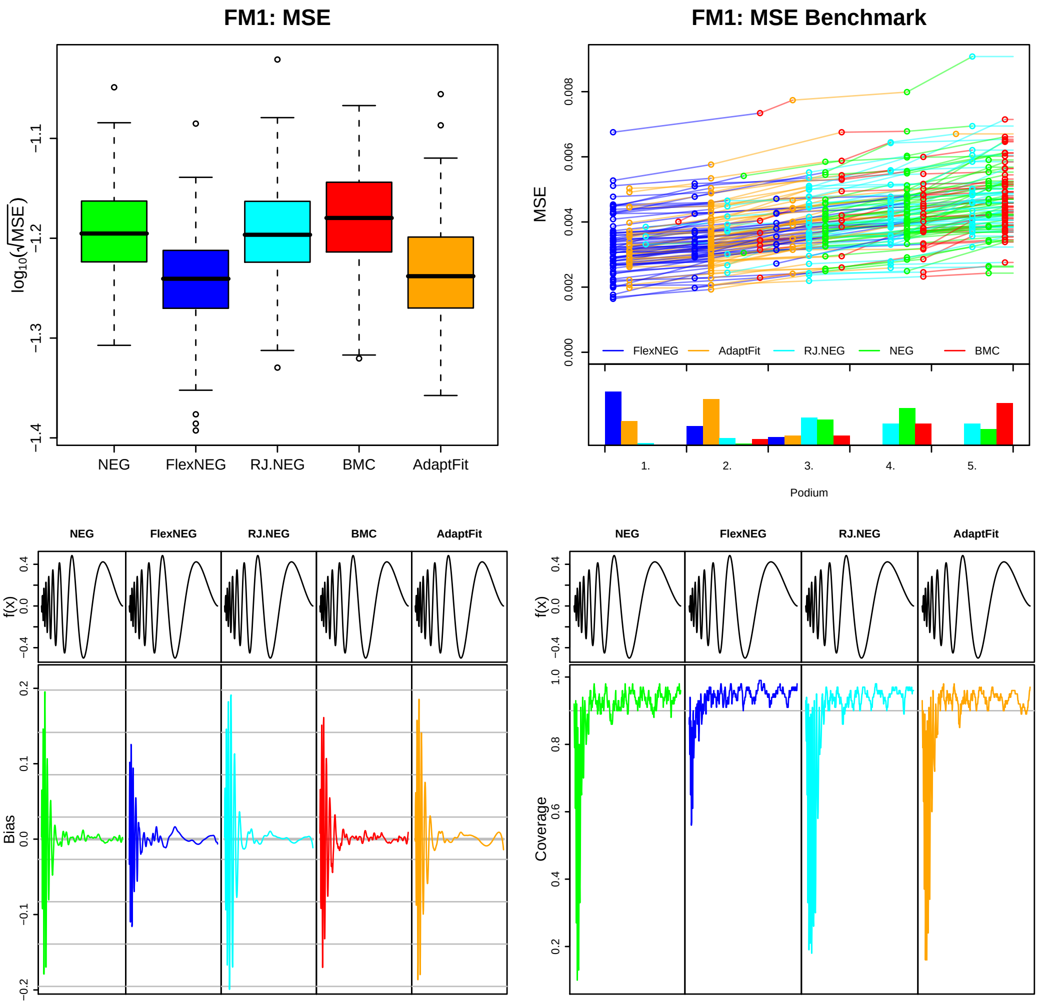

Figure 1: Simulation results for $m_1(\cdot)$ (100 data sets). Upper left panel shows boxplots of $\log_{10}(\sqrt{MSE})$ for the methods under consideration, upper right panel a benchmark plot (see App. C) for the MSE. Lower row shows the pointwise bias and the observed pointwise coverage based on a nominal level of .90.

modes corresponding to different degrees of b_z and the chain has to be long enough ($>$ 30000 iterations) to include visits to all of them. Differences between the function estimates from the basins of attraction of the various modes are negligible, however.

3.2. Constant to Oscillating Function

An even greater smooth variation in curvature properties is evident in the function

$$m_2(x) = \exp(-400(x-0.6)^2) + \frac{5}{3}\exp(-500(x-0.75)^2) + 2\exp(-500*(x-0.9)^2)$$

with $n = 1000$ observations and $\sigma_\varepsilon^2 = 0.25$ (SNR $\approx$ 1.2) in accordance with the set-up in Baladandayuthapani et al. [2005] and Krivobokova et al. [2007]. Results are based on cubic P-splines ($j = 40; d = 2$) with $B = 4$ for NEG, $B = 16$ for FlexNEG and $s_{\text{mean}} = 2, s_{\text{max}} = 10$ for RJNEG. The NEG-based methods show slightly stronger regularization and, therefore, smaller average bias for the region $x < 0.4$ where the function is constant. We assume this is due to the larger shrinkage of the strongly peaked NEG-prior. Note that the seemingly large average bias values of FlexNEG in the oscillating part of the function are negligible compared to the value of the true function - the function value at the rightmost peak is underestimated by about 3% at most. AMSEs for RJNEG and AdaptFit are about the same (0.0049) and slightly larger than those for FlexNEG (0.0045) and NEG (0.0047). Average coverage for FlexNEG (.94), RJNEG(.93) and NEG (.91) is conservative. Mean posterior median B for RJNEG is 3.

3.3. Blocks: Step Function

As an example for a very un-smooth function with many discontinuities, we considered the blocks function as specified in Donoho and Johnstone [1994] with $n = 2048$ observations and $\sigma_\varepsilon^2 = 1$ (SNR $\approx$ 3.7). Results are based on cubic P-splines ($j = 300; d = 1$) with $B = 60$ for NEG, $B = 45$ for FlexNEG and $s_{\text{mean}} = 50, s_{\text{max}} = 100$ for RJNEG. As might be expected, both AdaptFit and BMC, which attempt to model a smooth variance function do not perform as well in this situation as the NEG models which use a more flexible piecewise constant representation of the variance function. This can also be seen from the bias plot: It is apparent that bias is similarly large at the edges of the respective plateaus for all methods. This is due to the assumption of a continuous $f(x)$ common to all the models we consider, which is inappropriate in this case. However, the NEG-based fits have much smaller bias for the plateau regions, because their underlying variance functions return more quickly to much

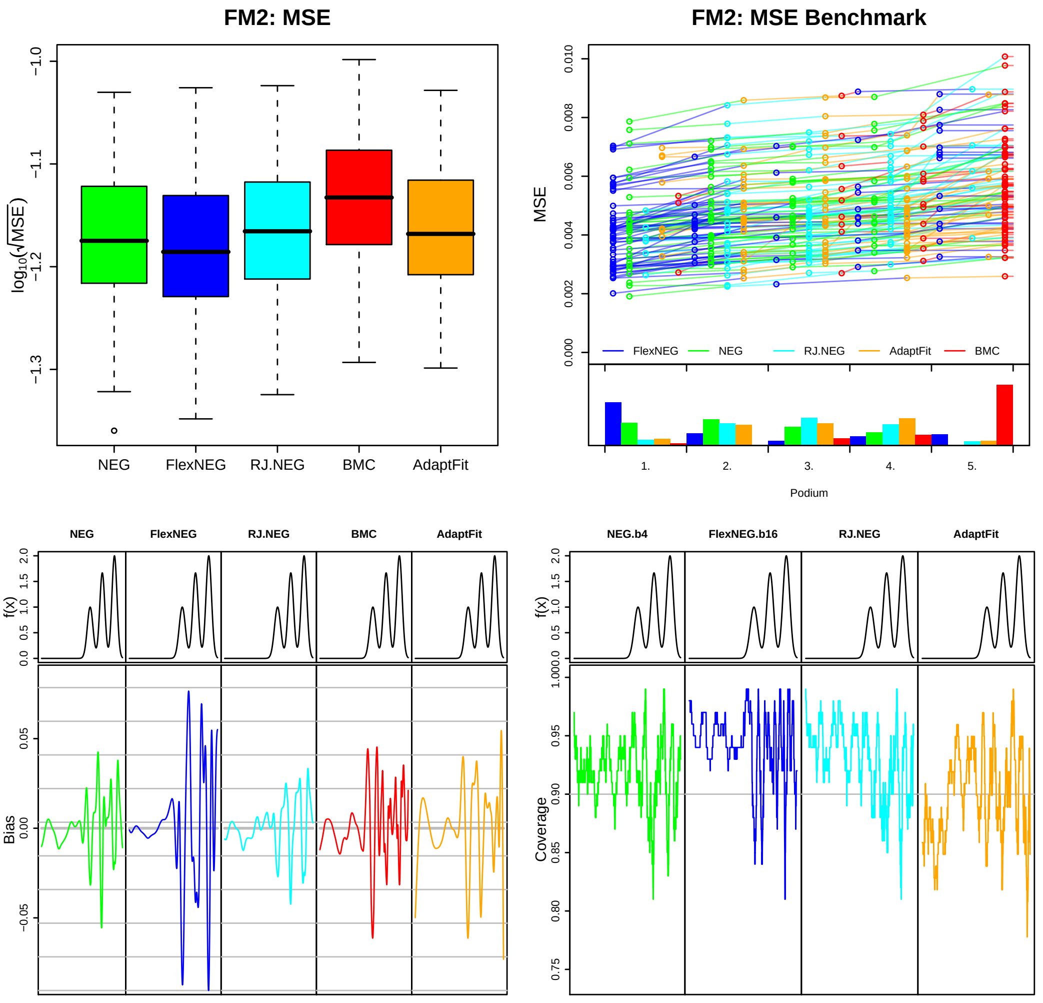

Figure 2: Simulation results for $m_2(\cdot)$ (100 data sets). Upper left panel shows boxplots of $\log_{10}(\sqrt{MSE})$ for the methods under consideration, upper right panel a benchmark plot (see App. C) for the MSE. Lower row shows the pointwise bias and the observed pointwise coverage based on a nominal level of .90.

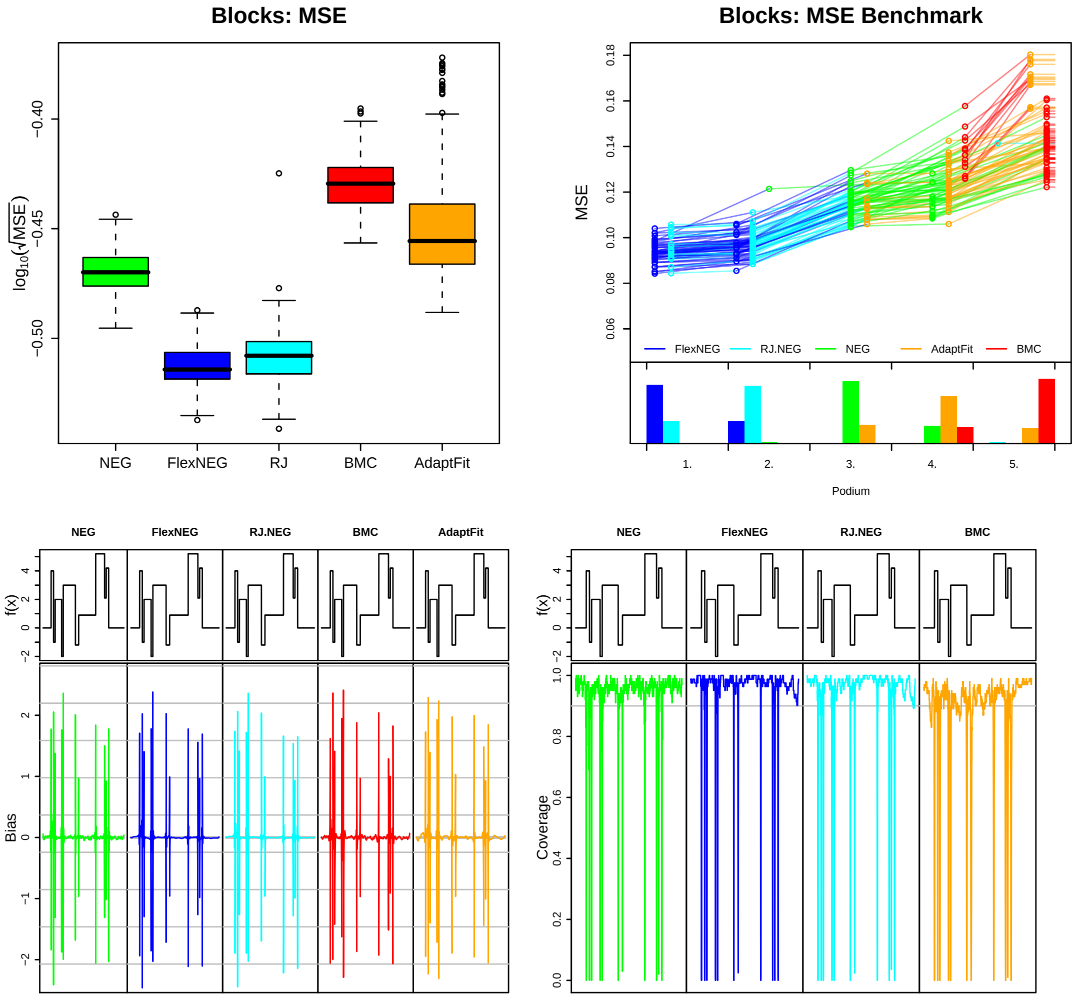

Figure 3: Simulation results for blocks function (100 data sets). Upper left panel shows boxplots of $\log_{10}(\sqrt{MSE})$ for the methods under consideration, upper right panel a benchmark plot (see App. C) for the MSE. Lower row shows the pointwise bias and the observed pointwise coverage based on a nominal level of .90.

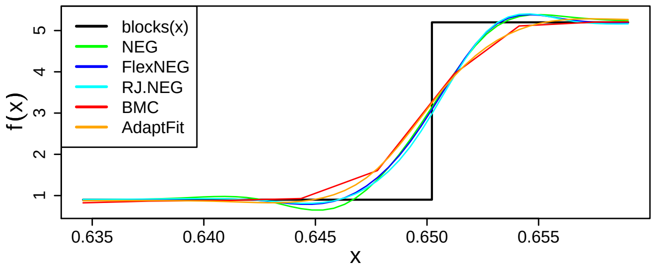

Figure 4: Mean estimates for the blocks function for the discontinuity at $x = 0.65$

smaller values implying strong regularization and less wiggliness of the fitted function. This can also be seen from the coverage plot: At the discontinuities, FlexNEG's coverage returns above the nominal level more quickly. Fig. 4 shows the mean over the 100 estimated functions at the discontinuity at $x = 0.65$ for the various methods. It is easy to see that the NEG-based methods show less oscillation and are better able to reproduce the jump. Average coverage for AdaptFit is anti-conservative (.87) and conservative ($.93 - .94$) for the NEG-based methods. Mean posterior median B for RJNEG is 43.

3.4. Heavisine: Smooth Function with Discontinuities

A second function with discontinuities but non-constant function values between the jumps is given by the heavisine function as specified in Donoho and Johnstone [1994] with $n = 2048$ observations and $\sigma_\varepsilon^2 = 1$ (SNR ≈ 8.8). Results are based on cubic P-splines ($j = 100; d = 2$) with $B = 10$ for NEG, $B = 30$ for FlexNEG and $s_{\text{mean}} = 60, s_{\text{max}} = 95$ for RJNEG. As for the blocks function, the NEG models are better able to deal with the discontinuities in this function because of the heavy tails of NEG prior and the ability of the piecewise constant variance function to model short spikes in variability. While the maximal bias values at the discontinuities themselves are practically identical for all methods, FlexNEG and RJNEG have much smaller bias and much better coverage in the proximity of the discontinuities. Mean posterior median B for RJNEG is 42. Fig. 6 shows the square root of estimated variance functions for an exemplary dataset. FlexNEG, RJNEG and, to a lesser extent due to the less flexible parametrization, NEG show pronounced spikes in variance around the two discontinuities of the function, while the variance estimated by AdaptFit is much too smooth and does not capture the true structure of the variability.

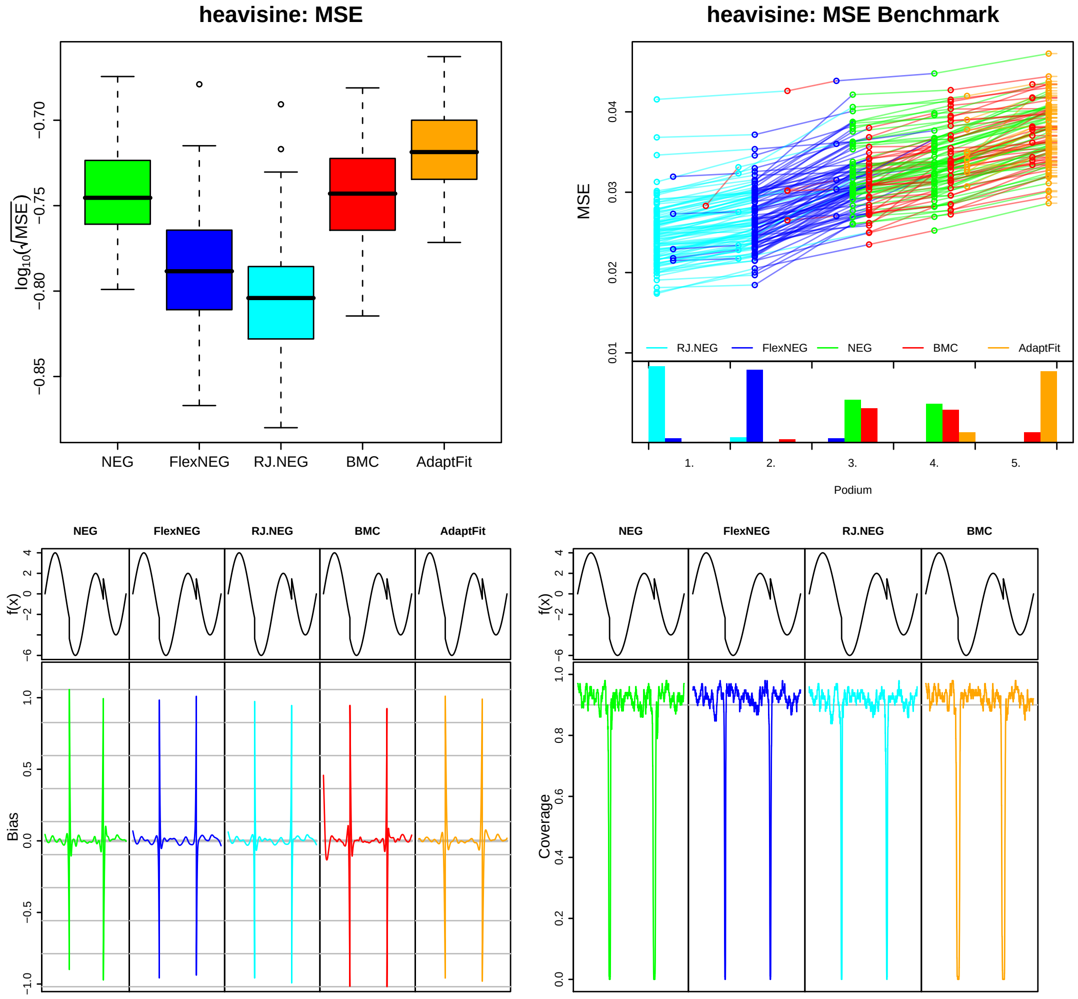

Figure 5: Simulation results for heavisine function (100 data sets). Upper left panel shows boxplots of $\log_{10}(\sqrt{MSE})$ for the methods under consideration, upper right panel a benchmark plot (see App. C) for the MSE. Lower row shows the pointwise bias and the observed pointwise coverage based on a nominal level of .90.

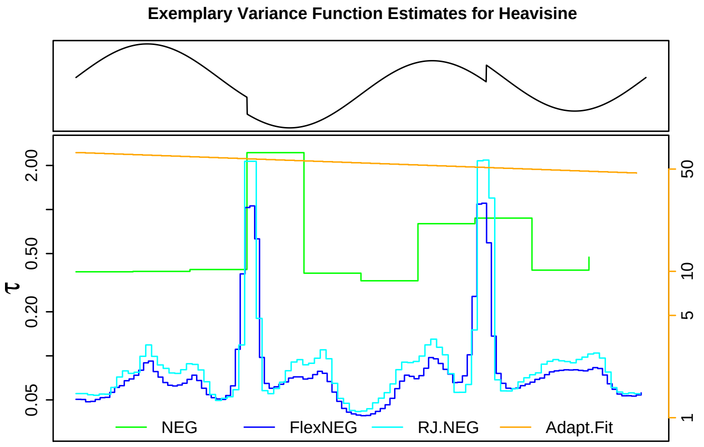

Figure 6: Square root of exemplary estimated variance functions for the heavisine function. Note the much larger scale for AdaptFit given on the right side of the plot.

3.5. Quantitative Analysis of Simulation Performances

Following the suggestions in Hothorn et al. [2005] we quantify the observed differences in $\log_{10}(\sqrt{MSE})$ via a linear mixed effects model. We include random effects for the simulated samples and an interaction term between function type and algorithm. Fig. 7 shows estimated Tukey contrasts for the algorithms with 95% confidence intervals corrected for multiple comparisons (single-step correction as implemented in R-package `multcomp` [Hothorn et al., 2008]). All three NEG-based algorithms are significantly better than both BMC and AdaptFit. Note that the estimated differences are quite relevant: an average difference in $\log_{10}(\sqrt{MSE})$ of -0.04 corresponds to a decrease in MSE by about 17%. FlexNEG outperforms both RJNEG and NEG, and RJNEG in turn outperforms NEG.

3.6. Robustness

3.6.1. Signal-to-Noise Ratio

We investigated the change in MSE for various signal-to-noise ratios (SNR) for both $m_1(x)$ (see section 3.1) and $m_2(x)$ (see section 3.2) for FlexNEG and compared it with the results of AdaptFit. Figure 8 shows that the change in MSE is about the same for both methods, with slight differences that do not yield a conclusive picture for small and medium SNR. FlexNEG seems to improve more strongly than AdaptFit for large SNR.

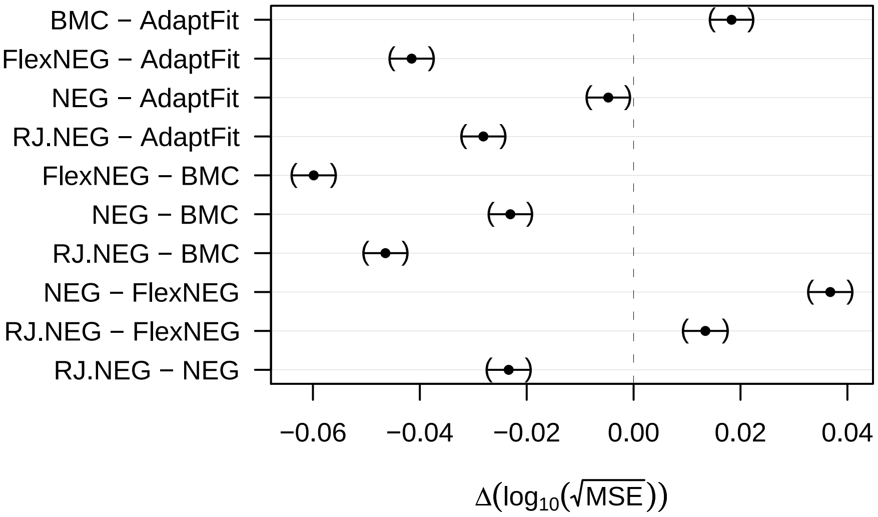

Figure 7: 95% family-wise confidence intervals and point estimates for differences in $\log_{10}(\sqrt{MSE})$ between algorithms

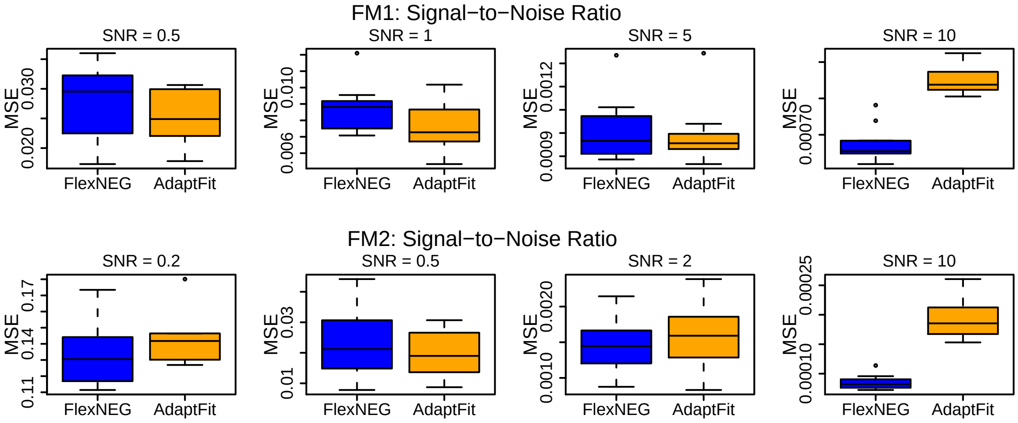

Figure 8: Boxplots of MSE for FlexNEG and AdaptFit for various signal-to-noise ratios. 10 datasets per SNR; settings correspond to sections 3.1 and 3.2, respectively.

3.6.2. Number of changepoints

We investigated the change in MSE for various specification of B or $(s_{\text{mean}}, s_{\text{max}})$ in a setting otherwise corresponding to the one given in section 3.1. Figure 9 shows that RJNEG's performance is very stable if the number of admitted changepoints is large enough, while FlexNEG and NEG lose a little performance for both too small and too large B. Note, however, that the performances of both NEG and FlexNEG are still very competitive even for sub-optimally chosen B and that the increase in MSE is relatively small. In order to see whether the best number of changepoints in real-world applications could be determined by the deviance information criterion (DIC) [Spiegelhalter et al., 2002] we computed DICs for the simulation runs. Model selection based on DIC would have worked very well for all three methods and simulated datasets, consistently selecting the MSE-optimal model or the next smallest candidate model if MSEs were very similar. The only exception is RJNEG for the Blocks function, where DIC would have selected the next larger model than the MSE-optimal one 4 out of 10 times.

4. Applications

4.1. Fractionation Curves

We apply our method to exemplary data from "Specificity Assessment From Fractionation Experiments" (SAFE) [Drobyshev et al., 2003] which are used for quality control of cDNA microarray experiments. Specifically, SAFE is used to investigate the degree of undesirable cross-hybridization of specific probe strands, e.g. how often cDNA sections pair with cDNA probes on the chip which have a similar, but not exactly equal, base sequence. For SAFE, microarrray chips are repeatedly treated with formamide solutions of increasing concentration and intensities are recorded after each washing. The series of resulting intensities for each probe on the chip is called a fractionation curve. As the cohesion between cross-hybridizing cDNA strands is weaker than between perfect matches, they are washed away at lower concentrations. If cross-hybridization occurs, there usually is a critical concentration in the lower range where a certain kind of cDNA sequence cross-hybridizing the probe sequence is abruptly washed away and a drop in signal intensity occurs.

Fits are based on P-splines of degree 0 with $j = 20$ basis functions and first order difference penalty for both the NEG-based methods and the non-adaptive fit with `mgcv::gam` [Wood, 2006] we used for comparison.

The left panel of Fig. 10 shows an example of a spot binding only the correct complementary cDNA. The location of the sharp decrease at about 65% indicates that the binding energy between complementary strands was no longer

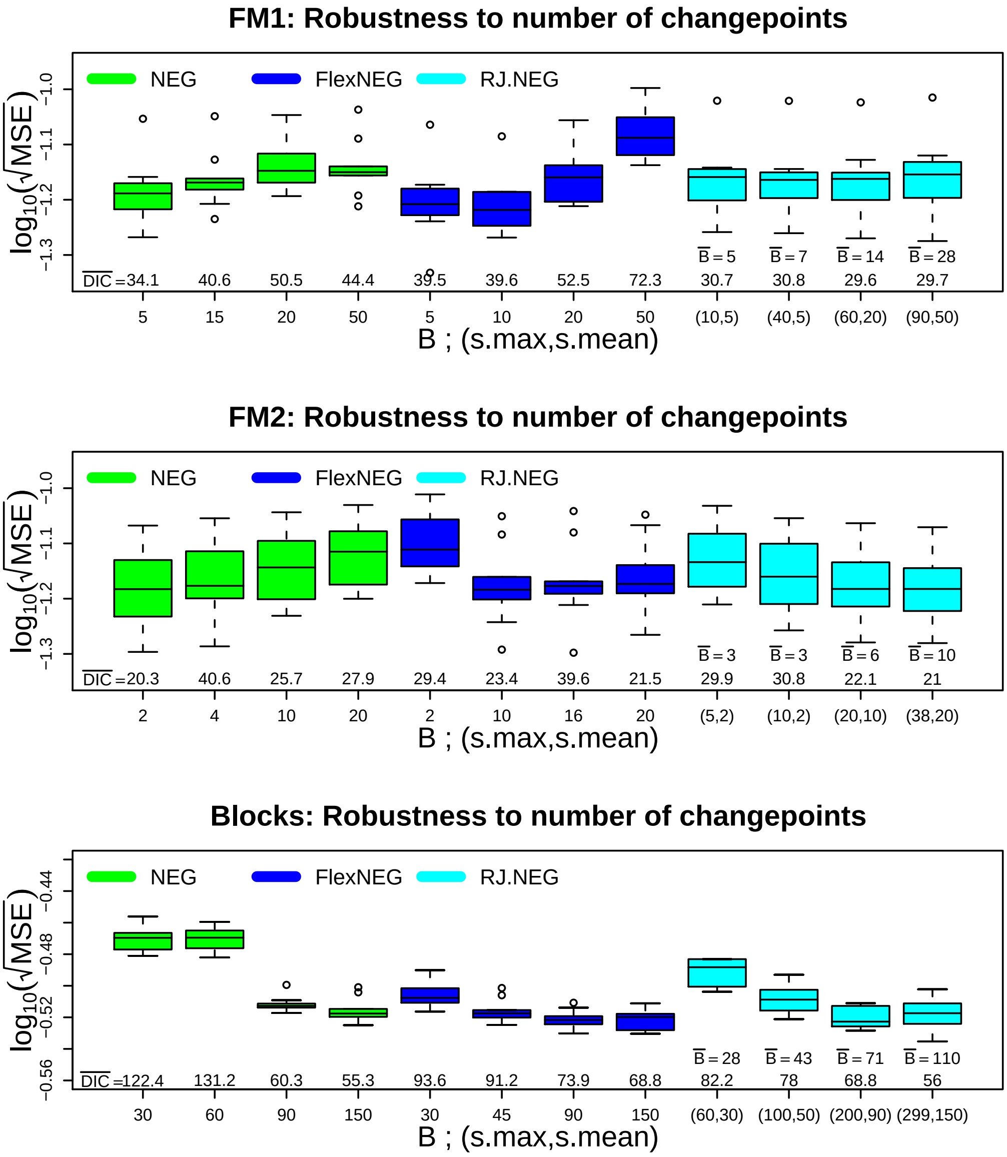

Figure 9: Boxplots of MSEs for various settings of B or $(s_{\max}, s_{\text{mean}})$. 10 datasets per value of B; other parameters as in settings of sections 3.1, 3.2 and 3.3. $\bar{B}$ is the (rounded) mean of posterior means of B for RJ.NEG. $\overline{\text{DIC}}$ is the (rounded) mean DIC over the 10 datasets.

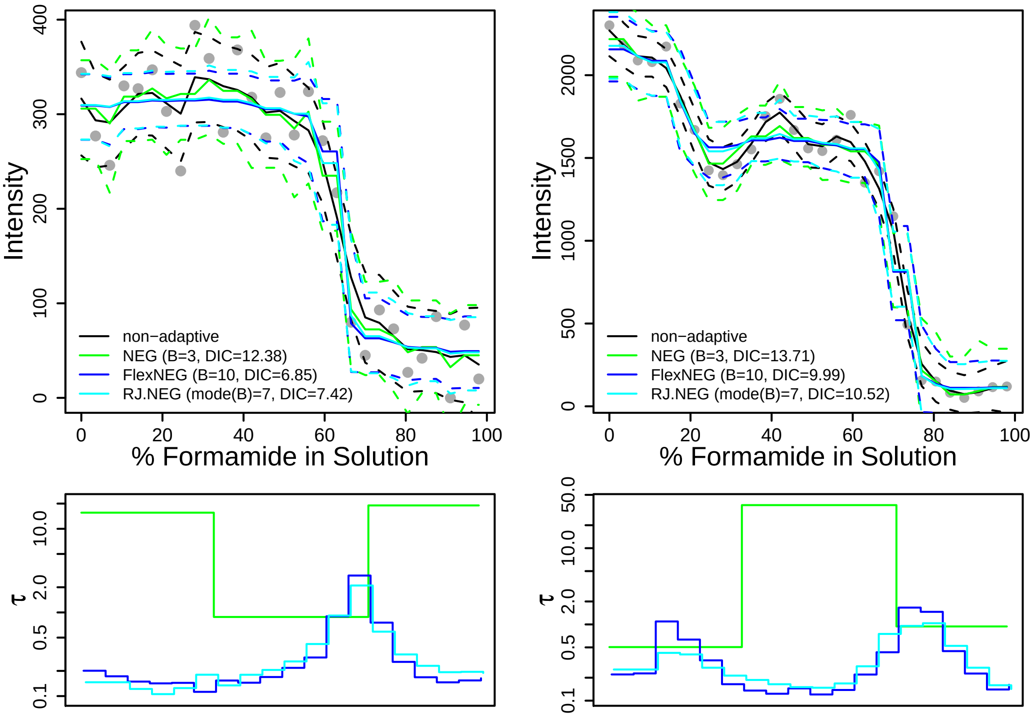

Figure 10: Two exemplary fractionation data sets and fitted functions with pointwise 95% confidence intervals. Lower panel shows the square root of the estimated variance functions for NEG, FlexNEG and RJNEG. Values on the abscissa of the lower panel are jittered to avoid overplotting.

sufficient for cohesion at this concentration . The right panel shows an example of a spot with cross-hybridization, where cross-hybridizing strands are washed away at a concentration of about 15%. We use the deviance information criterion (DIC) [Spiegelhalter et al., 2002] to choose B from 3, 5, and 10 and $(s_{\text{max}}, s_{\text{mean}})$ from $(19, 10)$, $(10, 5)$, and $(5, 3)$ for NEG, FlexNEG and RJNEG, respectively. Even without the explicit monotonicity constraints appropriate for this data, both FlexNEG and RJNEG fit the piecewise constant and decreasing structure that is expected very well. While the variance function of NEG does not seem interpretable due to the very low number of blocks in the DIC-optimal model, peaks in the variance function for both FlexNEG and RJNEG correspond exactly to the observable changepoints in the data. Note the excessive wiggliness of the non-adaptive fit for low concentrations in the left panel and for intermediate concentrations in the right panel which shows the improvement that can be gained by an adaptive fit in this context.

4.2. CP6 Data

We use the CP6 monthly sales data taken from West and Harrison [1989] and compare the fits and estimated variance functions for the methods under consideration. The data represent a time series of sales of tobacco and related products by a major UK company. Fits are based on cubic P-splines with $j = 20$ basis functions and first order difference penalty for the NEG-based methods and 10 knots (thin plate splines) for the non-adaptive fit with `mgcv::gam` [Wood, 2006] used for comparison.

The data seem to contain an additive outlier and a drastic change of slope in the last quarter of 1955 as well as further change points in the first months of 1957 and 1958. We use DIC to choose B from 3, 5, 10, and 19 and $(s_{\text{max}}, s_{\text{mean}})$ from $(19, 10)$ and $(10, 5)$ for NEG, FlexNEG and RJNEG, respectively. The differences between the fitted functions are fairly small (see Fig. 11), even between the adaptive methods and the conventional additive model fit (MGCV). All the methods seem to fit the data fairly well. The adaptive methods, except NEG, avoid the presumably spurious oscillations for 1959 and seem to identify a plateau for the first 3 quarters of 1957. Also note that FlexNEG seems to be more robust against the outlier in Dec. 1955 than the other methods. While the variance function of NEG does not seem interpretable, peaks in the variance function for both FlexNEG and, to a lesser degree, RJNEG correspond to the changepoints in the data well.

5. Conclusions

We showed how the NEG prior, combined with a flexible piece-wise constant representation of the local smoothing parameter, can be used for locally adap-

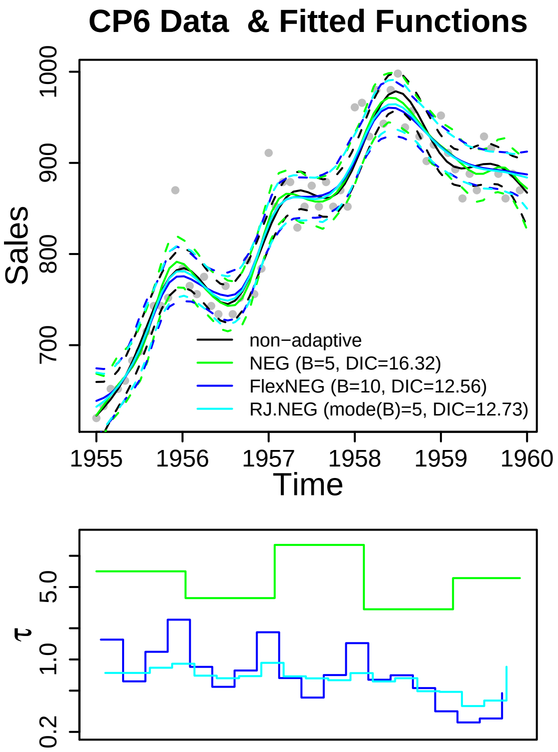

Figure 11: CP6 data and fitted functions with pointwise 95% confidence intervals. Lower panel shows the square root of the estimated variance functions for NEG, FlexNEG and RJNEG. Values on the abscissa of the lower panel are jittered to avoid overplotting.

tive smoothing in the linear model. We see the main strenghts of our approach in

1. its ability to deal with both discontinuous changes in the (complexity of the) fitted function and smoothly varying local complexity, while previous approaches we are aware of usually only succeed at the latter. Results for our approach are at least equivalent to the best previous approach (AdaptFit) for smoothly-varying variability and considerably better for functions with discontinuities. Both situations are also reflected in the applications considered in Section 4.
2. its very fast convergence and insensitivity to starting values due to the excellent mixing provided by the block-wise Gibbs samplers. Even for the very heavily parameterized Blocks function ($>$ 400 parameters) a burn-in period of about 5000 iterations is sufficient, while a burn-in period of at least 50000 iterations is recommended (personal comm. V. Baladandayuthapani) for the approach by Baladandayuthapani et al. [2005], for example.
3. its automatic applicability, since results for FlexNEG and RJNEG are fairly robust against the (only) user-specified hyperparameters which limit the maximal complexity of the implied variance function for the random walk increments.

Further work should embed our approach in a Bayesian backfitting algorithm and implement suitable update procedures for $\boldsymbol{\beta}$ to enable locally adaptive function estimation in the more general framework of structured additive regression models for non-Gaussian responses.

6. Acknowledgements

We would like to thank T. Krivobokova for patient assistance with the application of AdaptFit and the modification of its code and V. Baladandayuthapani for supplying his code and simulation scripts and his help in (unsuccessful) attempts at replicating previous results. We gratefully acknowledge Volkmar Liebscher's permission to let us use the fractionation curve data. This research has been funded by the German Science Foundation, grant FA 128/5-1.

A. Posterior and full conditionals

For the hierarchy given in section 2.3, the full posterior with given hyperparameters $s_{\text{mean}}, s_{\text{max}}, a_z, b_z, a_\sigma$ and b_σ can be written as

$$
\begin{aligned}
&p(B, \boldsymbol{s}, \boldsymbol{l}, \boldsymbol{z}, \boldsymbol{\tau}^2, \boldsymbol{\beta}, \sigma_\varepsilon^2, y) = \\
&\quad \left(1 - \sum_{i=1}^{s_{\max}} \frac{s_{\text{mean}}^i}{i!} e^{-s_{\text{mean}}}\right)^{-1} \frac{s_{\text{mean}}^B}{B!} e^{-s_{\text{mean}}} \cdot \frac{(B-1)!}{(j-d-2)^{B-1}} \cdot \\
&\quad \frac{b_z^{Ba_z}}{\Gamma(a_z)^B} \prod_{b=1}^{B} z_b^{a_z-1} \exp\left(-b_z z_b\right) \cdot \prod_{b=1}^{B} z_b \exp\left(-z_b \tau_b^2\right) \cdot \\
&\quad \frac{\prod_{b=1}^{B} \sqrt{\tau_b^2}^{-l_b}}{(2\pi)^{(j-d)/2}} \exp\left(-\frac{1}{2} \boldsymbol{\beta}' \boldsymbol{\Delta}^{(d)\prime} \boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l})^{-1} \boldsymbol{\Delta}^{(d)} \boldsymbol{\beta}\right) \cdot \\
&\quad \frac{b_\sigma^{a_\sigma}}{\Gamma(a_\sigma)} \sigma_\varepsilon^{2(-a_\sigma-1)} \exp\left(\frac{-b_\sigma}{\sigma_\varepsilon^2}\right) \\
&\quad \frac{1}{(2\pi\sigma_\varepsilon^2)^{n/2}} \exp\left(-\frac{\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2}{2\sigma_\varepsilon^2}\right).
\end{aligned}
$$

Accordingly, the full conditionals are:

$$
\begin{aligned}
&\pi(z_b | a_z, b_z, \tau_b^2) \propto z_b^{a_z} \exp\left(-(b_z + \tau_b^2) z_b\right) \\
&\Rightarrow z_b | \cdot \sim \Gamma(a_z + 1, b_z + \tau_b^2) \\
&\pi(\tau_b | \boldsymbol{s}, \boldsymbol{l}, \boldsymbol{\beta}, z_b) \propto \sqrt{\tau_b^2}^{-l_b} \exp\left(-\frac{1}{2\tau_b^2} \sum_{k=s_b}^{s_{b+1}-1} (\boldsymbol{\Delta}^{(d)} \boldsymbol{\beta})_k^2 - z_b \tau_b^2\right) \\
&\qquad\qquad\qquad = (\tau_b^2)^{-l_b/2} \exp\left(-\frac{1}{2}\left(\sum_{k=s_b}^{s_{b+1}-1} (\boldsymbol{\Delta}^{(d)} \boldsymbol{\beta})_k^2 (\tau_b^2)^{-1} + 2 z_b \tau_b^2\right)\right) \\
&\Rightarrow \tau_b^2 | \cdot \sim GIG\left(\chi = \sum_{k=s_b}^{s_{b+1}-1} (\boldsymbol{\Delta}^{(d)} \boldsymbol{\beta})_k^2; \psi = 2 z_b; \lambda = 1 - \frac{l_b}{2}\right)
\end{aligned}
$$

$GIG(\chi, \psi, \lambda)$ denotes the generalized inverse Gaussian distribution with density

$$f(x) = \frac{(\psi/\chi)^{\lambda/2}}{2K_\lambda(\sqrt{\psi\chi})} x^{\lambda-1} \exp\left(-\frac{1}{2}\left(\chi x^{-1} + \psi x\right)\right)$$

for $x > 0$, where $K_\lambda(\cdot)$ is the modified Bessel function of the third kind of (fractional) order λ [Jorgensen, 1982].

$$\begin{aligned}
&\pi(\boldsymbol{\beta}|\boldsymbol{\tau}^2, \boldsymbol{l}, \sigma_\varepsilon^2) \propto \exp\left(-\frac{\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2}{2\sigma_\varepsilon^2} - \frac{\boldsymbol{\beta}'\boldsymbol{\Delta}^{(d)\prime}\boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l})^{-1}\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}}{2}\right) \\
&\Rightarrow \boldsymbol{\beta}|\cdot \sim \mathcal{N}_{j-d}\left(\boldsymbol{\mu} = \sigma_\varepsilon^{-2}\boldsymbol{V}\boldsymbol{X}'\boldsymbol{y}; \boldsymbol{\Sigma} = \boldsymbol{V}\right); \\
&\qquad \boldsymbol{V} = \left(\sigma_\varepsilon^{-2}\boldsymbol{X}'\boldsymbol{X} + \boldsymbol{\Delta}^{(d)\prime}\boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l})^{-1}\boldsymbol{\Delta}^{(d)}\right)^{-1} \\
&\pi(\sigma_\varepsilon^2|a_\sigma, b_\sigma, \boldsymbol{\beta}) \propto \sigma_\varepsilon^{2(-a_\sigma - n/2 - 1)} \exp\left(-\frac{\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 + 2b_\sigma}{2\sigma_\varepsilon^2}\right) \\
&\Rightarrow \sigma_\varepsilon^2|\cdot \sim IG(a_\sigma + n/2, b_\sigma + \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2/2)
\end{aligned}$$

B. Reversible Jump Algorithm

B.1. Birth and Death Moves

The prior ratio for the birth step is

$$\begin{aligned}
\mathcal{A}_b =& \frac{\pi_B(B^\star|s_{\text{mean}}, s_{\text{max}})}{\pi_B(B|s_{\text{mean}}, s_{\text{max}})} \frac{\pi_s(\boldsymbol{s}^\star|B^\star)}{\pi_s(\boldsymbol{s}|B)} \frac{\pi_z\left((z_{b^\star}^\star, z_{b^\star+1}^\star)|b_z\right)}{\pi_z(z_{b^\star}|b_z)} \\
& \frac{\pi_\tau\left((\tau_{b^\star}^{2\star}, \tau_{b^\star+1}^{2\star})|\boldsymbol{z}^\star\right)}{\pi_\tau(\tau_{b^\star}^2|\boldsymbol{z})} \frac{\pi_{\Delta\beta}\left(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}|\boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l}^\star)\right)}{\pi_{\Delta\beta}\left(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}|\boldsymbol{T}(\boldsymbol{\tau}^2, \boldsymbol{l})\right)} \\
=& \frac{s_{\text{mean}}}{B^\star} \frac{B}{(j-d-2)} \\
& \frac{b_z^{a_z}}{\Gamma(a_z)} \left(\frac{z_{b^\star}^\star z_{b^\star+1}^\star}{z_{b^\star}}\right)^{a_z} \frac{\exp\left(-(b_z + \tau_{b^\star}^{2\star})z_{b^\star}^\star - (b_z + \tau_{b^\star+1}^{2\star})z_{b^\star+1}^\star)\right)}{\exp\left(-(b_z + \tau_{b^\star}^2)z_{b^\star}\right)} \\
& \frac{\sqrt{\tau_{b^\star}^{2\star}}^{-l_{b^\star}^\star} \sqrt{\tau_{b^\star+1}^{2\star}}^{-l_{b^\star+1}^\star} \exp\left(-\frac{1}{2}\sum_{b=1}^{B^\star}\sum_{k=s_b^\star}^{s_{b+1}^\star-1}(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})_k^2\tau_b^{-2\star}\right)}{\sqrt{\tau_{b^\star}^2}^{-l_{b^\star}} \exp\left(-\frac{1}{2}\sum_{b=1}^{B}\sum_{k=s_b}^{s_{b+1}-1}(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})_k^2\tau_b^{-2}\right)}
\end{aligned}$$

and the proposal ratio for the birth step is

$$\mathcal{P}_b = \frac{p_d(B^\star)}{p_b(B)} \frac{|\{2, \dots, j-d-1\} \setminus \{\boldsymbol{s}\}|}{B} \frac{\pi\left(z_{b^\star}|a_z, b_z, \tau_{b^\star}^2\right)}{\pi\left((z_{b^\star}^\star, z_{b^\star+1}^\star)|a_z, b_z, \tau_{b^\star}^2\right)}$$

$$
\begin{aligned}
&\frac{\pi\left(\tau^2_{b^\star}|\boldsymbol{s},\boldsymbol{l},\tilde{z}_{b^\star}=0.5(z^\star_{b^\star}+z^\star_{b^\star+1}),\boldsymbol{\beta}\right)}{\pi\left((\tau^{2\star}_{b^\star},\tau^{2\star}_{b^\star+1})|\boldsymbol{s}^\star,\boldsymbol{l}^\star,\boldsymbol{z}^\star,\boldsymbol{\beta}\right)}\\
=&\frac{p_d(B^\star)}{p_b(B)}\frac{j-d-B-1}{B}\frac{\Gamma(a_z+1)}{\left(b_z+\tau^2_{b^\star}\right)^{a_z+1}}\frac{z^{a_z}_{b^\star}}{z^{\star a_z}_{b^\star}z^{\star a_z}_{b^\star+1}}\frac{\exp\left(-(b_z+\tau^2_{b^\star})z_{b^\star}\right)}{\exp\left(-(b_z+\tau^2_{b^\star})(z^\star_{b^\star}+z^\star_{b^\star+1})\right)}\\
&\frac{\left(z^\star_{b^\star}+z^\star_{b^\star+1}\right)^{1/2-l_{b^\star}/4}\left(\sum_{k=s^\star_{b^\star}}^{s^\star_{b^\star+1}-1}(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})^2_k\right)^{1/2-l^\star_{b^\star}/4}\left(\sum_{k=s^\star_{b^\star+1}}^{s^\star_{b^\star+2}-1}(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})^2_k\right)^{1/2-l^\star_{b^\star+1}/4}}{2\left(2z^\star_{b^\star}\right)^{1/2-l^\star_{b^\star}/4}\left(2z^\star_{b^\star+1}\right)^{1/2-l^\star_{b^\star+1}/4}\left(\sum_{k=s_{b^\star}}^{s_{b^\star+1}-1}(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})^2_k\right)^{1/2-l_{b^\star}/4}}\\
&\frac{K_{1-l^\star_{b^\star}/2}\left(\sqrt{2\sum_{k=s^\star_{b^\star}}^{s^\star_{b^\star+1}-1}(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})^2_k z^\star_{b^\star}}\right)K_{1-l^\star_{b^\star+1}/2}\left(\sqrt{2\sum_{k=s^\star_{b^\star+1}}^{s^\star_{b^\star+2}-1}(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})^2_k z^\star_{b^\star+1}}\right)}{K_{1-l_{b^\star}/2}\left(\sqrt{\sum_{k=s_{b^\star}}^{s_{b^\star+1}-1}(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})^2_k(z^\star_{b^\star}+z^\star_{b^\star+1})}\right)}\\
&\frac{\sqrt{\tau^2_{b^\star}}^{-l_{b^\star}}}{\sqrt{\tau^{2\star}_{b^\star}}^{-l_{b^\star}}\sqrt{\tau^{2\star}_{b^\star+1}}^{-l_{b^\star+1}}}\\
&\frac{\exp\left(-\frac{1}{2}\left(\sum_{k=s_{b_\star}}^{s_{b^\star+1}-1}(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})^2_k\tau^{-2}_{b^\star}+(z^\star_{b^\star}+z^\star_{b^\star+1})\tau^2_{b^\star}\right)\right)}{\exp\left(-\frac{1}{2}\left(\sum_{k=s^\star_{b^\star}}^{s^\star_{b^\star+1}-1}(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})^2_k\tau^{-2\star}_{b^\star}+\sum_{k=s^\star_{b^\star+1}}^{s^\star_{b^\star+2}-1}(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})^2_k\tau^{-2\star}_{b^\star+1}+2(z^\star_{b^\star}\tau^{2\star}_{b^\star}+z^\star_{b^\star+1}\tau^{2\star}_{b^\star+1})\right)\right)}.
\end{aligned}
$$

This yields acceptance probability

$$
\begin{aligned}
\alpha_b=&\frac{p_d(B^\star)}{p_b(B)}\frac{s_{\text{mean}}}{B^\star}\frac{(j-d-B-1)}{(j-d-2)}\frac{a_z}{\left(b_z+\tau^2_{b^\star}\right)^{a_z+1}}\frac{\exp\left(-\frac{1}{2}\sum_{b=1}^{B^\star}\sum_{k=s^\star_b}^{s^\star_{b+1}-1}(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})^2_k\tau_b^{-2\star}\right)}{\exp\left(-\frac{1}{2}\sum_{b=1}^{B}\sum_{k=s_b}^{s_{b+1}-1}(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})^2_k\tau_b^{-2}\right)}\\
&\frac{\left(z^\star_{b^\star}+z^\star_{b^\star+1}\right)^{1/2-l_{b^\star}/4}\left(\sum_{k=s^\star_{b^\star}}^{s^\star_{b^\star+1}-1}(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})^2_k\right)^{1/2-l^\star_{b^\star}/4}\left(\sum_{k=s^\star_{b^\star+1}}^{s^\star_{b^\star+2}-1}(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})^2_k\right)^{1/2-l^\star_{b^\star+1}/4}}{2\left(2z^\star_{b^\star}\right)^{1/2-l^\star_{b^\star}/4}\left(2z^\star_{b^\star+1}\right)^{1/2-l^\star_{b^\star+1}/4}\left(\sum_{k=s_{b^\star}}^{s_{b^\star+1}-1}(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})^2_k\right)^{1/2-l_{b^\star}/4}}\\
&\frac{K_{1-l^\star_{b^\star}/2}\left(\sqrt{2\sum_{k=s^\star_{b^\star}}^{s^\star_{b^\star+1}-1}(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})^2_k z^\star_{b^\star}}\right)K_{1-l^\star_{b^\star+1}/2}\left(\sqrt{2\sum_{k=s^\star_{b^\star+1}}^{s^\star_{b^\star+2}-1}(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})^2_k z^\star_{b^\star+1}}\right)}{K_{1-l_{b^\star}/2}\left(\sqrt{\sum_{k=s_{b^\star}}^{s_{b^\star+1}-1}(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})^2_k(z^\star_{b^\star}+z^\star_{b^\star+1})}\right)}\\
&\frac{\exp\left(-\frac{1}{2}\left(\sum_{k=s_{b_\star}}^{s_{b^\star+1}-1}(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})^2_k\tau^{-2}_{b^\star}+(z^\star_{b^\star}+z^\star_{b^\star+1})\tau^2_{b^\star}\right)\right)}{\exp\left(-\frac{1}{2}\left(\sum_{k=s^\star_{b^\star}}^{s^\star_{b^\star+1}-1}(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})^2_k\tau^{-2\star}_{b^\star}+\sum_{k=s^\star_{b^\star+1}}^{s^\star_{b^\star+2}-1}(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})^2_k\tau^{-2\star}_{b^\star+1}+2(z^\star_{b^\star}\tau^{2\star}_{b^\star}+z^\star_{b^\star+1}\tau^{2\star}_{b^\star+1})\right)\right)}.
\end{aligned}
$$

B.2. Alternative Proposals

We also experimented with a more complex proposal scheme for the birth and death moves. Specifically, for the birth step we select interval

$b^\star \in \{1, \ldots, B-1\} \setminus \{b : l_b = 1\}$ with probability

$$p(b) \propto l_b^2 \frac{\mathrm{Var}\left((\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})_{s_{b-1},\ldots,s_b-1} \right)}{\sum_{k=s_{b-1}}^{s_b-1} |(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})_k|},$$

placing a higher proposal density on selecting long intervals with a large variation coefficient of the errors of the random walk. This increases the chance of splitting intervals in which both the proportion of small changes in $\boldsymbol{\beta}$ and the variation in $(\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})$ is large. Intervals with those properties are not homogeneous and can potentially benefit from at least one additional changepoint separating the small changes, which may warrant stronger regularization, from the larger ones responsible for the larger variation which potentially reflect jumps or curvature changes in the function to be fitted. The location of the new changepoint $s^\star_{b^\star}$ is then drawn uniformly from
$\{s_{b^\star} + 1, \ldots, s_{b^\star+1} - 1\}$.
In the death step, we select the changepoint $s_{b^\star}$; $b^\star \in \{1, \ldots, B-1\}$ to be removed with probability

$$p(b) \propto \frac{1}{l_b + l_{b+1}} \left| \frac{\sum_{k=s_{b-1}}^{s_b-1} (\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})_k}{l_b} - \frac{\sum_{k=s_b}^{s_{b+1}-1} (\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta})_k}{l_{b+1}} \right|.$$

This increases the chance of removing a changepoint s_b with short adjacent intervals and small difference between the neighboring local means of $\boldsymbol{\Delta}^{(d)}\boldsymbol{\beta}$.
The fitted functions based on these proposals and a uniform prior for the number of knots B were practically identical to fitted functions for the simpler algorithm with a truncated poisson prior for B (see section 2.3). We did not observe any improvement in the sense of a more parsimonious representation of the variance function of the random walk and acceptance probabilities for the dimension changing moves were unreasonably low $(0.1 - 0.2)$ in most cases.

C. Benchmark plots

The benchmarkplot (see section 3 for examples) is a visualization method for benchmark experiments described in Eugster and Leisch [2008]. It is a variant of the dotplot. For every dataset in the benchmark study, algorithms are ranked according to their performance. In the upper panel of the plot, a dotplot is drawn separately for every rank, so that the leftmost part of the upper panel contains a dotplot of the best performances for every dataset and the rightmost part of the upper panel contains a dotplot of the worst performances for every dataset. In this fashion, the ranks of the algorithms on each dataset are used to stretch the plot horizontally. All the dots representing the various results for the same dataset are then connected by lines, similar to a parallel coordinates plot, giving an impression of the differences in the achieved performances on identical data. The order in which the algorithms are plotted for each rank is determined by the frequency of their rankings: the algorithm with the most first places is leftmost, the algorithm with the most last places is the rightmost. The lower panel depicts the podium, a barplot for every rank showing how often each algorithm achieved the respective rank. This plot allows a more detailed visual analysis of benchmark experiments.

References

Veerabhadran Baladandayuthapani, Bani K. Mallick, and Raymond J. Carroll. Spatially adaptive Bayesian penalized regression splines (P-splines). *Journal of Computational and Graphical Statistics*, 14(2):378–394, 2005.

Clemens Biller. Adaptive Bayesian regression splines in semiparametric generalized linear models. *Journal of Computational and Graphical Statistics*, 9(1): 122–140, 2000.

D. G. T. Denison, B. K. Mallick, and A. F. M. Smith. Automatic Bayesian curve fitting. *Journal of the Royal Statistical Society, Series B: Statistical Methodology*, 60:333–350, 1998.

David L. Donoho and Iain M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81:425–455, 1994.

A.L. Drobyshev, C. Machka, M. Horsch, M. Seltmann, V. Liebscher, M.H. de Angelis, J. Beckers, and O. Journals. Specificity assessment from fractionation experiments (SAFE): a novel method to evaluate microarray probe specificity based on hybridisation stringencies. *Nucleic Acids Research*, 31(2):e1, 2003.

P.H.C. Eilers and B.D. Marx. Flexible smoothing with B-splines and penalties. *Statistical Science*, 11(2):89–121, 1996.

Manuel Eugster and Friedrich Leisch. Bench plot and mixed effects models: First steps toward a comprehensiv benchmark analysis toolbox. In *COMPSTAT 2008, 18th Symposium of IASC, Porto*, 2008.

Peter J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82:711–732, 1995.

J.E. Griffin and P.J. Brown. Alternative prior distributions for variable selection with very many more variables than observations. Technical Report UKC/IMS/05/08, IMS, University of Kent, 2005.

J.E. Griffin and P.J. Brown. Bayesian adaptive lassos with non-convex penalization. Technical Report No. 07-02, University of Warwick, 2007. URL http://www2.warwick.ac.uk/fac/sci/statistics/crism/research/working_papers/2007/paper07-2/07-2wv2.pdf.

Torsten Hothorn, Friedrich Leisch, Achim Zeileis, and Kurt Hornik. The design and analysis of benchmark experiments. *Journal of Computational and Graphical Statistics*, 14(3):675–699, 2005.

Torsten Hothorn, Frank Bretz, Peter Westfall, and Richard M. Heiberger. *multcomp: Simultaneous Inference for General Linear Hypotheses*, 2008. R package version 0.993-1.

B. Jorgensen. *Statistical Properties of the Generalized Inverse Gaussian Distribution*. Springer-Verlag, 1982.

T. Krivobokova, C. Crainiceanu, and G. Kauermann. Fast adaptive penalized splines. *Journal of Computational and Graphical Statistics*, (accepted), 2007.

Tatyana Krivobokova. *AdaptFit: Adaptive Semiparametic Regression*, 2007. R package version 0.2-1.

S. Lang, E.M. Fronk, and L. Fahrmeir. Function estimation with locally adaptive dynamic models. *Computational Statistics*, 17:479–500, 2002.

Stefan Lang and Andreas Brezger. Bayesian P-splines. *Journal of Computational and Graphical Statistics*, 13(1):183–212, 2004.

Trevor Park and George Casella. The Bayesian lasso. In *ASA Proceedings of the Joint Statistical Meetings*, pages 125–128. American Statistical Association, 2006.

Jose Pinheiro, Douglas Bates, Saikat DebRoy, and Deepayan Sarkar. *nlme: Linear and nonlinear mixed effects models*, 2006. R package version 3.1-77.

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2007. URL http://www.R-project.org.

David Ruppert and Raymond J. Carroll. Spatially-adaptive penalties for spline fitting. *Australian & New Zealand Journal of Statistics*, 42(2):205–223, 2000.

David J. Spiegelhalter, Nicola G. Best, Bradley P. Carlin, and Angelika van der Linde. Bayesian measures of model complexity and fit (Pkg: P583-639). *Journal of the Royal Statistical Society, Series B: Statistical Methodology*, 64 (4):583–616, 2002.

Mike West and Jeff Harrison. *Bayesian Forecasting and Dynamic Models*. Springer-Verlag Inc, 1989. ISBN 0-387-97025-8.

S. Wood. *mgcv: Multiple Smoothing Parameter Estimation by GCV or UBRE*, 2006. URL `http://www.maths.bath.ac.uk/~sw283/`.