CENTRE FOR APPLIED ECONOMIC RESEARCH

WORKING PAPER
(2006/03)

Locally Adaptive Semiparametric Estimation of the Mean and Variance Functions in Regression Model

By D. Chan, R. Kohn, D. Nott and C. Kirby

ISSN 13 29 12 70
ISBN 0 7334 2324 8
www.caer.unsw.edu.au

Faculty of Commerce and Economics

UNSW
THE UNIVERSITY OF NEW SOUTH WALES

Locally Adaptive Semiparametric Estimation of the Mean and Variance Functions in Regression Models

David Chan Robert Kohn David Nott Chris Kirby *

January 25, 2006

Abstract

This article considers the estimation of a regression model with Gaussian errors, where the mean and the log variance are modeled as a linear combination of explanatory variables. We consider Bayesian variable selection priors and model averaging to obtain efficient estimators when the number of explanatory variables is large. To make the model semiparametric using this framework we allow explanatory variables to enter the mean and log variance models flexibly by representing a covariate effect as a linear combination of basis functions. Our methodology for estimating flexible effects is locally adaptive in the sense that it works well when the flexible effects vary rapidly in some parts of the predictor space but only slowly in other parts. The whole model is estimated using a Markov chain simulation method that samples the posterior distribution with coefficients in the mean model integrated out analytically and highly dependent parameters generated in blocks. The methodology in the paper is applied to a number of simulated and real examples and is shown to work well.

Key Words: Additive model; Bayesian estimation; Markov chain Monte Carlo; Radial basis functions.

1 Introduction

In applied regression modeling it is often important to allow the error variance to be a general function of the covariates for the following reasons. First, the prediction intervals that are obtained using an approach that models the error variance as a function of the covariates are likely to be more realistic than those obtained assuming that the error variance is constant, since estimation of predictive uncertainty depends crucially on estimation of the response variance. Second, it is of interest to understand how the error variance depends on the covariates. Third, it is sometimes important to model the variance flexibly because it is difficult to specify its functional form a priori. Fourth, a correct model for the variance usually gives a more efficient

*David Chan is Manager, Customer Analytics, Cendant - Travel Link Group 1 Sylvan Way Parsippany NJ 07054, USA. Robert Kohn is Professor, Faculty of Commerce and Economics, UNSW SYDNEY NSW 2052, Australia (r.kohn@unsw.edu.au). David Nott is senior lecturer, Department of Statistics, School of Mathematics, UNSW Sydney NSW 2052 Australia. Chris Kirby is Associate Professor of Economics College of Business and Behavioral Science Clemson University Clemson, SC 29634, USA

estimator for the mean function because the weighted least squares estimator of the mean function is usually more efficient than the ordinary least squares estimator in the presence of heteroscedasticity; see, for example Seber (1977).

Heteroscedastic regression, i.e. a regression model with a non-constant variance, has been studied by a number of researchers and a summary of the literature till the late 1980's is given by Carroll and Ruppert (1988). Nonparametric work on heteroscadastic models has mostly been univariate and its aim was to obtain a weighted regression to estimate the mean function more efficiently. See, for example, Carroll (1982), Müller and Stadtmüller (1987), Ruppert et al. (1997), and Dette et al. (1998).

Our article makes the following contributions to the literature. First, a Bayesian approach is presented for simultaneously estimating the mean and variance functions. We provide a general methodology for efficiently estimating a mean and log variance function described as a linear combination of a possibly large number of explanatory variables. Efficient estimators are obtained using Bayesian variable selection priors and model averaging. This framework also allows estimation of flexible effects for some of the explanatory variables. We represent flexible effects through a linear combination of basis functions, making the estimators locally adaptive. This means that they can successfully estimate functions that vary rapidly in one part of the covariate space, but slowly in another part of the space. The effectiveness of the estimators for flexible effects is demonstrated empirically for both univariate and bivariate functions. The methodology also allows both the mean and the log variance to be additive functions of univariate and bivariate components. Each flexible component of the mean and the log variance is expressed as a linear combination of radial basis terms. The local adaptivity of each component is obtained by allowing the basis terms in that component to be included or excluded from the model through the Bayesian variable selection priors.

The second contribution of the paper is the development of an efficient Markov chain simulation method for estimating the model. The simulation method ensures that the sampling scheme generates iterates that converge quickly to the posterior distribution and that mix well. This is achieved as follows: (a) the coefficients of the mean function are integrated out of the sampling scheme, and (b) the indicator variables that determine the basis terms that enter each component are generated in blocks, rather than one at a time, without conditioning on the coefficients of that component.

The methods in this paper build on the work of Smith and Kohn (1996), Denison, Mallick, and Smith (1998) and Holmes and Mallick (1998) who propose locally adaptive methods for nonparametric regression models with constant variance. Our work also builds on the adaptive estimation of generalized linear models in Biller (2000) and Biller and Fahrmeir (2001). Biller (2000) consider adaptive estimation for a single univariate component, and Biller and Fahrmeir (2001) consider additive models with univariate components. The current model is considerably more complex because effectively two sets of additive semiparametric regressions are estimated simultaneously.

Yau and Kohn (2003) estimate heteroscedastic models nonparametrically using penalized splines. Our article provides an important alternative to their approach for the following reasons. First, our methods are semiparametric and allow Bayesian variable selection and model averaging where some or all of the predictors enter the model parametrically. Secondly, the estimator of the error variance can be made locally adaptive by using a large number of basis terms for representing a flexible effect for a predictor combined with variable selection, which chooses the required basis terms. Yau and Kohn (2003) use a single smoothing parameter for each function and are restricted to about 35 basis terms, because using more basis terms results in unacceptably high rejection rates in their Metropolis-Hastings sampling algorithm. Third, the estimation of the whole model is computationally more efficient because it is carried out in a single pass, whereas Yau and Kohn (2003) require two passes. The first pass obtains data-based priors for the smoothing parameters and the second pass estimates the model. Our method also utilizes basis terms efficiently. If only a few basis terms are required, then only these are used, whereas Yau and Kohn (2003) use all the basis terms at every iteration.

The paper is organized as follows. Section 2 presents the model, prior and sampling scheme. It also describes how we model a flexible effect for a single covariate. Section 3 applies the methods in section 2 to estimate flexible effects for univariate simulated and real examples. Section 4 extends the methods to additive models in the mean and the log variance and considers a number of real examples. The appendix gives technical details of the sampling scheme.

2 Model, Prior and Inference

2.1 Model description

Let $y = (y_1, ..., y_n)'$ be a vector of responses and $X = [x_1', ..., x_n']'$ and $Z = [z_1', ..., z_n']'$ be matrices of explanatory variables where $x_i = (x_{i1}, ..., x_{ik})'$ and $z_i = (z_{i1}, ..., z_{iq})'$ are vectors of covariates for the ith response. Consider the model

$$y = \alpha_0 \mathbf{1}_n + X\eta + \epsilon$$

where $\mathbf{1}_n$ is an n-vector of ones, α_0 is an intercept parameter, $\eta = (\alpha_1, ..., \alpha_k)'$ is a vector of unknown parameters, $\alpha = (\alpha_0, \eta')'$ and ϵ is an n-vector of independent zero mean normal errors, $\epsilon_i \sim N(0, \sigma_i^2)$. Write $\Sigma = (\sigma_1^2, ..., \sigma_n^2)^T$ and consider the variance model

$$\log \Sigma = \beta_0 \mathbf{1}_n + Z\beta$$

where again $\mathbf{1}_n$ is an n-vector of ones, β_0 is an intercept term and $\beta = (\beta_1, ..., \beta_k)'$ is a vector of unknown parameters. We have $\sigma_i^2 = \exp(\beta_0)\exp(z_i^T\beta) = \sigma^2 \exp(g(z_i))$ where $\sigma^2 = \exp(\beta_0)$ and $g(z_i) = z_i^T\beta$. We consider Bayesian variable selection priors and model averaging in this general heteroscedastic linear model. We are also concerned with semiparametric modelling where the effects of some explanatory variables enter the models for the mean and log variance flexibly. Section 2.2 explains how a covariate is modeled flexibly and describes the variable selection priors.

2.2 Flexible estimation with a single covariate

Suppose that we are given the observations $(y_i, w_i), i = 1, \ldots, n$, on the dependent variable y and the explanatory variable w. The model we consider is

$$y_i = \mu(w_i) + \sigma(w_i)\varepsilon_i, \tag{2.1}$$

where ε_i is an $N(0, 1)$ independent sequence. In order to express this model in the form given in section 2.1, the mean $\mu(w)$ is expressed as a linear combination of a global intercept term and L basis functions

$$\mu(w) = \alpha_0 + \sum_{j=1}^{L} \alpha_j \phi_j(w)\,.$$

The L basis functions used in our article are

$$\mathcal{B} = \left\{ \phi_1(w) = w \quad \text{and} \quad \phi_j(w) = ||w - \zeta_{j-1}||^2 \log \left(||w - \zeta_{j-1}||^2 \right), j = 2, \ldots, L \right\} , \qquad (2.2)$$

where $||w||$ is the Euclidean norm of w, which is $|w|$ in the univariate case. The abscissae $\zeta_1, \ldots, \zeta_{L-1}$ that determine the basis functions $\phi_2, \ldots, \phi_L$ are called knots and are chosen in the range of the data $w_i, i = 1, \ldots n$. In the univariate case, the knots are determined in the following way. We consider an equally spaced grid of points extending from the minimum to the maximum of the observed predictors. This grid defines a collection of intervals, and the mid-point of an interval is chosen as one of the knots if an observed predictor value lies in the interval. This method of selecting potential knots is simple and extends easily to the multivariate case, where we can cover the observed predictors by a grid of points defining a collection of cells, and choosing the mid-point of a cell as one of the potential knots if an observed predictor value lies in the cell. For more details of this approach to knot choice see Cripps et al. (2006). The basis functions B and knots $\zeta_1, \ldots, \zeta_{L-1}$ should be viewed as potential basis functions and knots because in carrying out nonparametric regression based on variable selection as we describe later relatively few basis functions are included in high posterior probability models. The basis functions $\mathcal{B}$ are called radial basis functions because, for $j \geq 2$, the value of the jth basis function $\phi_j(w)$ is determined by the distance of w from its knot ζ_{j-1}.

The variance function $\sigma^2(w)$ is modeled as $\sigma^2 \exp(g(w))$, with

$$g(w) = \sum_{j=1}^{L} \beta_j \phi_j(w) .$$

That is, the log of the variance function is a linear combination of the intercept $\log(\sigma^2)$ and the L basis functions. Note that we have written down a heteroscedastic linear model of the general form of Section 2.1, where flexible terms in the mean and log variance are represented through basis expansions defined through transformations of the original predictors.

To allow both the mean and the variance functions to be locally adaptive, it is necessary to use a large number of basis terms. However, when L is large the estimates of both the mean and the variance functions are likely to be highly variable locally due to over-fitting. To allow for a large number of potential basis terms while still not over-fitting we consider Bayesian variable selection priors, adapting the framework used by Smith and Kohn (1996) in the constant variance case. This framework allows individual basis terms to be included or excluded from the model by defining the binary variables J_j as 1 if $\phi_j(w)$ is included in the model and $J_j = 0$ if it is not, i.e. $J_j = 0$ when the coefficient α_j is zero. The vector of binary variables $J = (J_1, \ldots, J_L)$ determines the basis functions entering the model for the mean. We assume that the J_j are independent a priori with $\Pr(J_j = 1 \mid \pi_\mu) = \pi_\mu$. It follows that

$$\Pr(J|\pi_\mu) = \pi_\mu^{q(J)} (1 - \pi_\mu)^{L - q(J)} , \qquad (2.3)$$

where $q(J)$ is the number of $\alpha_j,\ j = 1, \ldots, L$, that are nonzero. The vector of indicators K is defined with respect to the vector β in the same way that J is defined with respect to α. Similar to (2.3), we have that

$$\Pr(K|\pi_\sigma) = \pi_\sigma^{q(K)} (1 - \pi_\sigma)^{L - q(K)} , \qquad (2.4)$$

where $q(K)$ is the number of $\beta_j,\ j = 1, \ldots, L$, that are nonzero and π_σ is the probability of an element of K equaling 1.

Remarks

1. Instead of the radial basis $\mathcal{B}$ in (2.2), we can use any other radial basis, a polynomial spline basis as in Smith and Kohn (1996), or even an orthogonal basis such as a wavelet basis. We choose the basis $\mathcal{B}$ because (a) it produces locally adaptive estimates but does not overfit, and (b) it extends in a straightforward way to the multivariate case. However, the choice of an appropriate basis is the subject of ongoing research. Generally we have found that 20 to 30 candidate knots are adequate to achieve locally adaptive estimation, if such local adaptivity is required.

2. We choose the same basis terms, that is the same number of terms as well as the knot locations, for both the mean and the variance functions, because our methods of choosing knots depend only on the spatial configuration of the covariates, not on the values of the dependent variable.

2.3 Priors for the parameters

We now discuss the priors for the parameters in the general model of section 2. Let $\alpha = (\alpha_0, \alpha_1, \ldots, \alpha_k)'$ and $\beta = (\beta_1, \ldots, \beta_q)'$. It is convenient to redefine X as

$$X \leftarrow X - \mathbf{1}_n(\mathbf{1}_n'\mathbf{1}_n)^{-1}\mathbf{1}_n' X\,, \tag{2.5}$$

so that each column of X is mean corrected and hence the columns of X are now orthogonal to $\mathbf{1}_n$. For a given value of β, let

$$D(\beta) = \operatorname{diag}\left(\exp\left(g(z_1)/2\right), \ldots, \exp\left(g(z_n)/2\right)\right)\,, \tag{2.6}$$

and define

$$\tilde{X} = D(\beta)^{-1}\left(\mathbf{1}, X\right) \qquad \tilde{y} = D(\beta)^{-1} y\ . \tag{2.7}$$

Similarly to section 2.2, define $J_j = 1$ if $\alpha_j \neq 0$ and $J_j = 0$ if $\alpha_j = 0$ and write $J = (J_1, ..., J_k)'$. Similarly $K = (K_1, ..., K_q)'$ with $K_j = 1$ if $\beta_j \neq 0$ and $K_j = 0$ if $\beta_j = 0$. For given values of J, let α_J be the subvector of those elements of α that are not identically 0, and let $\tilde{X}_J$ consist of the corresponding columns of $\tilde{X}$ so that $\tilde{X}\alpha = \tilde{X}_J\alpha_J$. For given values of σ^2, β and J, the prior for α_J is

$$\alpha_J \mid \sigma^2, \beta, J \sim N\left(0, \sigma^2 c_\alpha \left(\tilde{X}_J'\tilde{X}_J\right)^{-1}\right)\,, \tag{2.8}$$

with the other elements of α identically zero. This prior is similar to the one used by Smith and Kohn (1996) and Kohn et al. (2001) who interpret it as a shifted and inflated version of the likelihood. The parameter c_α is a scale parameter for those elements of α that are not identically zero.

Let β_K be the subvector of β that is defined with respect to K in the same way that α_J is defined with respect to J. The prior for β_K, for a given value of K, is

$$\beta_K \mid K \sim N(0, c_\beta I), \tag{2.9}$$

where c_β is the scale parameter for the elements of β that are not identically 0.

The priors we use for the three scale parameters σ^2, c_α and c_β are the same, so that it is sufficient to consider the prior for σ^2, which we take as an inverse gamma distribution with density

$$p(\sigma^2) = \frac{b_\sigma^{a_\sigma}}{\Gamma(a_\sigma)}\left(\sigma^2\right)^{-(1+a_\sigma)} \exp\left(-b_\sigma/\sigma^2\right)\ .$$

We choose $a_\sigma = 1 + 10^{-10}$ and $b_\sigma = 1 + 10^{-5}$ to make the prior for σ^2 proper, but highly uninformative. These hyperparameter choices lead to a density with finite mean but infinite variance. The priors for these scale parameters can be made informative and different to one another if prior information is available.

Finally, we explain how the priors for π_μ and π_σ are set. It is sufficient to consider π_μ because the prior for π_σ is similar. We take the prior for the probability π_μ as $\text{Beta}(a_\pi, b_\pi)$, and follow Kohn et al. (2001) in their method for choosing the parameters a_π and b_π by controlling the number of active basis terms, $q(J)$, in the mean function. Specifically, the parameters a_π and b_π are obtained by solving the pair of simultaneous equations

$$\begin{aligned}\frac{a_\pi}{a_\pi + b_\pi} &= E(q(J))/L\,, \\ \frac{a_\pi}{a_\pi + b_\pi + 1} &= \frac{\text{var}(q(J)) - E(q(J))(1 - E(q(J)))}{(L-1)E(q(J))}\,,\end{aligned}$$

where the values for $E(q(J))$ and $\text{var}(q(J))$ are chosen by the user. These equations can be derived by observing that the marginal prior distribution for $q(J)$ is beta-binomial, and hence we can write down expressions for its prior mean and variance. To make use of these equations for setting the prior, suppose, for example, that $k = 30$ and we would like the range of $q(J)$ values with high probability to be between 0 and 22 in the prior. We could achieve this by setting $E(q(J)) = 10$ and $\text{sd}(q(J)) = 6$ and solving the equations above to obtain $a_\pi = 1.864$ and $b_\pi = 3.727$.

Remarks

1. The models for the mean and the log variance always include an intercept term, which means X can be replaced, without loss of generality, by

$$X - \mathbf{1}_n(\mathbf{1}_n'\mathbf{1}_n)^{-1}\mathbf{1_n}'X$$

and similarly for Z. This ensures that the estimators are invariant to changes in the location of the dependent variable. This transformation also produces faster convergence and better mixing in both the mean and the variance functions than that obtained by using the original X and Z matrices.

2. In addition to carrying out element selection in both the mean and the variance functions, the parameters c_α and c_β act as scale factors for the nonzero elements of α and β. For the current problem, including data-adaptive scale parameters was more satisfactory than using fixed scale parameters as in Smith and Kohn (1996). Note that the prior for σ^2 is independent of the prior for β, which means that $\log\left(\sigma^2\right)$ is completely separate from the scale for β.

2.4 Bayesian inference

The purpose of modeling the mean and variance functions $\mu(x) = \alpha_0 + x^T\eta$ and $\sigma^2(z) = \sigma^2\exp(g(z))$ is to estimate them, and to carry out inference on a future observation y_*, given its covariates x_* and z_* and the past observations y. The Bayesian approach usually estimates the mean and the variance functions by their posterior means and uses the predictive distribution $p(y_* \mid x_*, z_*, y)$ to predict y_* and to obtain prediction intervals for it. The complex hierarchical model introduced in sections 2.1 and 2.3 makes it computationally intractable to explicitly evaluate the posterior means of $\mu(x)$ and $\sigma^2(z)$, and the predictive distribution $p(y_* \mid x_*, z_*, y)$, and it is necessary to estimate these terms using simulation. Let

$$\Theta = \left\{\alpha, J, \beta, K, \sigma^2, c_\alpha, c_\beta\right\} \tag{2.10}$$

be the vector of parameters and latent variables in the model. In this section we suppose that there exist iterates $\Theta_k, k = 1_, \ldots, M$, that are generated from the posterior distribution $p(\Theta \mid y)$ and are used for posterior inference. Section 2.5 presents an efficient sampling scheme to generate these iterates. To estimate the posterior mean of $\mu(x)$, we note that

$$E\left(\mu(x) \mid y\right) = \int E\left(\mu(x) \mid \Theta\right) p(d\Theta \mid y)\,,$$

which is estimated by

$$\hat{\mu}(x) = \frac{1}{M}\sum_{k=1}^{M} \mu^{[k]}(x)\,,$$

where

$$\mu^{[k]}(x) = \alpha_0^{[k]} + x^T \eta^{[k]}$$

and $\eta^{[k]}$, $k = 1, ..., M$ represent the iterates obtained for η in the sampling scheme after discarding a suitable "burn in" period. The posterior mean of $\sigma^2(z)$ is estimated similarly. The predictive density $p(y_* \mid x_*, z_*, y) = \int p(y_* \mid x_*, z_*, \Theta) p(d\Theta \mid y)$ and is estimated by

$$\hat{p}(y_* \mid x_*, z_*, y) = \frac{1}{M}\sum_{k=1}^{M} p(y_* \mid x_*, z_*, \Theta^{[k]}).$$

where $\Theta^{[k]}$, $k = 1, ..., M$, represent the iterates obtained for Θ in the sampling scheme.

2.5 Markov chain Monte Carlo simulation

This section presents the sampling scheme for generating Θ (defined in (2.10)), which is the vector containing all the parameters and latent variables in the model, and presents a brief explanation of each of the steps. The technical details of how Θ is generated are given in the appendix. The sampling scheme consists of an initialization step followed by 5 steps that are repeated at each iteration. It is assumed that after a sufficiently large number of iterations, called the 'warmup' or 'burn-in' period, the sampler produces iterates from the posterior distribution and it is these iterates that are used for inference. A sixth step, which is only carried out after the burn-in period, generates α. However, for reasons of efficiency that are discussed later, all the other elements of Θ are generated without conditioning on α. To describe the sampling scheme, we require the following notation. For a given set of latent variables or parameters ϕ, the notation $\Theta_{-\phi}$ means all elements of Θ with ϕ excluded. We use Metropolis-Hastings kernels in the sampling schemes to update subsets of Θ; see Liu (2001) for an introduction to the Metropolis-Hastings method and MCMC.

Sampling Scheme 1

0. Initialize Θ.

1. Generate J, conditional on y and $\Theta_{-\{J,\alpha\}}$, as follows. The elements of the vector J are first permuted randomly and then generated in blocks. Each block size is randomly chosen as 2, 4 or 6 with probability 1/3. Let J_B be one such block. Then J_B is generated from the prior $p(J_B \mid J_{-B})$, with π_μ integrated out. This proposed value is then either accepted or rejected according to the Metropolis-Hastings rule.

2. The parameter c_α is generated from a normal approximation to its conditional distribution, given y and $\Theta_{-\{c_\alpha,\alpha\}}$, and this proposed value is accepted or rejected according to the Metropolis-Hastings rule.

3. The vector K is generated in blocks similarly to the way that J is generated. For a given subvector K_B of K, K_B is generated from the prior $p(K_B \mid K_{-B})$ with π_σ integrated out. Given this generated value K_B^p of K_B, let $K^p = \{K_B^p, K_{-B}\}$. Then, the vector β_{K^p} is generated from a Gaussian approximation to $p(\beta_{K^p} \mid y, J, K^p, c_\beta, \sigma^2)$. The proposed value for (K_B, β) is then accepted or rejected according to the Metropolis-Hastings rule.

4. The parameter c_β is generated from a normal approximation to its conditional distribution and the proposal is accepted or rejected according to the Metropolis-Hastings rule.

5. The variance σ^2 is generated from its full conditional distribution, which is an inverse gamma distribution.

6. The vector α_J is generated from its full conditional distribution, which is normal with mean vector and covariance matrix

$$\hat{\alpha} = \frac{c_\alpha}{(1+c_\alpha)}\tilde{X}_J\left(\tilde{X}_J'\tilde{X}_J\right)^{-1}\tilde{X}_J'\tilde{y} \quad \text{and} \quad \hat{V}_\alpha = \frac{\sigma^2 c_\alpha}{1+c_\alpha}\left(\tilde{X}_J'\tilde{X}_J\right)^{-1}.$$

We make the following important remarks about the sampling scheme.

Remarks

1. The sampling scheme is run in two stages. The first stage is called the warmup or burn-in period. At the end of this period it is assumed that the iterates are generated from the posterior distribution. The second stage is called the sampling period and iterates from this period are used for inference.

2. Step 6, which generates α, is only carried out in the sampling period because α is not required in steps 1 to 5. This means that the convergence and mixing properties of the sampling scheme depend only on steps 1 to 5, and that step 6 is only used to carry out inference on functionals that depend on α.

3. We found that it was necessary to integrate out α from steps 1 to 5. Generating β and σ^2 conditional on α produced a sampling scheme that converged very slowly and produced estimates with a high variability because the iterates were highly autocorrelated. That is, conditioning on α in steps 1 to 5 was unsatisfactory.

4. The vector J is generated in blocks using the prior as a proposal distribution. This produced computationally efficient and reliable estimators. Our choice of block size was determined by the consideration that a large block size is best for allowing moves that combat the detrimental effects of dependence between components on mixing, but on the

other hand it is difficult to come up with a good block proposal for large blocks without suffering from very high rejection rates. Randomly choosing between blocks of size 2, 4 and 6 allows us to maintain a fairly high acceptance rate in the Metropolis-Hastings scheme, but still update moderately sized groups of dependent parameters in blocks. The basic idea of this scheme and the block generation of J is based on the proposals in Kohn et al. (2001) and is much more efficient than generating J in blocks of one or more from the exact conditional distribution $p(J_B \mid y, J_{-B}, \beta, K, \sigma^2)$.

The vector J is generated in blocks using the prior as a proposal distribution. The blocks are chosen at random to be of size 2, 4 or 6, which gives a relatively high acceptance rate in the Metropolis-Hastings scheme, but still updates moderately sized groups of parameters in blocks. Or approach is a compromise between using large block sizes which allow moves that overcome the detrimental effects on mixing of dependence between components, while avoiding unduly high rejection rates. The basic idea of this sampling scheme and the block generation of J is based on the proposals of Kohn et al. (2001). The sampling scheme that generates J in blocks of one or more from the exact conditional distribution is much less efficient than generating from the prior because it is necessary to factor the matrix $\tilde{X}_J'\tilde{X}_J$ using the Cholesky decomposition for each new configuration of J.

5. The vector K is generated in blocks from the prior distribution similarly to the way that J is generated. For the proposed value K^p of K, we have found that it necessary to generate the whole vector β to make it consistent with the value of K^p. Poor convergence and very high rejection rates were obtained if only that part of β is generated that corresponds to the generated subvector K_B of K. A compromise between updating the whole of β and updating only the subvector corresponding to K_B is to update a larger subset of the active components than that defined by the current block, but still not the whole vector. We adopt this approach later when we discuss handling multiple covariates flexibly in additive models.

3 Univariate results

This section considers the model (2.1) where we model the mean and variance flexibly as a function of a single predictor. Four simulated examples are described. In example 1, the true mean and the standard deviation functions are linear in w. In example 2, the standard deviation is a locally inhomogeneous function of w, in example 3 it is the mean that is a spatially inhomogeneous function of w, and in example 4 both the mean and the standard deviation are locally inhomogeneous function of w. These four examples cover the four possible combinations of spatially homogenous and inhomogeneous functions for the mean and standard deviation functions. Nonparametric estimators such as spline and kernel smoothers that use a single smoothing parameter do not estimate such spatially inhomogeneous functions satisfactorily, and it is necessary to use estimators that are locally adaptive.

In all four examples, the sample size was $n = 500$ and the explanatory variable w was generated uniformly on the interval $(0, 1)$.

Example 1

$$\mu(w) = 2w \quad \text{and} \quad \sigma(w) = 0.1 + w,$$

so that both the mean and standard deviation functions are linear in w.

Table 3.1: The table shows the average and standard deviation (in brackets) of the number of active knots for both the mean and the variance functions for the four simulated examples and for the three values of 30, 50 and 100 for the potential number of knots. The first three rows display the number of active knots for the mean function and the last three rows display the number of active knots for the variance function.

Number of potential knots	Example 1	Example 2	Example 3	Example 4
Mean - 30 knots	2.26(0.51)	2.03(0.16)	6.92(1.16)	8.55(1.50)
50 knots	2.27(0.57)	2.02(0.13)	6.97(1.25)	8.35(1.37)
100 knots	2.23(0.48)	2.04(0.20)	6.88(1.12)	10.01(1.63)
Variance - 30 knots	15.30(6.00)	18.22(6.75)	19.85(5.34)	19.80(5.55)
50 knots	15.06(6.43)	19.31(7.09)	19.33(6.83)	15.83(6.62)
100 knots	15.21(7.10)	18.23(6.82)	20.71(7.96)	18.12(7.46)

Example 2

$$\mu(w) = 2(1-w) \quad \text{and} \quad \sigma(w) = \left(N(w;\mu_1,\tau_1^2) + N(w;\mu_2,\tau_2^2))\right)/6\,,$$

where $N(w;\mu,\tau^2)$ means a normal density with mean μ and variance τ^2, and $\mu_1 = 0.2, \tau_1^2 = 0.004, \mu_2 = 0.6$ and $\tau_2^2 = 0.1$.

Example 3

$$\mu(w) = \left(N(w;\mu_1,\tau_1^2) + N(w;\mu_2,\tau_2^2))\right)/4 \quad \text{and} \quad \sigma(w) = 0.6 + 0.5\sin(2\pi w)\,,$$

where μ_1, τ_1^2, μ_2 and τ_2^2 are as in example 2.

Example 4

$$\begin{aligned} \mu(w) &= \left(N(w;\mu_1,\tau_1^2) + N(w;\mu_2,\tau_2^2)\right)/4 \quad \text{and} \\ \sigma(w) &= \left(N(w;\mu_1,\tau_1^2) + N(w;\mu_2,\tau_2^2)\right)/6, \end{aligned}$$

where μ_1, τ_1^2, μ_2 and τ_2^2 are as in example 2.

For the four examples, the estimation results are reported when 30, 50 and 100 knots were used. To determine the parameters in the prior for π_μ, we set $E(q(J)) = 10$ and $\text{sd}(q(J)) = 6$. A similar prior was set for π_σ. For all four examples, the sampling scheme converged quickly, needing less than 100 iterations for convergence. For these examples, we ran the sampling scheme using 200 burn-in iterations followed by 1000 iterations for inference, which we judged to be sufficient to explore the posterior distribution. We have also experimented with using longer runs, but the results are similar. The mean and variance function estimates were constructed as outlined in section 2.4.

Table 3.1 summarizes the average number and the standard deviation of the number of active knots used for both the mean and the standard deviation functions, when the total number of knots was 30, 50 and 100. The table shows that more active knots are required as the mean and variance functions become more locally inhomogeneous. The table also shows that the number of active knots required stays about the same as the total number of knots was increased from 30 to 50 to 100. The number of knots needed in estimating the variance functions is quite large, but we are still able to explore the posterior distribution efficiently, which testifies to the effectiveness of our block sampling method for the indicators.

Figures 3.1 - 3.4 plot the mean and standard deviation function estimates, when 30 knots are used. The function estimates for 50 and 100 knots are similar. The top panel in each of figures 3.1 - 3.4 displays the actual data, together with the true and estimated mean functions, $\mu(w)$ and $\widehat{\mu}(w)$. Each of the panels also contains the estimated 68% prediction interval bands, i.e., $\widehat{\mu}(w) \pm \widehat{\sigma}(w)$ and the true 68% prediction interval bands, i.e., $\mu(w) \pm \sigma(w)$. In the top panels, the true functions are given as dashed lines and the estimated functions are given as dotted lines. The bottom left panel in each figure plots the true $\mu(w)$ (solid line) and the posterior mean estimate $\widehat{\mu}(w)$ (dashed line). The bottom right panel in each figure plots the true standard deviation function $\sigma(w)$ (solid line) and the estimate of the posterior mean $\widehat{\sigma}(w)$ (dashed line). The plots demonstrate that our estimators are locally adaptive and can capture both spatially homogenous and spatially inhomogeneous mean and variance functions. The times taken by the sampling scheme for 1200 iterations were 75 seconds, 83 seconds, 81 seconds and 91 seconds respectively for the first, second, third and fourth examples using a program written in MATLAB and run on a PC with Pentium IV 2.4GHz processor.

At the suggestion of one of the referees we show in Figure 3.5 a plot similar to Figure 3.4 for Example 4, but where the sample size is 100 rather than 500. Similar plots for Examples 1-3 are not given due to space constraints. This example shows that the methodology still works well for a smaller sample size.

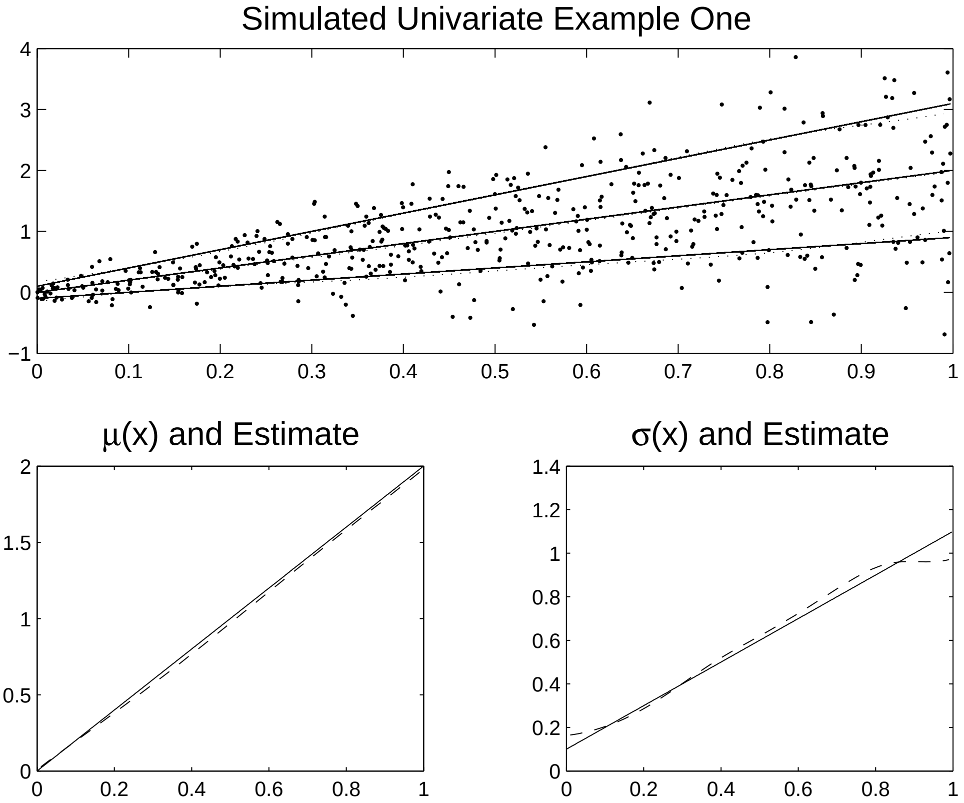


Figure 3.1: Results for the first simulated univariate example

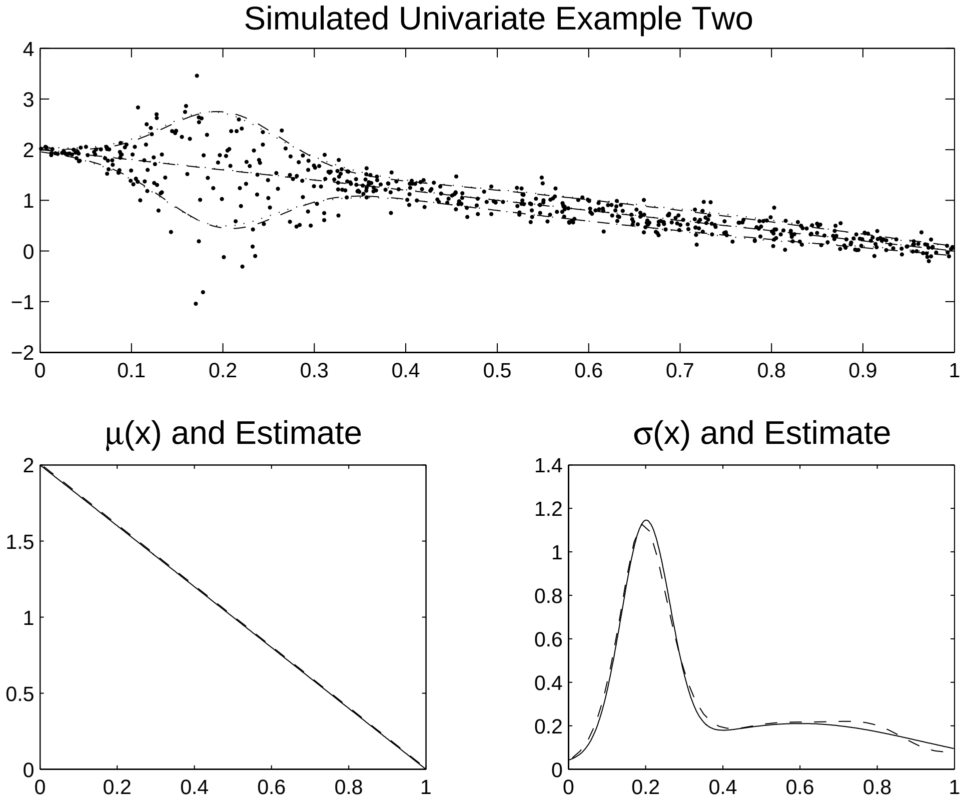


Figure 3.2: Results for the second simulated univariate example

4 Multivariate covariates

This section generalizes the results in section 2 to handle estimation of flexible effects for multivariate covariates and models that are additive in the mean function and the log of the variance.

4.1 Flexible effects for bivariate covariates

Except for the following minor amendment to the basis, the univariate treatment in section 2.2 extends to the bivariate case. Suppose that $w = (u, v)$. The radial basis is now

$$\mathcal{B} = \{u, v, \phi_3(w), \ldots, \phi_L(w)\}\,, \tag{4.1}$$

where, for $j \geq 3$, the $\phi_j(w)$ are the radial basis functions discussed in section 2.2. In the bivariate case, the knot locations are obtained using the clustering routine called *clara* developed by Kaufman and Rousseeuw (1990).

To demonstrate the method we generated a simulated data set using $n = 500$ observations. The bivariate w was generated uniformly from the unit square, and the dependent variable was generated from the model (2.1). The mean and the standard deviation functions were

$$\begin{aligned} \mu(w) &= 0.1 + N(w; \mu_1, \Sigma_1) + N(w; \mu_2, \Sigma_2)\,, \\ \sigma(w) &= 0.1 + \left(N(w; \mu_1, \Sigma_1) + N(w; \mu_2, \Sigma_2)\right)/2\,, \end{aligned}$$

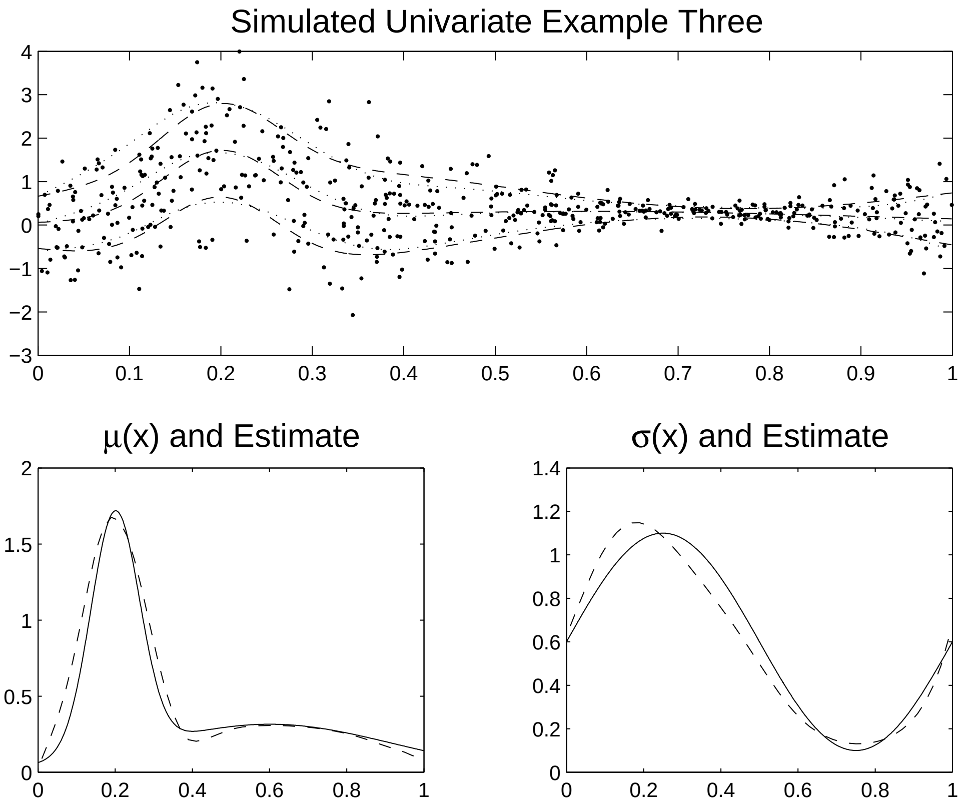


Figure 3.3: Results for the third simulated univariate example

where

$$\mu_1 = \begin{pmatrix} 0.25 \\ 0.75 \end{pmatrix},\ \Sigma_1 = \begin{pmatrix} 0.03 & 0.01 \\ 0.01 & 0.03 \end{pmatrix},\ \mu_2 = \begin{pmatrix} 0.65 \\ 0.35 \end{pmatrix}\ , \Sigma_1 = \begin{pmatrix} 0.09 & 0.01 \\ 0.01 & 0.09 \end{pmatrix}.$$

Knots were constructed as described in Section 2.2 where an equally spaced 10×10 grid on $[0,1]^2$ was considered defining a collection of cells and the centre of each cell containing one of the observed predictors was chosen as a potential knot. We set $E(q(J)) = 15$ and $\text{sd}(q(J)) = 6$ reflecting our belief that more knots may be required to estimate bivariate surfaces. Figure 4.1 plots the true and estimated mean and variance functions and shows that the estimates are very close to the true values. Run time for the Markov chain Monte Carlo sampling scheme in this example for 1200 iterations was 491 seconds using a program written in MATLAB and run on a Pentium IV 2.4 GHz PC. Again we have experimented with longer runs but attain near identical fits to the data based on these longer runs.

4.2 Additive models

In this section we consider the model (2.1) with w multivariate and with the mean and log variance additive functions of a number of components,

$$\mu(w) = \alpha_0 + \sum_{k=1}^{r} f_k(s_k) \quad \text{and} \quad g(w) = \sum_{k=1}^{r} g_k(s_k),$$

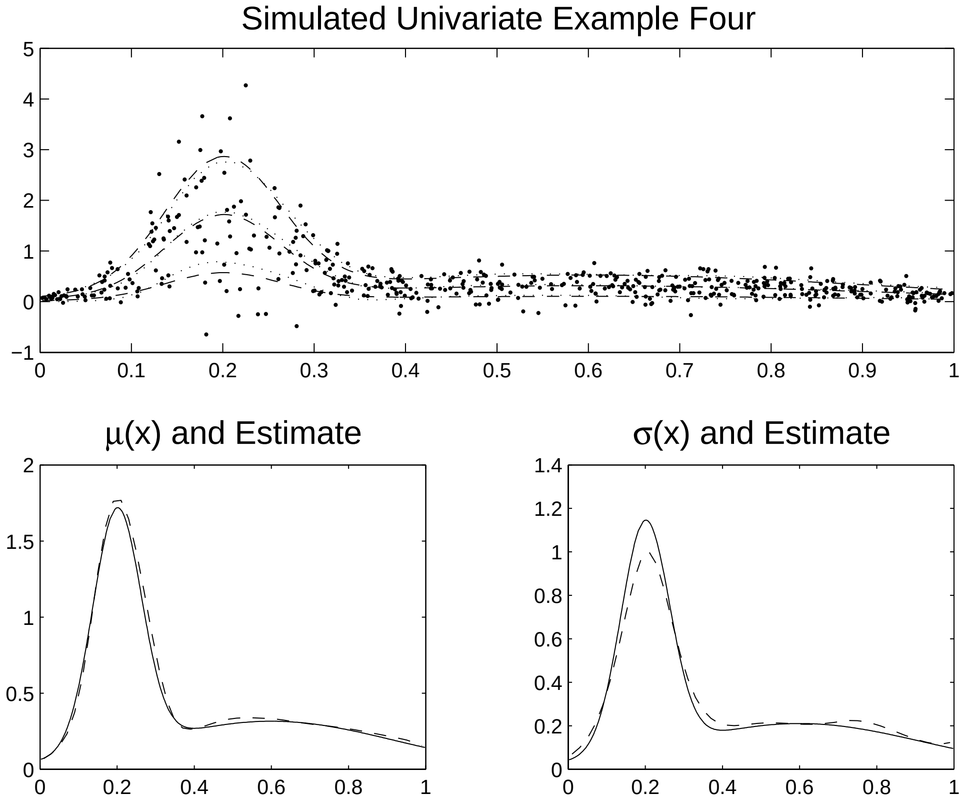


Figure 3.4: Results for the fourth simulated univariate example

where w is partitioned into r components as $w = (s_1, \dots, s_r)$, with each s_k either univariate or bivariate. Each of the components $f_k(s_k)$ and $g_k(s_k)$ is expressed as a linear combination of basis functions as in section 2.2, with

$$f_k(s_k) = \sum_{j=1}^{L_k} \alpha_{kj}\phi_{kj}(s_k) \quad \text{and} \quad g_k(s_k) = \sum_{j=1}^{L_k} \beta_{kj}\phi_{kj}(s_k)\,.$$

The basis functions ϕ_{kj} are defined as in sections 2.2 and 4.1.

To allow basis functions to be included or excluded in $f_k(s_k)$ and $g_k(s_k)$, we introduce the vectors of binary indicator variables $J_{k.} = (J_{k1}, \dots, J_{kL_k})$ and $K_{k.} = (K_{k1}, \dots, K_{kL_k})$, where J_{kj} has the same interpretation for ϕ_{kj} as J_j in section 2.1 had for ϕ_j. Let $J = (J_{1.}, \dots, J_{r.})$ and $K = (K_{1.}, \dots, K_{r.})$. Similarly to (2.3) and (2.4), we assume that

$$\Pr(J_{k.}|\pi_{\mu_k}) = {\pi_{\mu_k}}^{q(J_{k.})}\,(1-\pi_{\mu_k})^{L_k - q(J_{k.})} \tag{4.2}$$

and

$$\Pr(K_{k.}|\pi_{\sigma_k}) = {\pi_{\sigma_k}}^{q(K_{k.})}\,(1-\pi_{\sigma_k})^{L_k - q(K_{k.})}\;. \tag{4.3}$$

We now describe the priors on the parameters. Let $\alpha_{k.} = (\alpha_{k1}, \dots, \alpha_{kL_k})'$, $\alpha = \left(\alpha_0, \alpha'_{1.}, \dots, \alpha'_{r.}\right)'$, $\beta_{k.} = (\beta_{k1}, \dots, \beta_{kL_k})'$ and $\beta = (\beta'_{1.}, \dots, \beta'_{r.})'$. The design matrices X and Z are constructed as in section 2.3 and then transformed as in (2.5). The diagonal matrix is defined as in (2.6) and the matrix $\tilde{X}$ and vector $\tilde{y}$ are then given by (2.7). For given J and K, define α_J and β_K as in

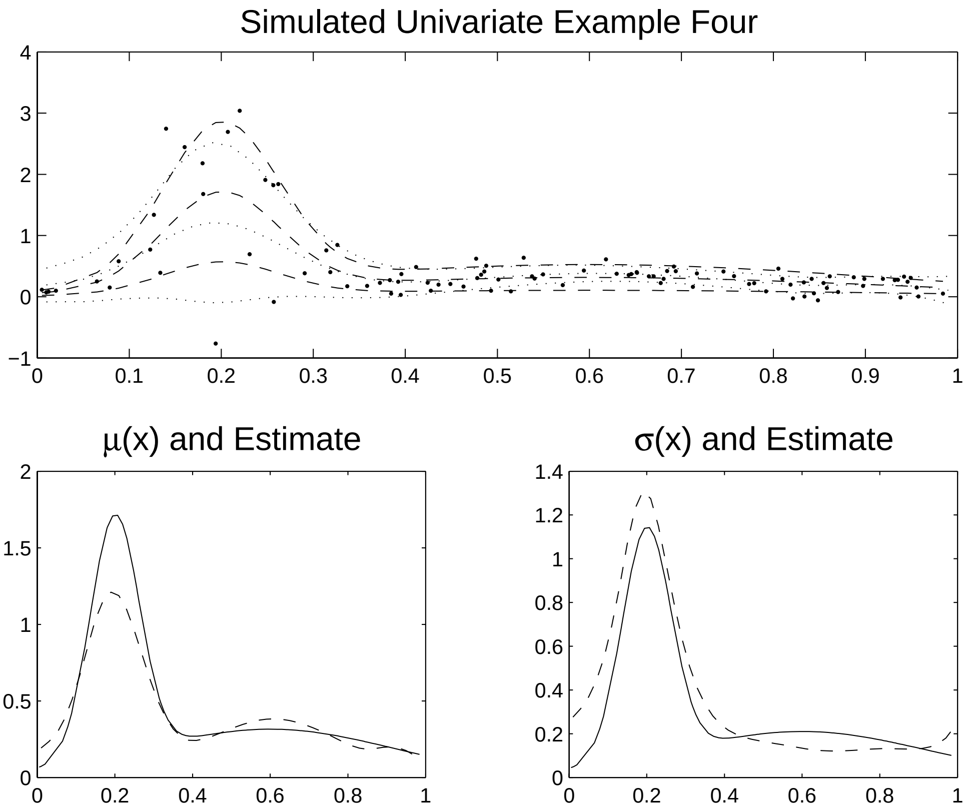


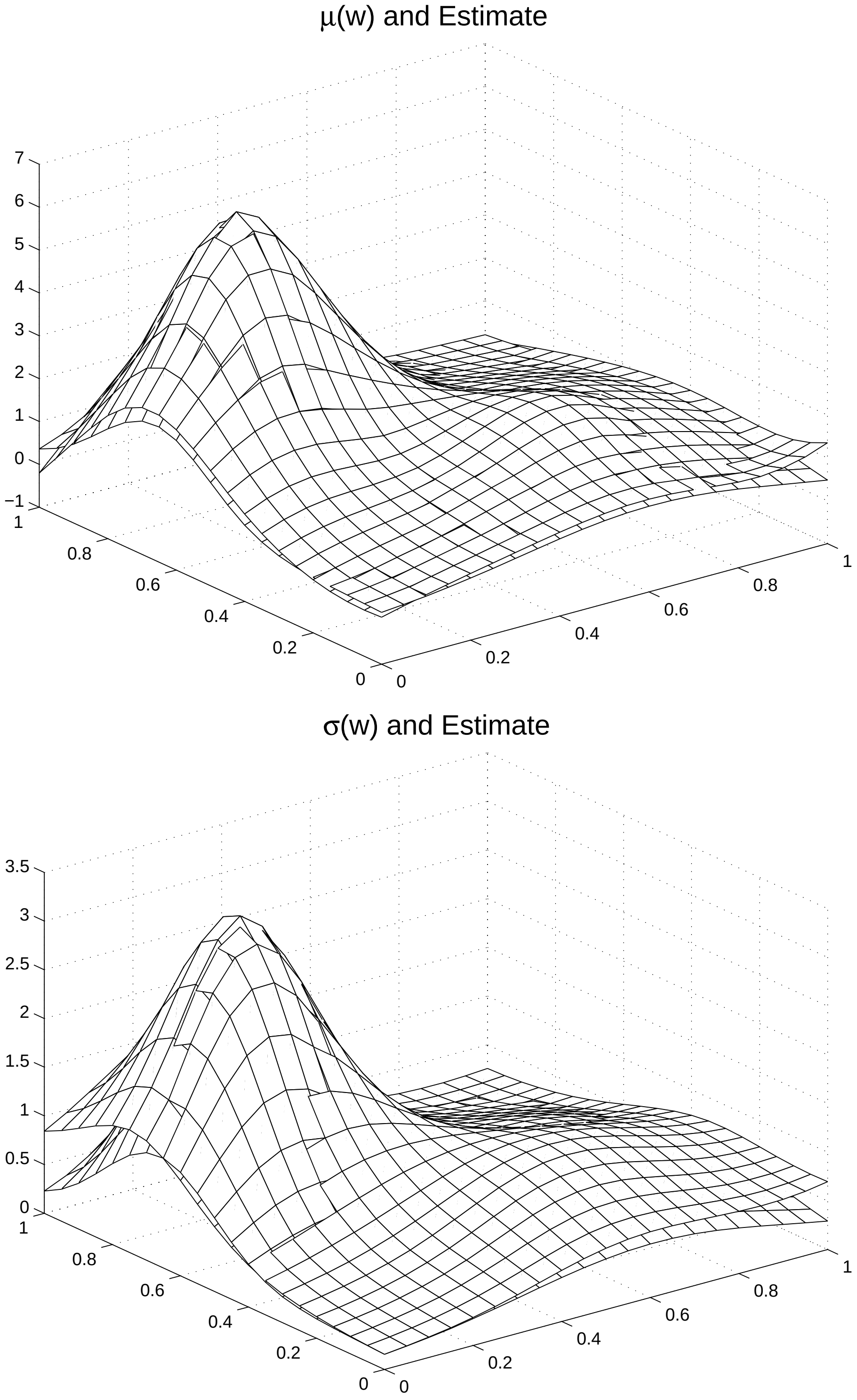
Figure 3.5: Results for the fourth simulated univariate example with n=100

section 2.3, and take their priors as (2.8) and (2.9). The prior specification for the three scale parameters σ^2, c_α and c_β is the same as in section 2.3. Finally, the prior specification for the probabilities π_{μ_k} and π_{σ_k}, $k = 1, \ldots, r$, is the same as for π_μ and π_σ in section 2.3. We note that it is very easy to allow different scale parameters in the prior for the blocks $\beta_{k.}$ rather than having just a single scale parameter c_β if desired since the full conditional distributions of these parameters can be calculated in closed form as inverse gamma distributions.

Next, sampling scheme 1 in section 2.5 is modified for the additive case.

Sampling scheme 2

0. Initialize Θ.
1. For $k = 1, \ldots, r$, generate $J_{k.}$, conditional on y and $\Theta_{-\{\alpha, J_{k.}\}}$. The vector $J_{k.}$ is generated in blocks in the same way that the vector J was generated in step 1 of sampling scheme 1.
2. The parameter c_α is generated as in step 2 of sampling scheme 1.
3. For $k = 1, \ldots, r$, generate $K_{k.}$ and $\beta_{k.}$, conditional on y and $\Theta_{-\{\alpha, J_{k.}, \beta_{k.}\}}$, as in step 3 in sampling scheme 1 .
4. The parameters c_β, σ^2, and α are generated as in steps 4 to 6 of sampling scheme 1.



μ(w) and Estimate

σ(w) and Estimate

Figure 4.1: Results for the simulated bivariate example

We make the following remarks about the sampling scheme.

Remarks

1. The sampling scheme is very similar to sampling scheme 1 except that steps 1 and 3 are now carried out component by component. This makes the simulation more tractable because it is only necessary to generate the active components of $\beta_{k.}$ and not the whole of β.
2. For simplicity we use the same scale factors c_α and c_β for all components of α and β respectively.

4.3 Simulated additive example

The method in section 4.2 is illustrated using a model with a four dimensional covariate vector having four univariate components. Thus $w = (s_1, \ldots, s_4)$, and the data is generated from the model (2.1), with $n = 500$ and

$$\mu(w) = \sum_{k=1}^{4} \mu_k(s_k) \quad \text{and} \quad \sigma(w) = \prod_{k=1}^{4} \sigma_k(s_k).$$

$$\begin{array}{lcllcl}
\mu_1(s_1) & = & 1.5 s_1, & \sigma_1(s_1) & = & \left(N(s_1;\theta_1,\tau_1^2) + N(s_1;\theta_2,\tau_2^2)\right)/2 \\
\mu_2(s_2) & = & \left(N(s_2;\theta_1,\tau_1^2) + N(s_2;\theta_2,\tau_2^2)\right)/2, & \sigma_2(s_2) & = & 0.6 + 0.5\sin(2\pi s_2) \\
\mu_3(s_3) & = & 1 + \sin(2\pi s_3), & \sigma_3(s_3) & = & 1.1 - s_3 \\
\mu_4(s_4) & = & -s_4, & \sigma_4(s_4) & = & 0.2 + 1.5 s_4
\end{array}$$

The model in section 4.2 takes the grand mean α_0 out of $\mu(w)$ and the grand mean $\log(\sigma^2)$ out of the log variance function. In doing so, the component estimates of the mean and standard deviation functions differ from the mean and standard deviation functions used to generate the data. All the mean functions should be centered around 0 as each column of X is mean corrected by the transformation given in (2.5). In a similar fashion, all the log standard deviation functions should be centered around 0, which implies the standard deviation functions should be approximately centered around 1. To make comparison easier in the plots of $\mu_k(s_k)$ and $\widehat{\mu}_k(s_k)$ we have shifted $\mu_k(s_k)$ and $\widehat{\mu}(s_k)$ so that the average value of these functions over the observed predictors is zero. Similarly, we have scaled $\sigma_k(s_k)$ and $\widehat{\sigma}_k(s_k)$ so that the average of $\log \sigma_k(s_k)$ and $\log \widehat{\sigma}_k(s_k)$ is zero over the observed predictor values.

Figures 4.2 and 4.3 report the estimation results. The figures show that the true shapes of both the mean and standard deviation functions are recovered quite well for all four components. The mean and standard deviation functions consisted of both locally homogeneous and locally inhomogeneous components. Run time for the Markov chain Monte Carlo sampling scheme in this example for 1200 iterations was 812 seconds for a program written in MATLAB and run on a Pentium IV 2.4 GHz PC. Again we have experimented with longer runs but attain near identical fits to those obtained above.

4.4 Sniffer data

We consider a data set described in Weisberg (1985), Example 6.2, and Smyth (1989). The data are concerned with the hydrocarbon vapours which escape when petrol is pumped into a tank.

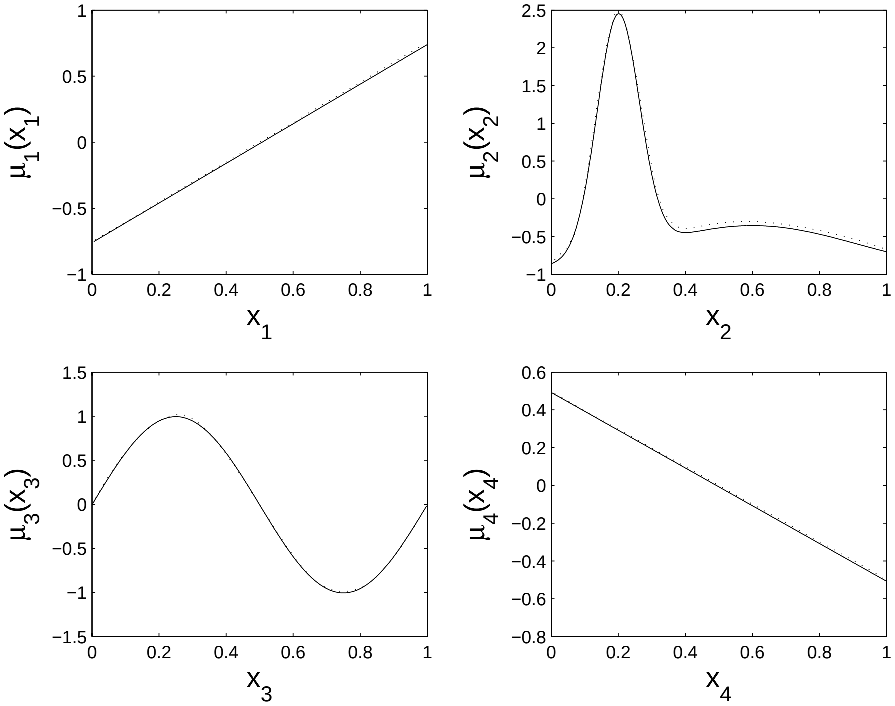


Figure 4.2: The true and estimated mean functions for the additive model

Petrol pumps are fitted with vapour recovery systems, which may only be partially effective. “Sniffer” devices are able to detect if some vapour is escaping. To estimate the efficiency of vapour recovery systems, an experiment was conducted in which the amount of hydrocarbon vapour given off (y), in grams, was measured, along with four predictor variables. The four predictor variables were initial tank temperature (x_1), in degrees Fahrenheit, the temperature of the dispensed gasoline (x_2), in degrees Fahrenheit, the initial vapor pressure in the tank (x_3), in pounds per square inch, and the initial vapour pressure of the dispensed gasoline (x_4), in pounds per square inch. Weisberg (1985) and Smyth (1989) consider only a subset of the full data set, but we consider the full data set here which is available at `www.statsci.org/data/`.

Here we consider a semiparametric model of the form

$$\mu(w) = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 + f(x_4)$$

$$g(w) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4.$$

This model was arrived at by first fitting a linear model for the mean and log variance and then examination of appropriate diagnostics suggested modeling the effect of x_4 flexibly. Fitting the model gives the results shown in Figures 4.4 and 4.5. The plots of the means μ_i and standard deviations σ_i, $i = 1, \ldots, 4$, are constructed in the same way as described at the end of the first paragraph of Section 4.3. The estimated posterior probabilities of inclusion for the predictors $x_1, ..., x_4$ in the mean model were 0.076, 1, 0.014 and 1 respectively. The estimated posterior probabilities of inclusion for the predictors $x_1, ..., x_4$ in the variance model were 0.56, 1, 0.8 and 0.82 respectively. These results were obtained based on 1000 sampling iterations with 200 iterations burn in for our sampling scheme. Longer runs gave similar results. Run time for the sampling scheme in this example for 1200 iterations was 33 seconds using a program written in

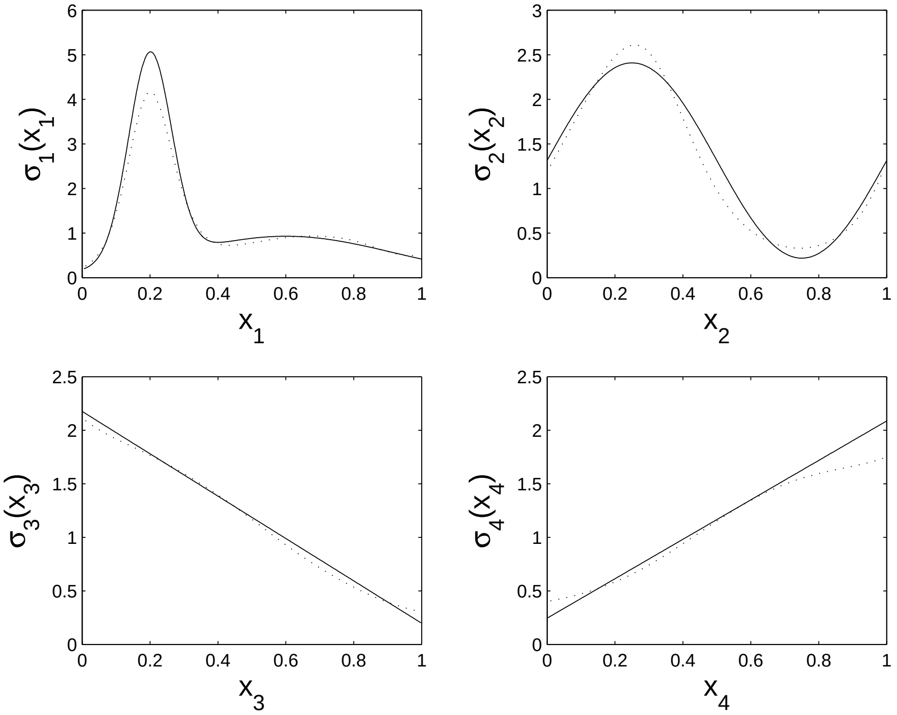


Figure 4.3: The true and estimated squared root variance functions for the additive model

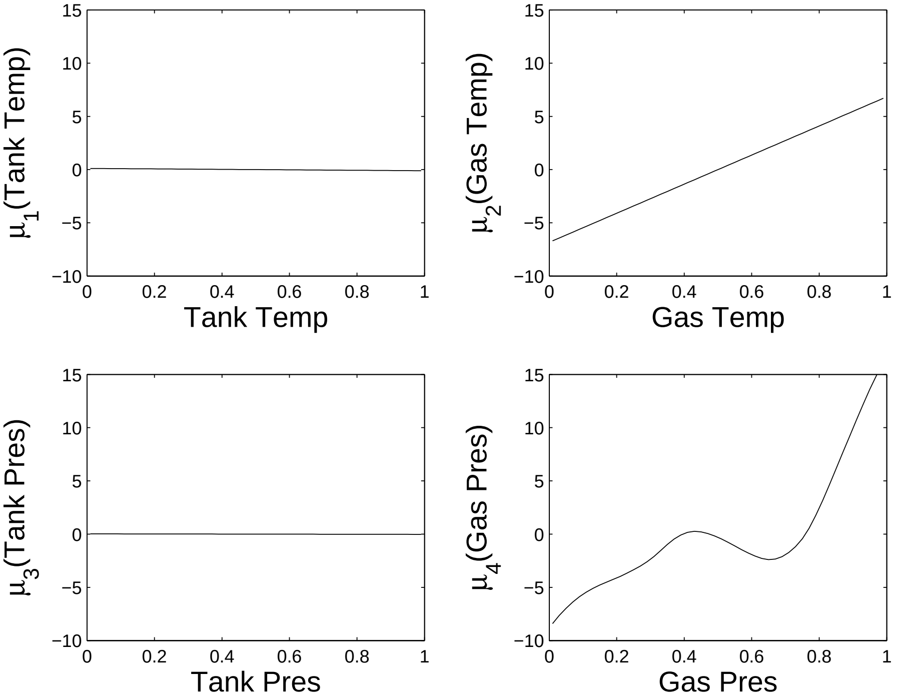


Figure 4.4: The estimated mean functions for the sniffer data

MATLAB and run on a Pentium IV 2.4 GHz PC. We have experimented also with longer runs attaining near identical fits to those described above.

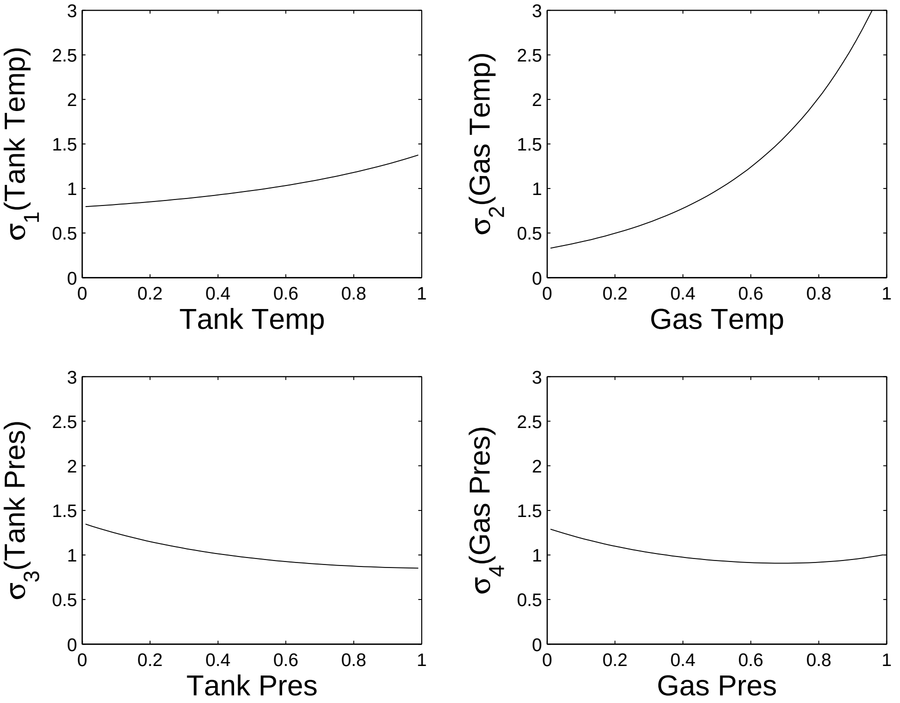


Figure 4.5: The estimated square root variance functions for the sniffer data

4.5 Normalization of cDNA microarrays

Microarray technology allows measurement of gene expression in tissue samples for thousands of genes simultaneously – see Nguyen et al. (2002) for an overview of various microarray technologies from a statistical point of view. The idea behind microarray technology is to measure differential abundance of mRNA as a way of measuring the differential abundance of proteins. In a cDNA microarray relative mRNA abundance is compared for two tissue samples. The mRNA from each of the samples is reverse transcribed back into cDNA or complementary DNA. Each of the cDNA samples is labelled with a different flourescent dye and the samples are hybridized to a glass slide upon which DNA sequences for different genes have been printed (one spot on the slide for each gene). The labelled cDNA sequences bind during hybridization to their complementary sequences on the slide and then when we scan the slide using light of an appropriate frequency for the dye for each sample we obtain a flourescence measurement for each spot indicating how much cDNA has bound to that spot for the given sample. The measurement is proportional to mRNA abundance. The end result after image analysis is that we have for each gene a "red" and "green" channel measurement, one for each of the tissue samples. The ratio of these values for a given gene can tell us about the differential expression of the gene in the two samples. However, it is found that there is systematic variation in the expression ratios not related to genetic effects but related to the intensity at each spot and the properties of the dyes. Another potential source of variation relates to the order in which the DNA spots were printed on the slide. Spots are also printed on the array in blocks, with different blocks being printed by the same print tip in the robotic printer used in making the slides – different blocks or print tips can behave differently.

Systematic variations in the expression ratios (which are usually analyzed on a log scale) need to be removed before further data analysis in a process called normalization (see Smyth and Speed

(2003) for an overview of normalization for cDNA microarrays). One approach to normalization involves fitting a flexible curve for the mean log relative expression values (called M values) as a function of the average spot intensity (average of log single channel values). This curve fitting is usually done separately for each print tip on each array in the experiment. The fitted values are subtracted from the raw M values to obtain normalized M-values used in subsequent data analysis. Further scale normalization may also be done. Here we consider using our methodology for mean and variance modelling for normalization of cDNA microarrays. For illustrative purposes we just consider one print tip block from an array that was part of a series of experiments investigating genetic control of gene transcription (see Cotsapas et al. (2003), for a description of the experiment). We fit an additive model to the raw M-values for both the mean and log variance with flexible terms for both the intensity A and the putdown time of the spot. Plots of M against the predictors are shown in figure 4.6 and the fitted terms in the additive model for the mean and log variance are shown in figure 4.7. All predictors have estimated posterior probability 1 of inclusion in both the mean and log variance models. The mean M-value varies in a complex way as a function of intensity, and there is also evidence of the variance changing as a function of both intensity and put down time. Estimating the variance may have a robustifying effect on smoothing, since groups of differentially expressed genes with large magnitude M-values may be downweighted in the smoothing: see, for instance, the peak in our estimated function for intensity in the variance function at around 0.4 and compare this with the plot of raw M values against intensity. The variance estimates can also give insight into the complex sources of variation which may be present in cDNA microarray experiments. These results were obtained based on 1000 sampling iterations with 200 iterations burn in for our sampling scheme. Longer runs gave similar results. Run time for the sampling scheme in this example for 1200 iterations was 285 seconds using a program written in MATLAB and run

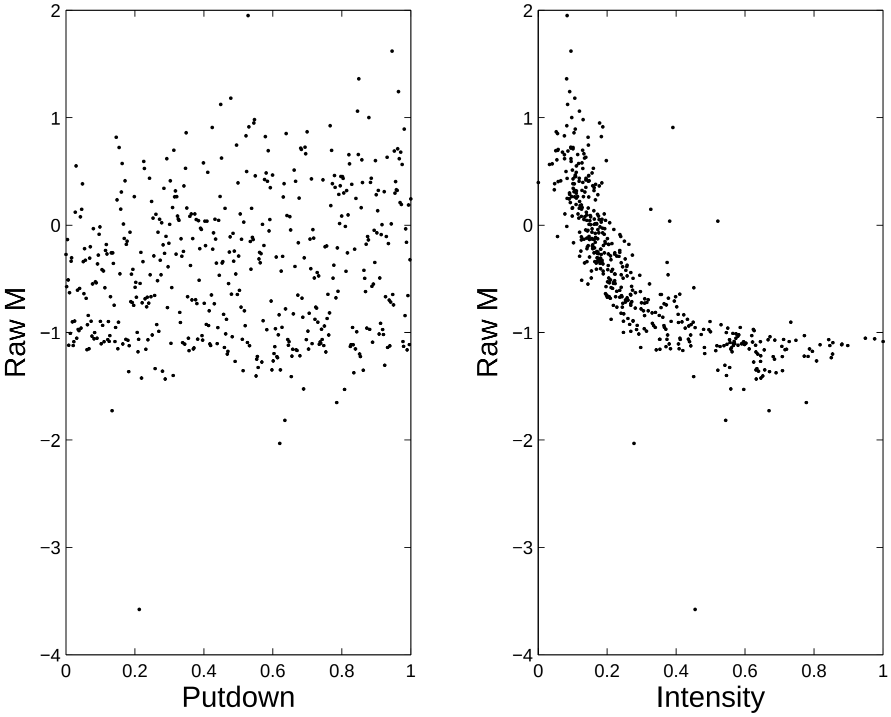


Figure 4.6: Plot of raw M-values against intensity and putdown time

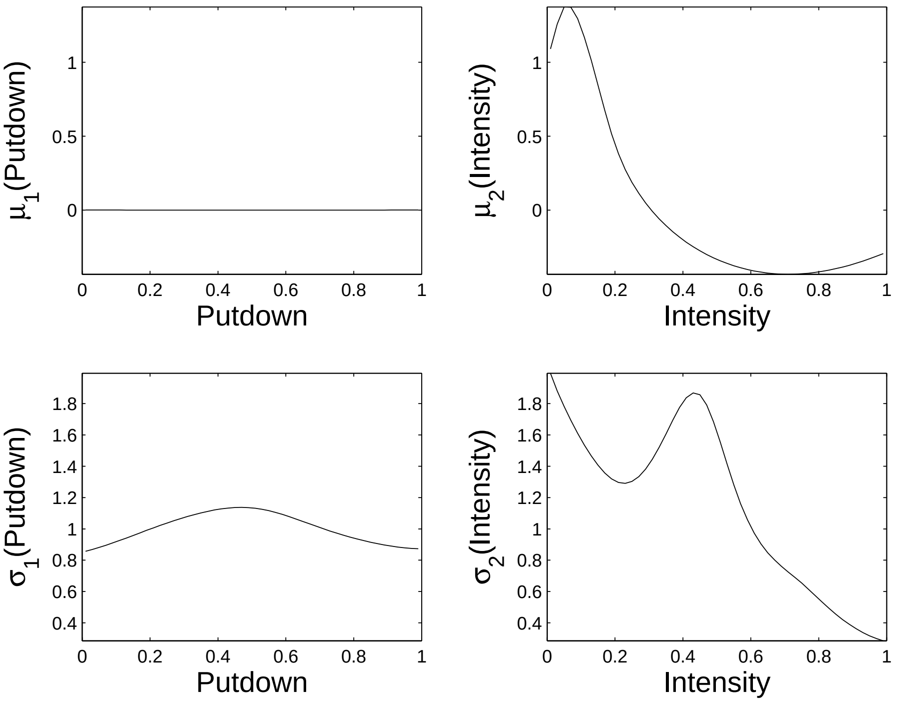


Figure 4.7: Estimated mean and square root variance functions for the microarray data

on a Pentium IV 2.4 GHz PC. Longer runs of the sampling scheme give near identical results.

5 Discussion

This article considers a Gaussian regression model with both the mean and the variance modeled as functions of the explanatory variables. Modeling the variance is important for a number of reasons. First, to obtain accurate prediction intervals for future observations. Second, to understand how the variance function depends on the covariates. Third, to estimate the mean function more efficiently.

Our article makes the following contributions. First, a Bayesian model is proposed for both the mean and the variance functions which allows variable selection on the explanatory variables. Flexible effects for covariates can be estimated adaptively using a linear combination of basis functions approach. Second, a simulation method is developed to ensure that the sampling scheme converges quickly and mixes well.

Appendix: Generating Θ

This appendix presents the technical details of how Θ is generated in sampling scheme 1. To derive the proposal densities for steps 1 to 5 of sampling scheme 1, it is necessary to obtain the expression for the likelihood with α integrated out. This expression is given by

$$\begin{aligned} p(y \mid J, K, \sigma^2, \beta, c_\alpha, c_\beta) &= \int p(y \mid J, K, \sigma^2, \beta, \alpha, c_\alpha, c_\beta) p(\alpha \mid \sigma^2, \beta, J) d\,\alpha \\ &\propto \left|\sigma^2 D(\beta)^2\right|^{-\frac{1}{2}} (1+c_\alpha)^{-\frac{q(J)}{2}} \exp\left\{-\frac{1}{2\sigma^2} S(J,y)\right\}, \qquad \text{(A.1)} \end{aligned}$$

where $S(J,y) = \widetilde{y}^T\widetilde{y} - \frac{c_\alpha}{1+c_\alpha}\widetilde{y}^T\widetilde{X}_J(\widetilde{X}_J^T\widetilde{X}_J)^{-1}\widetilde{X}_J^T\widetilde{y}$.

Generating J

First, the order of the elements of J is randomly permuted. For notational convenience, we retain the symbol J for this permuted vector. Next, the elements of J are generated in blocks, with the block size chosen at random to be of size $2, 4$ or 6. Let J_B be one such block. Then J_B is generated from its prior density with the rest of the elements of J kept at their current values. That is, the proposal density for J is

$$Q\left(J^c \to J^p | B\right) = \Pr\left(J_B^p | J_{-B}^c\right) I(J_{-B}^p = J_{-B}^c), \qquad \text{(A.2)}$$

where $I(\cdot)$ is the indicator function. A computable expression for $\Pr\left(J_B^p \mid J_{-B}^c\right)$ is given by Kohn et al. (2001). The Metropolis-Hastings acceptance probability for the proposal density (A.2) is

$$\begin{aligned} \alpha\left(J^c \to J^p\right) &= \min\left\{\frac{p(J_B^p \mid J_{-B}, K, \sigma^2, \beta, c_\beta, y)}{p(J_B^c \mid J_{-B}, K, \sigma^2, \beta, c_\beta, y)} \times \frac{q(J^p \to J^c)}{q(J^c \to J^p)}\right\} \\ &= \min\left\{\frac{(1+c_\alpha)^{-\frac{q^p}{2}} \exp\left\{-\frac{1}{2\sigma^2}S(J^p,y)\right\}}{(1+c_\alpha)^{-\frac{q^c}{2}} \exp\left\{-\frac{1}{2\sigma^2}S(J^c,y)\right\}}\right\}, \end{aligned}$$

where q^p and q^c are the numbers of active elements in J^p and J^c respectively.

Generating the scale factor c_α

The scale factor c_α has the conditional density

$$p(c_\alpha \mid J, K, \sigma^2, \beta, c_\beta, y) \propto (1+c_\alpha)^{-\frac{q(J)}{2}} \exp\left\{-\frac{1}{2\sigma^2}S(J,y)\right\} p(c_\alpha), \qquad \text{(A.3)}$$

and c_α is generated from a proposal density because it is difficult to generate it from its conditional density. We take the proposal density to be a Gaussian approximation to the conditional density of c_α, with the Gaussian approximation centered at the mode of the conditional density. Let $f(c_\alpha) = \log p(c_\alpha \mid J, K, \sigma^2, \beta, c_\beta, y)$. Then the proposal density for c_α is $\mathrm{N}(c_\alpha; \widehat{c}_\alpha, \delta)$, where $\widehat{c}_\alpha$ is the mode of $f(c_\alpha)$ and is obtained using a Newton-Raphson routine, and $\delta = -1/f''(\widehat{c}_\alpha)$. The Metropolis-Hastings acceptance probability is

$$\alpha\left\{c_\alpha^c \to c_\alpha^p\right\} = \min\left\{\frac{\exp(f(c_\alpha^p))}{\exp(f(c_\alpha^c))} \times \frac{\exp\{-\frac{1}{2\delta}(c_\alpha^c - \widehat{c}_\alpha)^2\}}{\exp\{-\frac{1}{2\delta}(c_\alpha^p - \widehat{c}_\alpha)^2\}}\right\}$$

Generating σ^2

It is straightforward to show that the conditional density of σ^2 is inverse gamma, with parameters $a_\sigma + \frac{n}{2}$ and $b_\sigma + \frac{1}{2}S(J,y)$, i.e.,

$$\begin{aligned} p(\sigma^2 \mid J, K, \beta, c_\alpha, c_\beta, y) &\propto p(y \mid J, K, \sigma^2, \beta, c_\alpha, c_\beta) p(\sigma^2) \\ &\propto (\sigma^2)^{-(1+a_\sigma+\frac{n}{2})} \exp\left\{ -\frac{1}{\sigma^2} \left(\frac{1}{2} S(J,y) + b_\sigma \right) \right\} \end{aligned} \tag{A.4}$$

Generating K and β

We next show how to generate K and β. First, the order of the elements of K is permuted at random. For notational convenience, we still retain the symbol K for this permuted value. Next, the elements of K are generated in blocks, with the block size chosen at random to be $2, 4$ or 6. Let K_B be one such block. Then K_B is generated simultaneously with β, as follows. First, K_B is generated from its prior density similarly to the way that J_B was generated. That is, the proposal density for K is

$$Q\left(K^c \to K^p \mid B\right) = \Pr\left(K_B^p \mid K_{-B}^c\right) I(K_{-B}^p = K_{-B}^c),$$

Next, β_{K^p} is generated from a proposal density because it is intractable to generate it from its conditional density.

Let $\hat{\alpha}^c = c_\alpha/(1+c_\alpha)(\tilde{X}_J^T \tilde{X}_J)^{-1} \tilde{X}_J^T \tilde{y}$ denote the current value for $E(\alpha_J | \theta^c \backslash \alpha^c, y)$. Let e^c denote the n-vector with

$$e_i^c = (y_i - x_i^T \hat{\alpha}^c)^2,$$

$i = 1, ..., n$. The vector e^c can be thought of as a vector of squared residuals. These squared residuals will have a distribution approximately $\sigma_i^2 \chi_1^2$, where $\sigma_i^2 = \sigma^2 \exp(z_i^T \beta)$. That is, the e_i^c follow approximately a gamma distribution with mean σ_i^2 and $\log \sigma_i^2 = \log \sigma^2 + z_i^T \beta$. This is a gamma generalized linear model with an offset. For a given value of K, we construct a proposal distribution for β_K based on one step of an iteratively reweighted least squares algorithm, an idea suggested by Gamerman (1997) and extended to model selection problems by Nott and Leonte (2004). To be precise, write σ_i^{2c}, β^c for the current values of σ_i^2, β,

$$\eta_i^c = z_i^T \beta^c + \frac{e_i^c - \sigma_i^{2c}}{\sigma_i^{2c}}, \ \eta^c = (\eta_1^c, ..., \eta_n^c)^T,$$

$$\Delta_{K^p} = \left(\frac{1}{c_\beta} I + \frac{1}{2} Z_{K^p}^T Z_{K^p} \right)^{-1}$$

and

$$\hat{\beta}_{K^p} = (Z_{K^c}^T Z_{K^c})^{-1} Z_{K^c}^T \eta^c$$

and then generate a proposal value β^p with $\beta_{K^p}^p$ multivariate T with four degrees of freedom, $T_4(\hat{\beta}_{K^p}, \Delta_{K^p})$ and remaining elements zero. Write $q(\beta_{K^p} | \hat{\beta}_{K^p}, \Delta_{K^p})$ for this proposal density. Write $\hat{\beta}_{K^c}$, Δ_{K^c} for the proposal mean and scale matrix for $\hat{\beta}_{K^c}$ when taking a step in the reverse direction going from model K^p to model K^c, and $q(\beta_{K^c} | \hat{\beta}_{K^c}, \Delta_{K^c})$ for the corresponding multivariate T proposal density with four degrees of freedom. Then our proposal value (K^p, β^p) for (K, β) is accepted with probability $\min\{1, \alpha\}$, where, for simplicity, we write J, σ^2, c_α and c_β for the current values of these parameters,

$$\begin{aligned} \alpha = & \frac{p(y|J, K^p, \sigma^2, \beta^p, c_\alpha, c_\beta)}{p(y|J, K^c, \sigma^2, \beta^c, c_\alpha, c_\beta)} \times \\ & \frac{(2\pi c_\beta)^{-q_p/2}}{(2\pi c_\beta)^{-q_c/2}} \frac{\exp\left(-\frac{1}{2c_\beta} \beta_{K^p}^T \beta_{K^p}\right)}{\exp\left(-\frac{1}{2c_\beta} \beta_{K^c}^T \beta_{K^c}\right)} \frac{q(\beta_{K^p} | \hat{\beta}_{K^p}, \Delta_{K^p})}{q(\beta_{K^c} | \hat{\beta}_{K^c}, \Delta_{K^c})}. \end{aligned}$$

Generating the scale parameter c_β

The parameter c_β is generated from its conditional density, which is an inverse gamma, with

$$\begin{aligned} p(c_\beta \mid J, K, \beta, c_\alpha, y) &\propto p(\beta \mid K, c_\beta) p(c_\beta) \\ &= (c_\beta)^{-(1+a_c+\frac{q(K)}{2})} \exp\left\{-\frac{1}{c_\beta}\left(\frac{\beta_{K^p}^T \beta_{K^p}}{2} + b_c\right)\right\} \end{aligned} \qquad \text{(A.5)}$$

Acknowledgments

The research in the article was supported by an Australian Research Council grant on mixture models. We would like to thank an associate editor and two referees whose comments improved the presentation of the paper. We thank Eva Chan, Chris Cotsapas, Peter Little and Rohan Williams for supplying the microarray dataset and answering numerous questions.

References

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

Biller, C. and Fahrmeir, L. (2001), "Bayesian varying-coefficient models using adaptive regression splines," *Statistical Modelling*, 1, 195–211.

Carroll, R. J. (1982), "Adapting for heteroscedasticity in linear models," *Ann. Statist*, 10, 1224–1233.

Carroll, R. J. and Ruppert, D. (1988), *Transformation and Weighting in Regression*, London, Chapman and Hall.

Cotsapas, C., Chan, E., Kirk, M., Tanaka, M., and Little, P. (2003), "Genetic variation in the control of transcription," *Cold Spring Harbor Symposia on Quantitative Biology*, 68, 109–114.

Cripps, E., Kohn, R., and Nott, D. (2006), "Bayesian subset selection and model averaging using a centred and dispersed prior for the error variance," *To appear in Australian and New Zealand Journal of Statistics.*

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

Dette, H., Munk, A., and Wagner, T. (1998), "Estimating the variance in nonparametric regression - what is a reasonable choice?" *Journal of the Royal Statistical Society, B*, 60, 751–764.

Gamerman, D. (1997), "Sampling from the posterior distribution in generalized linear mixed models," *Statistics and Computing*, 7, 57–68.

Holmes, C. C. and Mallick, B. K. (1998), "Radial basis functions of variable dimension," *Neural Computation*, 10, 1217–1233.

Kaufman, L. and Rousseeuw, P. J. (1990), *Finding Groups in Data*, New York: John Wiley & Sons.

Kohn, R., Smith, M., and Chan, D. (2001), "Nonparametric regression using linear combinations of basis functions," *Statistics and Computing*, 11, 301–310.

Liu, J. (2001), *Monte Carlo strategies in scientific computing*, New York, Springer.

Müller, H. G. and Stadtmüller, U. (1987), "Estimation of heteroscedasticity in regression analysis," *Annals of Statistics*, 15, 610–625.

Nguyen, D., Arpat, A., Wang, N., and Carroll, R. (2002), "DNA microarray experiments: Biological and technological aspects," *Biometrics*, 58, 701–717.

Nott, D. and Leonte, D. (2004), "Sampling schemes for Bayesian variable selection in generalized linear models," *Journal of Computational and Graphical Statistics*, 13, 362–382.

Ruppert, D., Wand, M. P., Holst, U., and Hössjer, O. (1997), "Local polynomial variance-function estimation," *Technometrics*, 39, 262–273.

Seber, G. (1977), *Linear Regression Analysis*, New York, Wiley.

Smith, M. and Kohn, R. (1996), "Nonparametric regression using Bayesian variable selection," *Journal of Econometrics*, 75, 317–344.

Smyth, G. (1989), "Generalized linear models with varying dispersion," *Journal of the Royal Statistical Society, B*, 51, 47–60.

Smyth, G. and Speed, T. (2003), "Normalization of DNA microarray data," *Methods: Selecting candidate genes from DNA array screens: Applications to Neuroscience. D. Carter (ed.)*, 31, 265–273.

Weisberg, S. (1985), *Applied Regression (Second Edition)*, New York: John Wiley & Sons.

Yau, P. and Kohn, R. (2003), "Estimation and variable selection in nonparametric heteroscedastic models," *Statistics and Computing*, 13, 191–208.