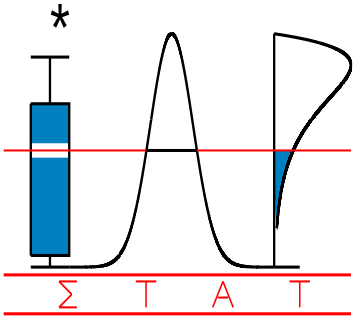
T E C H N I C A L

R E P O R T

0618

BAYESIAN MULTI-DIMENSIONAL DENSITY ESTIMATION WITH P-SPLINES

LAMBERT P., and P. EILERS



I A P S T A T I S T I C S

N E T W O R K

INTERUNIVERSITY ATTRACTION POLE

http://www.stat.ucl.ac.be/IAP

Bayesian multi-dimensional density estimation with P-splines

Philippe Lambert[1,2*] & Paul H.C. Eilers[3]

[1] *Institut de statistique, Université catholique de Louvain, Louvain-la-Neuve, Belgium*

[2] *Unité d'épidémiologie, biostatistique et méthodes opérationnelles, Faculté de Médecine, Université catholique de Louvain, Belgium.*

[3] *Department of Medical Statistics, Leiden University, Leiden, The Netherlands*

May 22, 2006

Abstract

Polytomous logistic regression combined with spline smoothing gives a powerful tool for Bayesian density estimation. Using fast array algorithms, multiple dimensions can be handled in a fast and uniform way. The Langevin-Hastings algorithm allows efficient sampling from the associated (reparameterized) posterior distribution. Illustrations of density estimation are provided, as well as a new approach to smooth quantile regression.

Key words: histogram; polytomous logistic regression; P-splines; Langevin-Hastings algorithm; quantile regression.

1 Introduction

Density estimation is an almost neglected topic in modern Bayesian statistics. Heavy computation with sophisticated algorithms is the standard, but results are mostly presented as kernel-smoothed (one-dimensional) distributions, using default settings, or just as histograms. In the Bayesian literature, density estimation has been treated as a step-child; it is not considered explicitly in standard reference books. Our goal here is to fill in the gaps, using a combination of polytomous logistic regression, penalized splines and efficient simulation.

Undoubtedly, BUGS is the most popular package for doing Bayesian statistics. One can present results as scatterplots or as kernel-smoothed densities. The

*Correspondence to: Philippe Lambert, Université catholique de Louvain, Institut de statistique, Voie du Roman Pays 20, B-1348 Louvain-la-Neuve (Belgium). E-mail: lambert@stat.ucl.ac.be Phone: +32-10-47.28.01 Fax: +32-10-47.30.32

bandwidth of the smoothing kernel can be tuned, but this is a subjective process. There is hardly any discussion of this issue in the BUGS documentation. The package CODA, which is available for R and S-PLUS, provides functions for post-processing of BUGS output. It uses the S function `density()` for kernel smoother, providing several automatic plug-in type of choices for the bandwidth.

Perhaps the only advantage of kernel smoothing is that it is easy to explain: replace the individual observations by little humps and add them up. This is not a good recipe for computation with large data sets. Several authors have proposed algorithms that first count observations in (narrow) bins. Each bin then is replaced by a hump, scaled by the number of observations in the bin. When many bins are being used, the effort can be decreased dramatically with the Fast Fourier Transform (FFT) (Silverman, 1986). In principle the FFT scheme can be extended to two dimensions or more, but this seems to be applied seldomly. A practical alternative is the use of a discrete smoother with a positive impulse response (Eilers and Goeman, 2004).

With appropriate algorithms, computation of kernel smoothers does not have to be a real obstacle, but several other properties are undesirable. The kernel smoother always increases spread, i.e. the variance of the density estimate will always be larger than that of the raw data. There is also a problem at intrinsic domain boundaries. If a variable can only be positive, spreading by the kernel will produce non-zero density estimates on the negative axis, which looks sloppy. Positive variables often show densities with a peak near or at the origin. It can become severely rounded by a kernel smoother. Ad-hoc solutions exist, like mirroring the data, or designing specialized boundary kernels, but they are not very successful.

Optimization of the kernel bandwidth can be done by cross-validation (CV). Using, say, 10-fold, CV, this is not a real hurdle, but leave-one-out CV schemes are unattractive. From a Bayesian perspective, it is far from clear how to state a proper model to represent kernel smoothing.

Many researchers have approached Bayesian density estimation as the estimation of normal mixtures. A seminal paper was the one by Roeder and Wasserman (1997). They considered one-dimensional smoothing, but the concept can be generalized directly to higher dimensions ("model based clustering"). When doing McMC simulations, each individual observation has to be connected to each component of the mixture. For large data sets one should also introduce some way of binning of the data, to keep the computational effort in reasonable bounds. The model is simple, but special care may be needed, to maintain proper labeling of the components of the mixtures (Stephens, 2000). Additional complications arise when one allow the number of components to change during the simulations. Normal mixtures have problems to respect domain boundaries.

Müller and Vidakovic (1998) model the square root of a density as a sum of wavelets, implicitly computing a narrow-bin histogram. They do not mention the size of the wavelet basis, but presumably it is relatively large. Simulation is done

with a Metropolis sampler.

Heikkinen and Arjas (1998) consider spatial Poisson intensity estimation, which is of course equivalent to two-dimensional density estimation, as they indicate in their discussion. They construct a Voronoi tesselation of the plane, determined by the positions of the observations. Within each Voronoi tile the intensity is assumed to be constant. A difference prior, on the logs of the intensities of neighboring tiles, models the assumption of smoothness. Simulation is done by a complex McMC scheme. For large data sets this method will not work, because of the complexity of computing the tesselation. Also the number of parameters increases linearly with the number of observations. Extension to higher dimensions is far from straightforward.

Hansen and Kooperberg (2002) discuss Bayesian estimation of the Logspline model. They model the logarithm of a density as a sum of natural cubic regression splines (in one dimension) or as a sum of linear triangular patches (in two dimensions). The number and the positions (chosen from those of the observations) are the parameters of the model, and specialized priors are introduced. In addition there is a roughness penalty, based on second derivatives. The weight of this penalty is set to a fixed number.

We model the logarithm of the density as a sum of scaled B-splines. This is similar to work by Kooperberg and Stone (1991). However, we do not try to optimize the number and positions of the knots that define the B-splines. Instead, we start out with many equally spaced knots, defining a basis that is "too rich": it provides more flexibility than needed. To get the desired smoothness, we put a difference penalty on the coefficients and the weight of this penalty is tuned to the data. This is the P-spline approach, advocated by Eilers and Marx (1996) and inspired by the work of O'Sullivan (1988).

In a Bayesian setting the penalty is the logarithm of a prior density of the differences of the coefficients. Efficient simulation is possible with the Langevin-Hastings algorithm (Roberts and Tweedie, 1996) and proper rotation of the parameter vector. This approach has shown its value in hazard estimation in survival models with varying coefficients (Lambert and Eilers, 2005). In some sense one-dimensional density estimation is simplified hazard estimation (without covariates), so we could stop here. However, we extend our approach to multidimensional density estimation, making use of recently developed fast methods for weighted regression on tensor product basis functions when the data are positioned on grids (Eilers *et al.*, 2006; Currie *et al.*, 2006).

The spline coefficients are found by penalized polytomous logistic regression applied to the counts in a histogram with narrow bins. Polytomous logistic regression can be extended to multidimensional histograms, using tensor products of B-splines as basis functions. The difference penalties can be extended to multiple dimensions too, so P-splines can be generalized to this setting. However, straightforwardly constructing the basis matrix and performing the weighted regressions leads to problems, in memory use and in computation time. Eilers *et al.*

(2006) present an algorithm in which the multidimensional basis matrix is avoided completely. The computations are rearranged in such a way that one works along each dimension separately. This saves orders of magnitude in memory use and computation time. The algorithm has been used successfully for smoothing (and extrapolation) of large mortality tables (Currie *et al.*, 2004b). The underlying array algorithms are presented more formally by Currie *et al.* (2006).

The plan of the paper is as follows. In the next section we present univariate density smoothing, introducing the histogram approach, logistic regression on penalized splines, the corresponding Bayesian setting and the choice of prior. Section 3 extends our approach to two-dimensional smoothing. A new element is anisotropic smoothing, with different amounts of smoothing along the two dimensions. In this section we already use fast algorithms that avoids explicit computations with tensor product bases. Matrix operations are sufficient there, but in higher dimensions one has to switch to specific array algorithms, which are described in Section 4. Efficient sampling is the key to success in the application of Bayesian models. In Section 5 we present an adaptation of the Langevin-Hastings algorithm, with automatic tuning and rotation of the parameter vector. Section 6 presents several applications in one and two dimensions. It also contains some details on how we implemented the algorithm, using the R system. Of special interest is a new approach to quantile smoothing. The paper ends with a short Discussion.

2 Univariate density smoothing

Assume that a random sample $\{y_j, j = 1 \ldots, n\}$ of a random variable Y has been observed and that an estimation of the density f_Y of Y is of interest.

2.1 Histogram

Following Eilers and Marx (1996), we propose to tackle the problem by starting from the histogram associated to a large number I of bins with equally spaced limits. This requires the specification of a compact interval $[y_{\min}, y_{\max}]$ over which most of the probability mass is expected to be found. Let x_i denote the center of the ith bin and n_i be the number of observations in that bin of width Δ. Then, it is well known that

$$(N_1, \ldots, N_I) \sim \text{Mult}(n; \pi_1, \ldots, \pi_I) \quad \text{where} \quad \pi_i = \int_{x_i - \Delta/2}^{x_i + \Delta/2} f(z)\, dz \approx f(x_i)\Delta \; .$$

where Mult stands for multinomial distribution.

Consider a basis $\{b_k(\cdot) : k = 1, \ldots, K\}$ of cubic B-splines associated to equidistant knots on $[y_{\min}, y_{\max}]$. If $(B)_{ik} = b_{ik} = b_k(x_i)$ denotes the $I \times K$ matrix giving

the basis functions evaluated at the bin midpoints, then a possible model for $\pi = (\pi_1, \ldots, \pi_I)'$ is the polytomous logistic regression

$$\log\left(\frac{\pi_i}{\pi_1}\right) = \eta_i = \sum_k b_{ik}\phi_k \quad (i > 1)$$

where the 1st bin is the (arbitrary) reference category and the ϕ_k are regression coefficients. It corresponds to

$$\pi_i \;=\; \frac{\mathrm{e}^{\eta_i}}{\mathrm{e}^{\eta_1} + \mathrm{e}^{\eta_2} + \ldots + \mathrm{e}^{\eta_I}} \tag{1}$$

with $\eta_1 = 0$.

Equivalently, one could consider Equation (1) with

$$\eta_i = \sum_k b_{ik}\phi_k \quad \forall\ i$$

Then, for identifiability reasons, one should constrain the regression parameters $\boldsymbol{\phi}' = (\phi_1, \ldots, \phi_K)$ as $\pi_i(\boldsymbol{\phi}) = \pi_i(\boldsymbol{\phi} + a)$ for any real a. This was done in the first specification by requiring $\eta_1 = 0$. Below, we shall require that $\sum_k \phi_k = 0$. The corresponding log-likelihood, with the identifiability constraint, is

$$\log L(\boldsymbol{\phi}|\boldsymbol{n}) = \sum_i n_i \log \pi_i \tag{2}$$

with gradient

$$\frac{\partial l}{\partial \boldsymbol{\phi}} = B'(\boldsymbol{n} - \boldsymbol{\mu}) \tag{3}$$

where $\boldsymbol{\mu} = n\boldsymbol{\pi}$. Note that the gradient of the log-likelihood or of the log-posterior will be systematically provided as it is required in the proposed inference process (see Section 5).

2.2 Roughness penalty

A large number of knots (say 20) is recommended to give enough flexibility in the approximation; however, that flexibility should be counterbalanced by a roughness penalty to give a smooth estimate of the density.

In a Bayesian setting, a roughness penalty translates into a prior distribution on the splines coefficients. More specifically, we shall translate the frequentist proposal made by Eilers and Marx (1996) by penalising the rth order differences of the successive B-splines coefficients by assuming that

$$\Delta^r \phi_k \sim N(0, \tau_q^{-1})$$

That idea was already successfully used in several papers in various contexts (see e.g. Lambert and Eilers (2005) in survival analysis, Lambert (2005) in copula estimation, Berry *et al.* 2002 in normal regression models and Lang and Brezger 2004 in additive models to cite a few). Consequently, we propose to multiply the prior for the B-splines coefficients by

$$\tau^{\mathcal{R}(P)/2} \exp\left\{-\frac{1}{2}\ \tau\ \boldsymbol{\phi}' P \boldsymbol{\phi}\right\}.$$

where $\mathcal{R}(P)$ denotes the rank of P and $P = D'D$ is the matrix such that

$$\sum_k (\Delta^r \phi_r)^2 = \boldsymbol{\phi}' P \boldsymbol{\phi}$$

For example, with $r = 2$, one has

$$D = \begin{bmatrix} 1 & -2 & 1 & 0 & \dots & 0 \\ 0 & 1 & -2 & 1 & \dots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 1 & -2 & 1 \end{bmatrix}.$$

and $\mathcal{R}(P) = K - 2$. A gamma prior $\mathcal{G}(a,\ b)$ with a large variance (as obtained by taking $a = b = .0001$, say) is usually advocated (Lang and Brezger, 2004) to express our prior ignorance about suitable values for τ. However, this cannot be true in specific circumstances (Jullion and Lambert, 2005). Alternatives will be presented in Section 2.4.

With the preceding gamma prior, the log of the joint posterior is

$$\begin{aligned} \log p(\boldsymbol{\phi}, \tau|\boldsymbol{n}) &= \sum_i n_i \log \pi_i + \frac{1}{2}\ \mathcal{R}(P) \log \tau - \frac{1}{2}\ \tau\ \boldsymbol{\phi}' P \boldsymbol{\phi} \\ &\quad + (a-1)\log\tau - b\tau \end{aligned} \tag{4}$$

with associated gradient

$$\frac{\partial \log p(\boldsymbol{\phi}, \tau|\boldsymbol{n})}{\partial \boldsymbol{\phi}} = B'(\boldsymbol{n} - \boldsymbol{\mu}) - \tau\ P\boldsymbol{\phi} \tag{5}$$

$$\frac{\partial \log p(\boldsymbol{\phi}, \tau|\boldsymbol{n})}{\partial \tau} = \frac{\mathcal{R}(P)}{2\tau} - \frac{1}{2}\ \boldsymbol{\phi}' P \boldsymbol{\phi} + \frac{a-1}{\tau} - b \tag{6}$$

2.3 Marginal posterior

The penalty parameter τ can be integrated out (Lambert, 2005) yielding the marginal log-posterior

$$\log p(\boldsymbol{\phi}|\boldsymbol{n}) = \sum_i n_i \log \pi_i - [a + 0.5\mathcal{R}(P)] \log\left(b + 0.5\ \boldsymbol{\phi}' P \boldsymbol{\phi}\right) \tag{7}$$

with gradient

$$\frac{\partial \log p(\boldsymbol{\phi}|\boldsymbol{n})}{\partial \boldsymbol{\phi}} = B'(\boldsymbol{n} - \boldsymbol{\mu}) - \frac{a + 0.5\mathcal{R}(P)}{b + 0.5\ \boldsymbol{\phi}' P \boldsymbol{\phi}}\ P\boldsymbol{\phi} \tag{8}$$

2.4 Robust penalty prior

In specific circumstances, the variance of the gamma prior for the roughness penalty parameter τ can strongly influence the smooth of the fitted curve (Jullion and Lambert, 2005). A robust mixture (see Bolstad, 2004, Chap. 14) for τ can be chosen instead (Lambert, 2005). Let $\mathcal{Q} = \{a_q = 10^{-q} : q = 1, \ldots, Q\}$ (say) be the set of values that we would like to evaluate for $a = b$ and denote by Q_q the qth prior model. A mixture prior for τ giving an equal prior weight to the Q possibilities is

$$(\tau|Q_q) \sim \mathcal{G}(a_q, b_q) \quad \text{with} \quad p(Q_q) = \frac{1}{Q}$$

where $p(Q_q)$ denotes the prior probability associated to the qth prior. The joint posterior distribution for $(\boldsymbol{\phi}, \tau, Q_q)$ is

$$p(\boldsymbol{\phi}, \tau, Q_q|\boldsymbol{n}) \propto L(\boldsymbol{\phi}|\boldsymbol{n})\ p(\boldsymbol{\phi}|\tau)\ p(\tau|Q_q)\ p(Q_q)$$

The conditional posterior distributions, useful to set up a Gibbs sampler, are

$$\begin{aligned} p(\boldsymbol{\phi}|\tau, Q_q; \boldsymbol{n}) \equiv p(\boldsymbol{\phi}|\tau; \boldsymbol{n}) &\propto L(\boldsymbol{\phi}|\boldsymbol{n})\ p(\boldsymbol{\phi}|\tau) \\ (\tau|\boldsymbol{\phi}, Q_q; \boldsymbol{n}) &\sim \mathcal{G}(a_q + 0.5\ \mathcal{R}(P), b_q + 0.5\ \boldsymbol{\phi}' P \boldsymbol{\phi}) \\ p(Q_q|\boldsymbol{\phi}, \tau; \boldsymbol{n}) \equiv p(Q_q|\tau; \boldsymbol{n}) &= \frac{p(\tau|Q_q)\ p(Q_q)}{\sum_l p(\tau|Q_l)\ p(Q_l)} \end{aligned}$$

A marginal posterior for $\boldsymbol{\phi}$ can be derived from the joint:

$$\begin{aligned} p(\boldsymbol{\phi}|\boldsymbol{n}) &= \sum_{q=1}^{Q} \int_0^{+\infty} p(\boldsymbol{\phi}, \tau, Q_q|\boldsymbol{n})\ d\tau \\ &\propto L(\boldsymbol{\phi}|\boldsymbol{n})\ \frac{1}{Q} \sum_{p=1}^{Q} A_q(\boldsymbol{\phi}) \end{aligned}$$

where

$$A_q(\boldsymbol{\phi}) = \frac{\Gamma(a_q + 0.5\ \mathcal{R}(P))\ b_q^{a_q}}{\Gamma(a_q)\ (b_q + 0.5\ \boldsymbol{\phi}' P \boldsymbol{\phi})^{a_q + 0.5\ \mathcal{R}(P)}}$$

The corresponding log-posterior is

$$\log p(\boldsymbol{\phi}|\boldsymbol{n}) = \sum_i n_i \log \pi_i + \log \left(\frac{1}{Q} \sum_{q=1}^{Q} A_q(\boldsymbol{\phi}) \right) \tag{9}$$

with gradient

$$\frac{\partial \log p(\boldsymbol{\phi}|\boldsymbol{n})}{\partial \boldsymbol{\phi}} = B'(\boldsymbol{n} - \boldsymbol{\mu}) - \frac{\frac{1}{Q} \sum_{q=1}^{Q} w_q(\boldsymbol{\phi}) A_q(\boldsymbol{\phi})}{\frac{1}{Q} \sum_{q=1}^{Q} A_q(\boldsymbol{\phi})}\ P\boldsymbol{\phi} \tag{10}$$

where

$$w_q(\boldsymbol{\phi}) = \frac{a_q + 0.5\ \mathcal{R}(P)}{b_q + 0.5\ \boldsymbol{\phi}' P \boldsymbol{\phi}}$$

3 Bivariate density smoothing

Assume that a random sample $\{(y_{1k}, y_{2k}) : k = 1, \ldots, n\}$ of a bivariate random variable (Y_1, Y_2) has been observed and that an estimation of the bivariate density $f_{Y_1Y_2}$ is of interest.

3.1 Histogram in 2D

The same ideas as in the 1D case can be used successfully. Consider for simplicity that most of the probability mass is within the rectangle $[y_1^{\min}, y_1^{\max}] \times [y_2^{\min}, y_2^{\max}]$. That region can be subdivided into a large number of cells. Again for simplity, assume that these cells are the rectangles corresponding to the partition of $[y_1^{\min}, y_1^{\max}]$ and $[y_2^{\min}, y_2^{\max}]$ into, respectively, I and J segments of constant width Δ_1 and Δ_2. If (x_{1i}, x_{2j}) and n_{ij} denote, respectively, the midpoint and the number of observations in cell (i, j), then

$$(N_{11}, \ldots, N_{IJ}) \sim \text{Mult}(n; \pi_{11}, \ldots, \pi_{IJ})$$

where

$$\pi_{ij} = \int\int_{\text{cell } (i,j)} f_{Y_1Y_2}(z_1, z_2)\ dz_1\ dz_2 \approx f_{Y_1Y_2}(x_{1i}, x_{2j})\Delta_1\Delta_2$$

Consider two bases of cubic B-splines $\{\breve{b}_k(\cdot) : k = 1, \ldots, K\}$ and $\{b_l(\cdot) : l = 1, \ldots, L\}$ associated with equidistant knots on $[y_1^{\min}, y_1^{\max}]$ and $[y_2^{\min}, y_2^{\max}]$. If $(\breve{B})_{ik} = \breve{b}_{ik} = \breve{b}_k(x_{1i})$ and $(B)_{jl} = b_{jl} = b_l(x_{2j})$ denote the $I \times K$ and $J \times L$ matrices associated to these bases at their respective bin midpoints, then a possible model for the $I \times J$ matrix of probabilities $(\Pi)_{ij} = \pi_{ij}$ is the polytomous logistic regression

$$\pi_{ij} \quad = \quad \frac{\mathrm{e}^{\eta_{ij}}}{\mathrm{e}^{\eta_{11}} + \ldots + \mathrm{e}^{\eta_{IJ}}} \tag{11}$$

where

$$\eta_{ij} = \sum_k \sum_l \breve{b}_{ik} b_{jl} \phi_{kl} = (\breve{B}\Phi B')_{ij}\ .$$

For identifiability reasons, one should constrain the $K \times L$ matrix of regression parameters $(\Phi)_{kl} = \phi_{kl}$ as $\pi_{ij}(\Phi) = \pi_{ij}(\Phi + a)$ for any real a. Below, we shall require that $\sum_k \sum_l \phi_{kl} = 0$.

The corresponding log-likelihood, with the identifiability constraint, is

$$\log L(\Phi|\boldsymbol{n}) = \sum_i n_{ij} \log \pi_{ij} = \sum \boldsymbol{N} \odot \log(\Pi) \tag{12}$$

with gradient

$$\frac{\partial l}{\partial \Phi} = \breve{B}'(\boldsymbol{N} - n\Pi)B \tag{13}$$

where $\boldsymbol{N}$ is the $I \times J$ matrix such that $(\boldsymbol{N})_{ij} = n_{ij}$ and $[\frac{\partial l}{\partial \Phi}]_{k,l} = \frac{\partial l}{\partial \phi_{kl}}$

3.2 Roughness prior and associated posterior

Consider the following notation for the K rows and L columns of Φ,

$$\Phi' = (\Phi_1^{r\prime}, \ldots, \Phi_K^{r}{}') \;\; ; \;\; \Phi = (\Phi_1^c, \ldots, \Phi_L^c)$$

respectively. We propose to force smoothness by considering a prior distribution on the rth order differences of the successive B-splines coefficients associated to each row and to each column of Φ:

$$\begin{aligned} p(\Phi_k^r|\tau_{r,k}) &\propto \exp\left\{-\frac{1}{2}\,\tau_{r,k}\;\Phi_k^r P_r \Phi_k^{r\prime}\right\} \\ p(\Phi_l^c|\tau_{c,l}) &\propto \exp\left\{-\frac{1}{2}\,\tau_{c,l}\;\Phi_l^{c\prime} P_c \Phi_l^c\right\} \end{aligned}$$

A different roughness penalty coefficient could be used for each row and for column, as suggested in the previous equation with, for example, $\tau_{r,k}$ standing for the penalty associated to the kth row. If this general case is of interest, then some smooth evolution should also be forced on these penalty coefficients (see Jullion and Lambert (2005) for spatially adaptive penalties). For most practical purposes, assuming that $\tau_{r,k} = \tau_r \;\forall k$ and $\tau_{c,l} = \tau_c \;\forall l$ provides enough flexibility.

Of course, these prior distributions cannot be specified independently as it concerns the same Φ parameters. Considering them jointly is essential to obtain the multiplicative constant that involve the penalty parameters. The contribution of these $K + L$ prior distributions to the posterior can be written as

$$\begin{aligned} &\left\{\prod_{k=1}^{K} p(\Phi_k^r|\tau_r)\right\}\left\{\prod_{l=1}^{L} p(\Phi_l^c|\tau_c)\right\} \propto \left\{\;\mathrm{e}^{-0.5\tau_r \sum_k \Phi_k^r P_r \Phi_k^{r\prime}}\right\}\left\{\;\mathrm{e}^{-0.5\tau_c \sum_l \Phi_l^{c\prime} P_c \Phi_l^c}\right\} \\ = \;& \exp\left\{-\frac{1}{2}\,\mathrm{vec}\,(\Phi)'\,(\tau_r\;P_r \otimes \mathcal{I}_K + \tau_c\;\mathcal{I}_L \otimes P_c)\;\mathrm{vec}\,(\Phi)\right\} \\ = \;& \exp\left\{-\frac{1}{2}\,\mathrm{vec}\,(\Phi)'\,P\;\mathrm{vec}\,(\Phi)\right\} \end{aligned}$$

where $\mathrm{vec}\,(\cdot)$ turns a matrix into a vector by stacking its columns, $\otimes$ is the Kronecker product and $\mathcal{I}_r$ is the identity matrix of size r. Thus, the prior on Φ is

$$p(\Phi|\tau_r, \tau_c) \;\propto\; \sqrt{d(P)}\exp\left\{-\frac{1}{2}\,\mathrm{vec}\,(\Phi)'\,P\;\mathrm{vec}\,(\Phi)\right\} \tag{14}$$

where $d(P)$ denotes the product of non-zero eigenvalues of P. It is a function of τ_r and τ_c. If one denotes the L and K eigenvalues of P_r and P_c by

$$\begin{aligned} \lambda_1^r \geq \ldots \geq \lambda_{\mathcal{R}(P_r)}^r > 0, \ldots, 0 \\ \lambda_1^c \geq \ldots \geq \lambda_{\mathcal{R}(P_c)}^c > 0, \ldots, 0 \end{aligned}$$

then one can show that the eigenvalues of P are

$$\lambda_{lk} = \tau_r \lambda_l^r + \tau_c \lambda_k^c \quad \text{where} \quad (l,k) \in \{1,\ldots,L\} \times \{1,\ldots,K\}$$

Hence, the number of nonzero eigenvalues of P is

$$\mathcal{R}(P) = KL - [L - \mathcal{R}(P_r)]\,[K - \mathcal{R}(P_c)]$$

As in the 1D case, many different priors can be used for the penalty coefficients. The simplest choice is obviously a gamma prior with a large variance, say $\tau_r \sim \mathcal{G}(a_r,\ b_r)$ and $\tau_r \sim \mathcal{G}(a_c,\ b_c)$ with $a_r = a_c = b_r = b_c = 10^{-4}$. The resulting log-posterior is then

$$\begin{aligned}\log p(\Phi, \tau_r, \tau_c|\boldsymbol{n}) &= \sum \boldsymbol{N} \odot \log(\Pi) \\ &+ \frac{1}{2} \sum_{l,k:\lambda_{lk}\neq 0} \log(\tau_r \lambda_l^r + \tau_c \lambda_k^c) - \frac{1}{2} \text{vec}\,(\Phi)'\, P\ \text{vec}\,(\Phi) \\ &+ (a_r - 1)\log\tau_r - b_r\tau_r + (a_c - 1)\log\tau_c - b_c\tau_c \end{aligned} \tag{15}$$

Note that, for computational purposes, it is more convenient to rewrite the kernel part of the prior:

$$\text{vec}\,(\Phi)'\, P\ \text{vec}\,(\Phi) = \tau_r \text{tr}\,(\Phi P_r \Phi') + \tau_c \text{tr}\,(\Phi' P_c \Phi) \tag{16}$$

The gradient of the log-posterior is

$$\begin{aligned}\frac{\partial \log p(\Phi, \tau_r, \tau_c|\boldsymbol{n})}{\partial \phi} &= \breve{B}'(\boldsymbol{N} - n\Pi)B - \tau_r \Phi P_r - \tau_c P_c \Phi \qquad (17)\\ \frac{\partial \log p(\Phi, \tau_r, \tau_c|\boldsymbol{n})}{\partial \tau_r} &= 0.5 \sum_{l,k:\lambda_{lk}\neq 0} \frac{\lambda_l^r}{\tau_r \lambda_l^r + \tau_c \lambda_k^c} - 0.5\ \text{tr}\,(\Phi P_r \Phi') + \frac{a_r - 1}{\tau_r} - b_r \\ \frac{\partial \log p(\Phi, \tau_r, \tau_c|\boldsymbol{n})}{\partial \tau_c} &= 0.5 \sum_{l,k:\lambda_{lk}\neq 0} \frac{\lambda_k^c}{\tau_r \lambda_l^r + \tau_c \lambda_k^c} - 0.5\ \text{tr}\,(\Phi' P_c \Phi) + \frac{a_c - 1}{\tau_c} - b_c \end{aligned}$$

3.3 Special case: $\tau_r = \tau_c$

An interesting special case is $\tau_r = \tau_c = \tau$ that assumes that the density smoothness is the same along both axes. In that case, simplifications appear in the roughness prior in Equation (14) as

$$d(P) \propto \tau^{\mathcal{R}(P)}$$

Hence, the log-posterior becomes

$$\begin{aligned}\log p(\Phi, \tau|\boldsymbol{n}) &= \sum \boldsymbol{N} \odot \log(\Pi) + \frac{\mathcal{R}(P)}{2}\log(\tau) \\ &- \frac{\tau}{2}[\text{tr}\,(\Phi P_r \Phi') + \text{tr}\,(\Phi' P_c \Phi)] + (a-1)\log\tau - b\tau \end{aligned} \tag{18}$$

with gradient

$$\begin{aligned}\frac{\partial \log p(\Phi,\tau|\boldsymbol{n})}{\partial \phi} &= \breve{B}'(\boldsymbol{N}-n\Pi)B-\tau(\Phi P_r+P_c\Phi) \\ \frac{\partial \log p(\Phi,\tau|\boldsymbol{n})}{\partial \tau} &= \frac{\mathcal{R}(P)}{2\tau}-\frac{1}{2}\left[\mathrm{tr}\left(\Phi P_r\Phi'\right)+\mathrm{tr}\left(\Phi' P_c\Phi\right)\right]+\frac{a-1}{\tau}-b\end{aligned} \tag{19}$$

3.4 Marginal posterior

No closed form can be obtained for the marginal posterior of Φ in the general case. However, when $\tau_r = \tau_c = \tau$, one can integrate out the roughness penalty parameter τ, yielding the marginal log-posterior

$$\begin{aligned}\log p(\Phi|\boldsymbol{n}) &= \sum \boldsymbol{N}\odot\log(\Pi) \\ &\quad -[a+0.5\mathcal{R}(P)]\log\left\{b+0.5[\mathrm{tr}\left(\Phi P_r\Phi'\right)+\mathrm{tr}\left(\Phi' P_c\Phi\right)]\right\}\end{aligned} \tag{20}$$

with gradient

$$\begin{aligned}\frac{\partial \log p(\Phi|\boldsymbol{n})}{\partial \phi} &= \breve{B}'(\boldsymbol{N}-n\Pi)B \\ &\quad -[a+0.5\mathcal{R}(P)]\,\frac{\Phi P_r+P_c\Phi}{b+0.5[\mathrm{tr}\left(\Phi P_r\Phi'\right)+\mathrm{tr}\left(\Phi' P_c\Phi\right)]}\end{aligned} \tag{21}$$

4 Extension to higher dimensions

The same ideas as in Sections 2 and 3 can be used successfully in higher dimensions. Of course, some care should be devoted to computational aspects to avoid memory problems. We were careful in the 2D case by computing the linear predictor using product of matrices, $(\boldsymbol{\eta})_{ij} = (\breve{B}\Phi B')_{ij}$, instead of the equivalent memory demanding expression based on Kronecker products, $(B\otimes\breve{B})\mathrm{vec}\,(\Phi)$. A generalization of this trick to higher dimensions was proposed in Eilers *et al.* (2006) with algorithmic details in Currie *et al.* (2004a, 2006).

If the number of dimensions is d, then the cell probabilities Π and the associated linear predictor $\boldsymbol{\eta}$ become d-dimensional arrays of size $i_1\times\ldots\times i_d$ and the B-splines basis the Kronecker product $B = B_d\otimes\ldots\otimes B_1$ (where B_r is $i_r\times K_r$) such that

$$\begin{aligned}\mathrm{vec}\,(\Pi)_i &= \frac{\exp[\mathrm{vec}\,(\boldsymbol{\eta})_i]}{\sum_r \exp[\mathrm{vec}\,(\boldsymbol{\eta})_r]} \\ \mathrm{vec}\,(\boldsymbol{\eta}) &= B\boldsymbol{\phi}\end{aligned} \tag{22}$$

where $\boldsymbol{\phi}$ is a vector of length $\prod_{r=1}^{d} K_r$. Again, an identifiability constraint like $\sum_k \boldsymbol{\phi}_k$ is necessary. Using the ρ operator (Currie *et al.*, 2006) defining the product

of a $i_1 \times K_1$ matrix by a $K_1 \times \ldots \times K_d$ array, yielding an array of size $K_2 \times \ldots \times K_d \times i_1$, one can rewrite Equation (22) as

$$\boldsymbol{\eta} = \rho(B_d, \ldots, \rho(B_2, \rho(B_1, \Phi)) \ldots)$$

where Φ is the $K_1 \times \ldots \times K_d$ array associated to the vector $\boldsymbol{\phi}$. When $d = 2$, we get back

$$\boldsymbol{\eta} = \rho(B_2, \rho(B_1, \Phi)) = B_1 \Phi B_2'$$

The roughness penalty prior becomes

$$p(\Phi|\tau_1, \ldots, \tau_d) \quad \propto \quad \sqrt{d(P)} \exp\left\{-\frac{1}{2} \operatorname{vec}(\Phi)' P \operatorname{vec}(\Phi)\right\}$$

where

$$\begin{aligned} P &= \tau_1 \mathcal{I}_{K_d} \otimes \ldots \otimes \mathcal{I}_{K_2} \otimes P_1 + \tau_2 \mathcal{I}_{K_d} \otimes \ldots \otimes \mathcal{I}_{K_3} \otimes P_2 \otimes \mathcal{I}_{K_1} \\ &\quad + \ldots + \tau_d P_d \otimes \mathcal{I}_{K_{d-1}} \otimes \ldots \otimes \mathcal{I}_{K_1} \end{aligned}$$

and K_i is the size of P_i.

In the special case where $\tau = \tau_1 = \ldots = \tau_d$, we have $d(P) \propto \tau^{\mathcal{R}(P)}$ where

$$\mathcal{R}(P) = \prod_{i=1}^{d} K_i - \prod_{i=1}^{d} [K_i - \mathcal{R}(P_i)]$$

Again, memory problems can be avoided by a careful implementation of the penalty prior. If the rotation of the d-dimensional array $\boldsymbol{A}$ of size $c_1 \times \ldots \times c_d$ is the d-dimensional array $R(\boldsymbol{A})$ of size $c_2 \times \ldots \times c_d \times c_1$ obtained by permuting the indices of $\boldsymbol{A}$ (Currie *et al.*, 2006), then one can write

$$\operatorname{vec}(\Phi)' P \operatorname{vec}(\Phi) \quad = \quad \sum \left\{ \Phi \odot \sum_{i=1}^{d} \tau_i \Psi_i \right\} \tag{23}$$

with

$$\Psi_i = R^{d-i}(\rho(P_i, R^{i-1}(\Phi)))$$

where $\sum(\boldsymbol{A})$ denotes the sum of the elements of the array $\boldsymbol{A}$ and $\odot$ is the obvious extension of the matrix dot product to arrays. When $d = 2$, we get back the expression in Equation (16) as $\operatorname{tr}(AB) = \sum A \odot B$ when B is symmetric.

The log of the joint posterior is

$$\begin{aligned} \log p(\Phi, \tau|\boldsymbol{N}) &= \sum \boldsymbol{N} \odot \log(\Pi) + \frac{1}{2} \log d(P) \\ &\quad - \frac{1}{2} \operatorname{vec}(\Phi)' P \operatorname{vec}(\Phi) + \sum_{i=1}^{d} [(a_i - 1) \log \tau_i - b\tau_i] \end{aligned} \tag{24}$$

where $\boldsymbol{N}$ denotes the $i_1 \times \ldots \times i_d$ array of frequencies and one assumes $\tau_i \sim \mathcal{G}(a_i,\ b_i)$.

In the special case where $\tau = \tau_1 = \ldots = \tau_d$, one can show that the gradient of Equation (24) is

$$\begin{aligned}
\frac{\partial \log p(\boldsymbol{\phi}, \tau|\boldsymbol{N})}{\partial \boldsymbol{\phi}} &= B'[\mathrm{vec}\,(\boldsymbol{N}) - \mathrm{vec}\,(n\Pi)] - P\ \mathrm{vec}\,(\Phi) \\
\frac{\partial \log p(\Phi, \tau|\boldsymbol{N})}{\partial \boldsymbol{\tau}} &= \frac{\mathcal{R}(P)}{2\tau} - \frac{1}{2\tau}\ \mathrm{vec}\,(\Phi)'\ P\ \mathrm{vec}\,(\Phi) + \frac{a-1}{\tau} - b
\end{aligned}$$

The first equation can be rewritten in a computationally efficient way using arrays:

$$\frac{\partial \log p(\Phi, \tau|\boldsymbol{N})}{\partial \Phi} = \rho(B'_d, \ldots, \rho(B'_2, \rho(B'_1, \boldsymbol{N} - n\Pi))\ldots) - \tau \sum_{i=1}^{d} \Psi_i$$

The same is true with the second equation after substitution of Equation (23).

The marginal posterior for Φ can also be derived in the special case:

$$\begin{aligned}
\log p(\Phi|\boldsymbol{N}) &= \sum \boldsymbol{N} \odot \log(\Pi) \\
&\quad - [a + 0.5\mathcal{R}(P)] \log \left\{ b + 0.5 \sum \left[\Phi \odot \sum_{i=1}^{d} \Psi_i \right] \right\}
\end{aligned}$$

with gradient

$$\begin{aligned}
\frac{\partial \log p(\Phi, \tau|\boldsymbol{N})}{\partial \Phi} &= \rho(B'_d, \ldots, \rho(B'_2, \rho(B'_1, \boldsymbol{N} - n\Pi))\ldots) \\
&\quad - [a + 0.5\mathcal{R}(P)]\ \frac{\sum_{i=1}^{d} \Psi_i}{b + 0.5 \sum \left[\Phi \odot \sum_{i=1}^{d} \Psi_i \right]}
\end{aligned}$$

Note that all the equations of this section generalize the corresponding results in Sections 2 and 3.

5 The Langevin-Hastings algorithm

5.1 The basic algorithm

Several algorithms based on McMC can be set up to explore the posterior. Conditionally on the penalty parameters, we are left with the well-studied problem of exploring the posterior of regression parameters in generalized linear models (see e.g. Gamerman, 1997; Brezger and Lang, 2006). The roughness penalty parameters, given the spline parameters, have identified conditional posterior distributions. Hence, a Gibbs sampler is easy to set up.

However, given the potentially large number of B-spline parameters, we believe that the Metropolis-adjusted Langevin algorithm (MALA, Roberts and Tweedie, 1996) is better suited as it just requires the computation of the log-posterior and of its gradient at each iteration: no potentially large precision matrix must be computed as in the Bayesian version of the Iterative Weighted Least Squares (IWLS) algorithm involved in the last two references. Moreover, if the marginal posterior of the B-spline parameters is considered, no sampling from the roughness penalty parameters is required.

The MALA algorithm builds McMC chains with proposals relying on the gradient of the log posterior distribution at the current state. More precisely, if $p(\boldsymbol{\theta}|\boldsymbol{y})$ is the posterior distribution and $\boldsymbol{\theta}^t \in \mathbb{R}^K$ the state of the chain at iteration t, then the proposal $\boldsymbol{\theta}$ for the next state is obtained by a random generation from the K-variate normal distribution $N_K(\boldsymbol{\theta}^t + 0.5\ \delta\ \nabla \log p(\boldsymbol{\theta}^t|\boldsymbol{y}), \delta\mathcal{I}_K)$ where $\mathcal{I}_K$ is the K dimensional identity matrix and δ a carefully chosen variance parameter. This proposal is accepted with probability

$$\alpha(\boldsymbol{\theta}^t, \boldsymbol{\theta}) = \min\left\{1, \frac{p(\boldsymbol{\theta}|\boldsymbol{y})}{p(\boldsymbol{\theta}^t|\boldsymbol{y})}\frac{q(\boldsymbol{\theta}, \boldsymbol{\theta}^t)}{q(\boldsymbol{\theta}^t, \boldsymbol{\theta})}\right\}$$

where

$$q(\boldsymbol{x}, \boldsymbol{z}) = (2\pi\delta)^{-K/2}\ \exp\left[-\frac{1}{2\delta}\left\|\boldsymbol{z} - \boldsymbol{x} - 0.5\delta\ \nabla \log p(\boldsymbol{x}|\boldsymbol{y})\right\|^2\right]$$

i.e. $\boldsymbol{\theta}^{t+1}$ is set equal to $\boldsymbol{\theta}$ if accepted and to $\boldsymbol{\theta}^t$ otherwise.

Roberts and Rosenthal (1998) have shown that the relative efficiency of the algorithm can be characterized by its overall acceptance rate, independently of the target distribution. The asymptotic optimal value for that last quantity is 0.57 with acceptance probabilities in the range $(0.40, 0.80)$ still reasonable. The parameter δ must be tuned to have an acceptance rate in that range.

5.2 Automatic tuning of the Langevin algorithm

An automatic tuning of δ targetting the optimal 0.57 rate is even possible (Haario *et al.*, 2001; Atchadé and Rosenthal, 2005): at the end of each iteration, set

$$\sqrt{\delta_{t+1}} = h\left(\sqrt{\delta_t} + \gamma_t\left(\alpha(\boldsymbol{\theta}^t, \boldsymbol{\theta}) - 0.57\right)\right)$$

where

$$h(x) = \begin{cases} \epsilon & \text{if} \quad x < \epsilon \\ x & \text{if} \quad x \in (\epsilon, A) \\ A & \text{if} \quad x > A \end{cases},$$

ϵ being a small number (say 10^{-4}) and A a large one (say $A = 10^4$). These two constants must be modified if the targetted acceptance rate is not attained.

The series $\{\gamma_t\}$ is a non-increasing sequence of positive real numbers such that $|\gamma_t - \gamma_{t-1}| \leq t^{-1}$. A possible choice for γ_t is $\gamma_t = t^{-1}$.

In practice, after reparametrization of the posterior (see Section 5.3), we run the adaptive Langevin algorithm for a few hundreds iteration with $\delta = 1.65^2/K^{1/3}$ as starting value for the tuning parameter. That value can be derived from the equations in Roberts and Rosenthal (1998, see Section 2) when the target posterior is the multivariate normal with identity variance-covariance matrix. The last value of δ_t in the so-generated sequence can then be used for the tuning parameter in the non-adaptive version of the Langevin algorithm to produce the long chain(s) used for inference.

5.3 Reparametrization of the posterior

If the tuned Langevin algorithm is directly used on the above derived posteriors, then one will observe large auto- and cross-correlations in the so-generated chain. This is not surprising as one expects that the B-splines parameters associated with neighbouring knots will take similar values, as imposed by the smoothness prior and also probably by the observed data (Lambert and Eilers, 2005). Therefore, a safe strategy consists in reparametrising the posterior before running the McMC algorithm. This can be achieved by a rough estimation of the B-splines parameters using, for example, a frequentist method for fixed and reasonably chosen values of the roughness penalty parameters. The IWLS algorithm is a possible choice as it quickly provides the MLEs and the hessian of the parameters in a polytomous logistic regression model. For example, in the 1D case, we iteratively apply

$$\boldsymbol{\phi}_{t+1} = (B'W_tB + \tau P)^{-1}B'(\boldsymbol{y} - n\boldsymbol{\pi}_t + W_tB\boldsymbol{\phi}_t)$$

where $\boldsymbol{\pi}_t = \pi(\boldsymbol{\phi}_t)$ and $W_t = \text{diag}\,(n\boldsymbol{\pi}_t(1-\boldsymbol{\pi}_t))$. The value of τ can be selected using cross-validation or an information criterion like the AIC (Eilers and Marx, 1996). The so-obtained MLE $\hat{\boldsymbol{\phi}}_\tau$ and its asymptotic variance-covariance matrix V_τ suggest reparametrising the posterior using $\boldsymbol{\phi}'$ where

$$\boldsymbol{\phi} = V_\tau^{1/2}\boldsymbol{\phi}' + \hat{\boldsymbol{\phi}}_\tau \ .$$

This device considerably reduces the posterior correlation and, hence, the poor mixing of the chain in the original parametrisation.

Note that one could also be tempted to plug-in a non-diagonal variance-covariance matrix in the proposal normal distribution of the Langevin algorithm and to estimate it in an adaptive McMC procedure (Haario *et al.*, 2001; Atchadé, 2005). This is certainly worth trying in situations where no reliable approximation of the posterior variance-covariance matrix is available, keeping in mind that this can be iteration (time) consuming before being effective[1].

[1]G.O. Roberts (2005) *Current challenges of adaptive Monte Carlo* at the workshop

6 Applications

6.1 The Old Faithful geyser

6.1.1 1D case

The data of interest are the durations of 272 eruptions of the Old Faithful geyser in Yellowstone National Park, Wyoming, USA. (Härdle, 1991). The interval (1,6) was split into 50 bins of width = 0.1. A cubic B-splines basis corresponding to 20 equidistant knots was considered on that interval. A chain of length 10,000 was run with a burn-in of 500 iterations to explore the marginal posterior (see Section 2.3). The obtained fit is reported in Figure 1. In the left part, one can see the fitted densities corresponding to the estimated posterior mean (solid line) of $\boldsymbol{\phi}$ and to the kernel density estimate (dashed line) with a bandwidth selected using a pilot estimation of derivatives (Sheather and Jones, 1991). The right part of the graph shows the fitted density (solid line) together with the 90% pointwise credible interval. It reveals the rather large uncertainty inherent in a density estimate. Note that (global) credibile envelopes can also be computed.

6.1.2 2D case

The data of interest now are the waiting times between and durations of 272 eruptions of the Old Faithful geyser. The waiting time and the duration axes were both divided into 50 bins on (35,105) and (1,6) respectively. Twenty-one equidistant knots were considered on both axes, yielding 529 spline parameters. Two different roughness penalty parameters were allowed, one for each axis (see Section 3.2). A chain of length 20,000 was generated after a burn-in of 500 iterations.

A graphical represensation of the fitted density is available on Figure 2. The left part shows the scatterplot together with the contours of the fitted density rescaled to be 1 at its maximum ; the right part displays the fitted bivariate density. The corresponding marginal densities are in Figure 3: the one corresponding to `Duration` is nearly identical to that obtained in Section 6.1.1 with the 1D approach, see Figure 1. Note that differences exist when one forces the two roughness penalty parameters to be the same.

6.2 Suicide treatment data

These data give the length of 86 spells of psychiatric treatments in a suicide study (Silverman, 1986). They were used by Eilers and Marx (1996) to show that the frequentist P-splines density estimate can be free from the boundary

Bayesian inference with biomedical applications, 17 November 2005, Brussels, Belgium. `http://www.stat.ucl.ac.be/ lambert/BiostatWorkshop2005/`

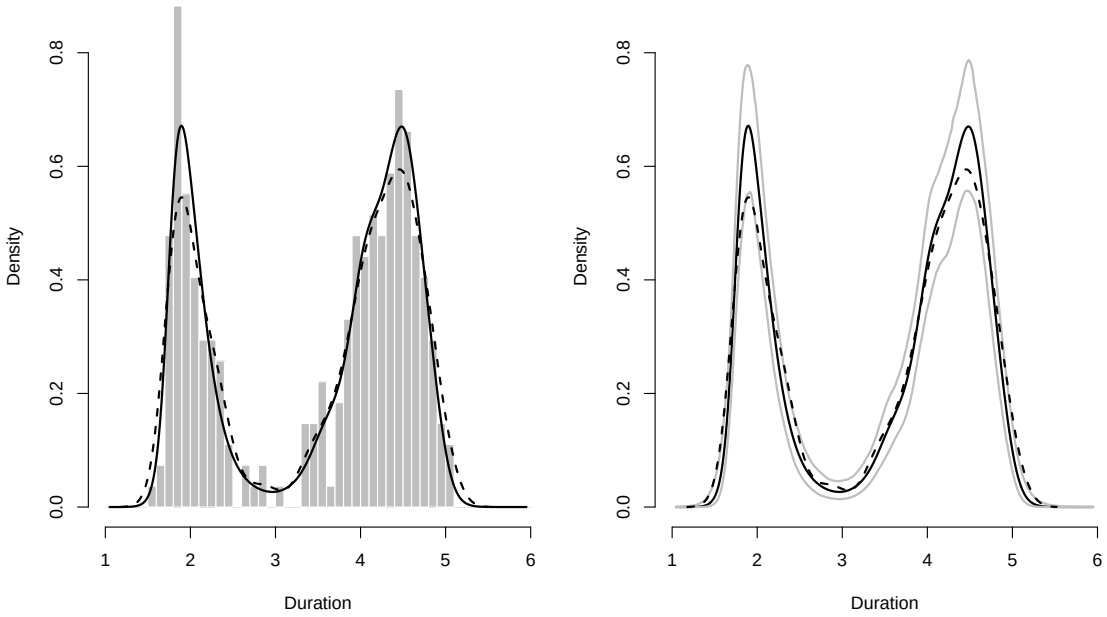


Figure 1: Durations of 272 eruptions of the Old Faithful geyser. Left graph: histogram and fitted density corresponding to the posterior mean (solid line) of $\boldsymbol{\phi}$ and to the kernel density estimate (dashed line). Right graph: fitted density (solid line) corresponding to the posterior mean of $\boldsymbol{\phi}$ together with the 90% pointwise credible interval and to the kernel density estimate (dashed line).

effect appearing with kernel smoothers, if the domain is properly chosen. Not surprisingly, this is confirmed with the Bayesian density estimate, see Fig. 4, where 100 bins and 20 knots were taken on $(0, 1000)$; a chain of length $10,000$ with a burn-in of 500 was built to explore the marginal posterior (see Section 2.3). Provided that the lower limit of the domain is chosen equal to the smallest possible value for a duration (here 0) no boundary effect shows up.

6.3 Graphical summary of Monte Carlo simulations

In this subsection we apply our density smoother, which use McMC simulations to visualize the output of a another simulation. Graphical summaries of Monte Carlo generations most often rely on 1D estimates of marginal densities. As an example, consider the bioassay experiment reported in Gelman *et al.* (2004, pp. 88-93, 104-106) giving the number of deaths y_i observed in batches of n_i animals exposed to different possible log-doses x_i of a chemical (see Table 1). These data were modelled using logistic regression

$$y|\theta_i \sim \text{Bin}(n_i, \theta_i)$$

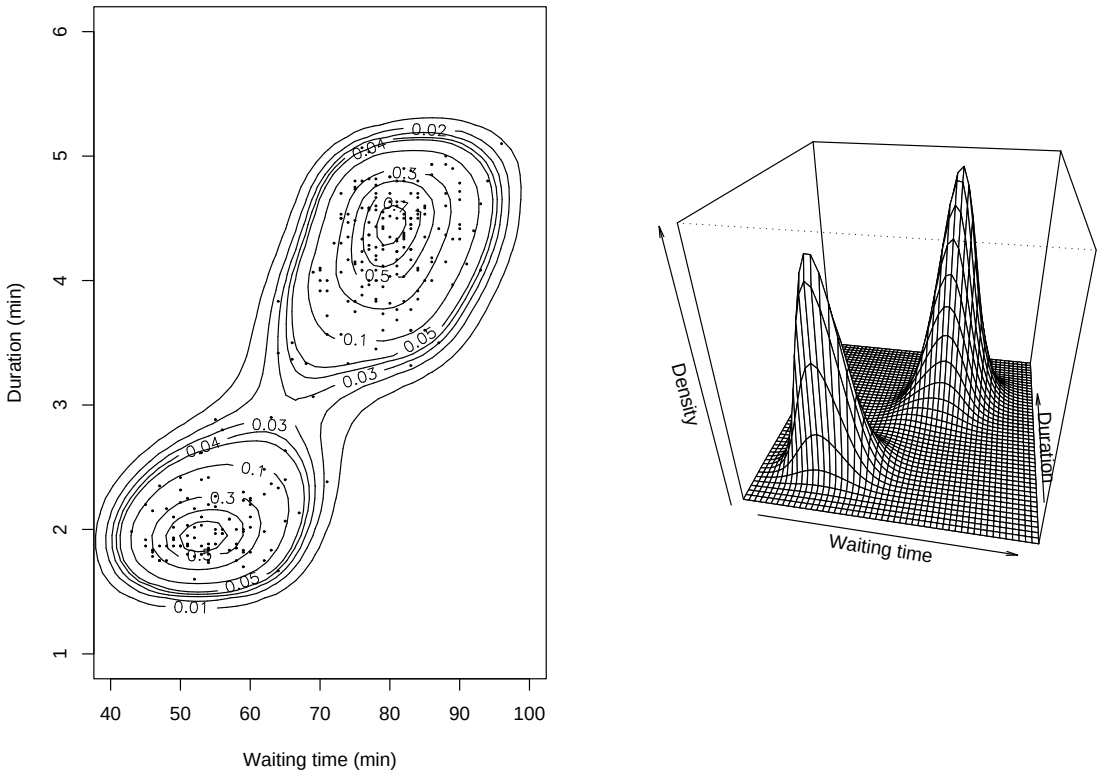


Figure 2: Waiting times between and durations of 272 eruptions of the Old Faithful geyser. Left part: scatterplot and contours of the fitted density (rescaled to be 1 at its maximum); right part: fitted bivariate density.

log-dose x_i (log g/ml)	Number of animals n_i	Number of deaths y_i
-0.86	5	0
-0.30	5	1
-0.05	5	3
0.73	5	5

Table 1: Bioassay data

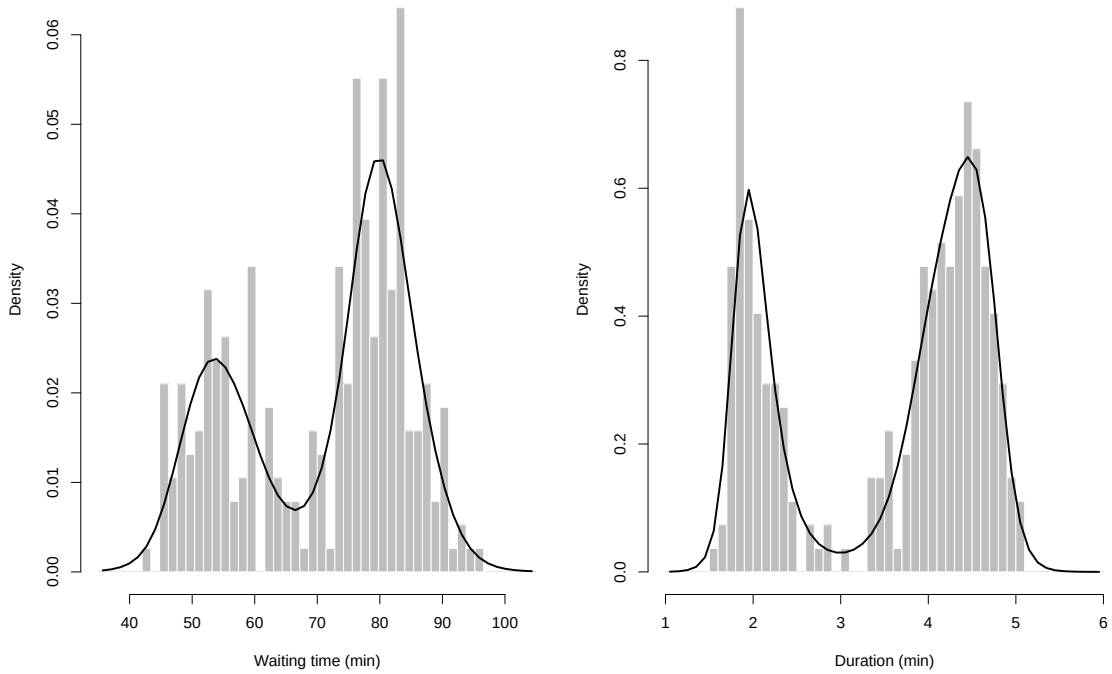


Figure 3: Waiting times between and durations of 272 eruptions of the Old Faithful geyser: marginal densities corresponding to the fitted bivariate density.

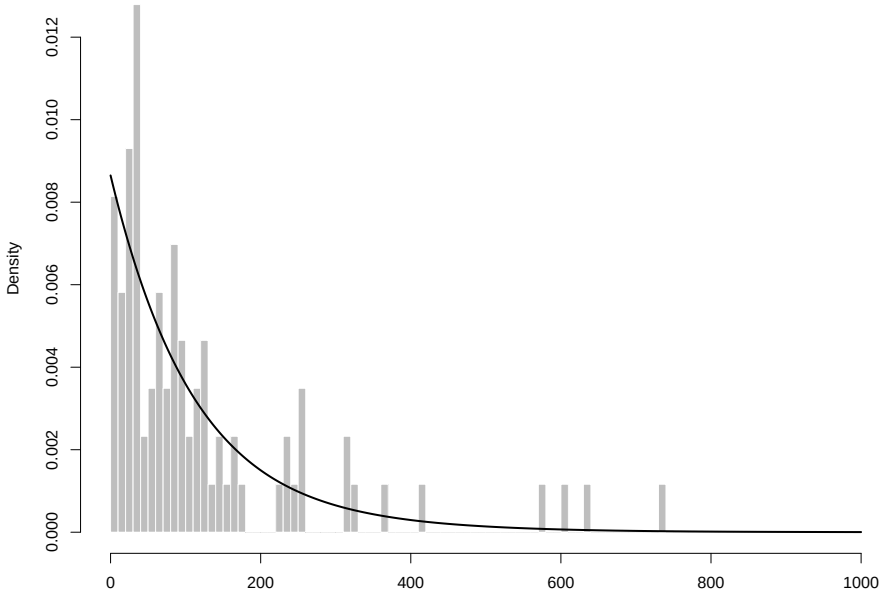


Figure 4: Suicides data: fitted density estimate.

$$\log \frac{\theta_i}{1-\theta_i} = \alpha + \beta x_i \quad ; \quad p(\alpha, \beta) \propto 1$$

where θ_i denotes the probability of death at log-dose x_i.

A random sample $\{(\alpha_t, \beta_t) : t = 1, \ldots, 10000\}$ from the joint posterior for (α, β) was generated using the Metropolis-Hastings algorithm: the corresponding chain can be visualized using CODA, see Fig. 5. Alternatively, the marginal density estimates can be computed using the Bayesian method described in Section 2 with, for α and β, 50 bins and 20 equidistant knots on $[-3, 8]$ and $[-5, 45]$, respectively. Figure 6 shows the autocorrelation of the Markov chain for the slope of the regression line. Interpreting the generated chain as an independent sample of size $10,000$ gives the potentially misleading density estimate in the middle part of Figure 6. If, instead, the estimates are built on the thinned samples obtained by taking every 20th element of the chain (as suggested by the autocorrelation function estimate given in the left part of the same figure), one obtains the estimates given in the right part of Figure 6.

For this small data set the joint posterior $p(\alpha, \beta | \boldsymbol{y}, \boldsymbol{n}, \boldsymbol{x})$ can computed numerically: some contours are plotted on the left part of Fig. 7 (see also Fig. 3.4 in Gelman *et al.* (2004)). Alternatively, the bivariate posterior density can be estimated from the thinned chain $\{(\alpha_{1+20j}, \beta_{1+20j}) : j = 0, \ldots, 499\}$ using the method presented in Section 3 with the same bins and knots as in the 1D case. Two different roughness penalty parameters were allowed, one for each axis (see Section 3.2). A chain of length 20,000 was generated after a burn-in of 1,000 iterations. The fitted contours are given on the right part of Fig. 7: they are in good agreement with the exact ones (in the left part of Fig. 7).

6.4 Quantile regression

A nice application of density estimation is quantile regression. If an estimate of the joint density $f_{Y_1 Y_2}$ is available, then it is straightforward to derive the estimates of the corresponding marginal and conditional densities from their respective definitions.

In addition, one can derive the conditional distribution functions $F_{Y_1|Y_2=y_2}$, $F_{Y_2|Y_1=y_1}$ by integrating the corresponding density estimate. These can be inverted to finally obtain estimates of the conditional quantile functions.

For the Old Faithful data we illustrate this with conditional quantiles of `Duration` given `Waiting time`. The deciles are plotted in Figure 8: as expected, these are smooth and do not cross as sometimes happens with some nonparametric methods.

Pointwise credible interval for these deciles can be obtained from the generated chain: the 80% credible intervals are plotted in Figure 9 for selected deciles.

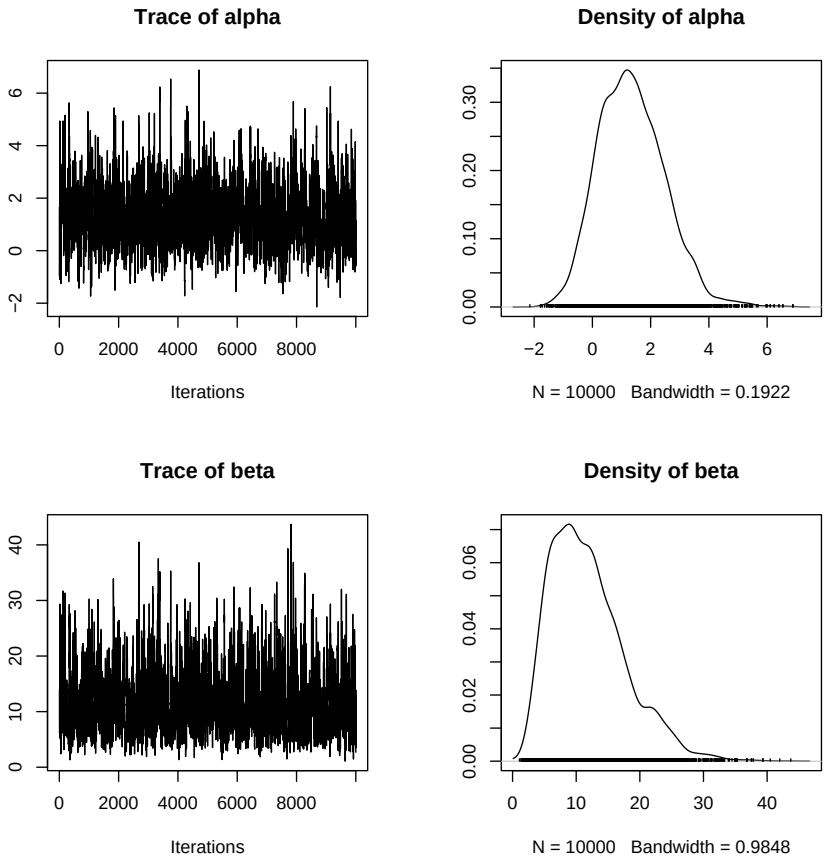


Figure 5: Bioassay data: graphical visualization, using CODA, of a chain of length 10,000 drawn from the posterior $p(\alpha, \beta|\boldsymbol{y})$.

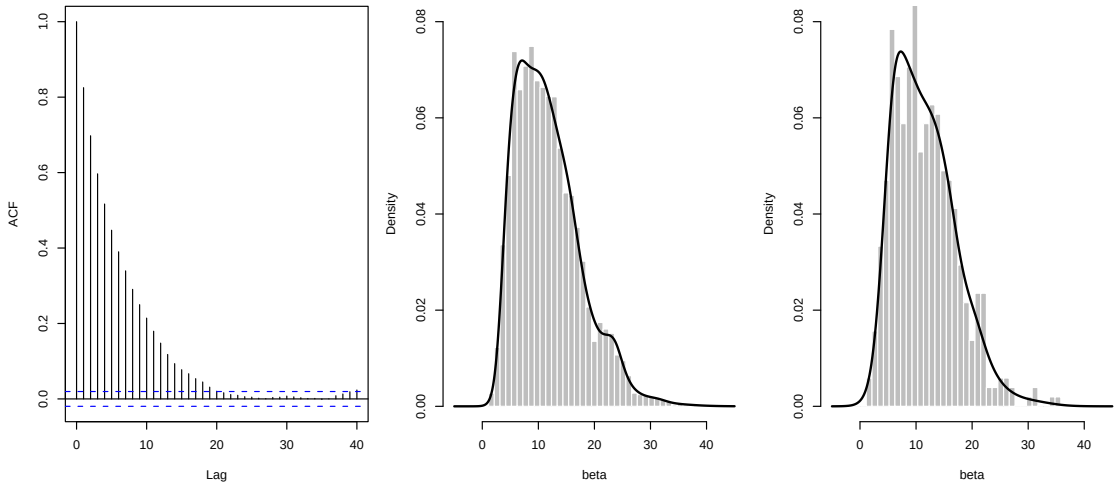


Figure 6: Bioassay data. Left part: autocorrelation function of the Markov chain for the slope of the regression line. Middle part: Bayesian density estimate of when assuming 10000 independent observation. Right part: Bayesian density estimate after thinning (keeping every 20th simulated value), to break the autocorrelation.

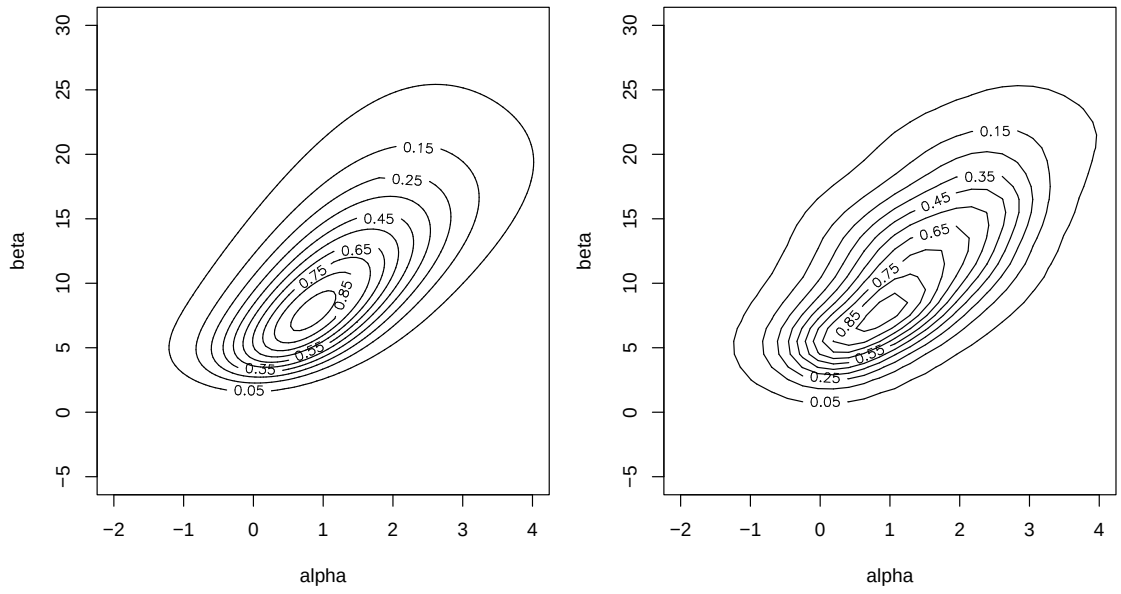


Figure 7: Bioassay data. Exact (left part) and estimated (right part) contours of $p(\alpha, \beta|\boldsymbol{y}, \boldsymbol{n}, \boldsymbol{x})$ rescaled to be one at its maximum.

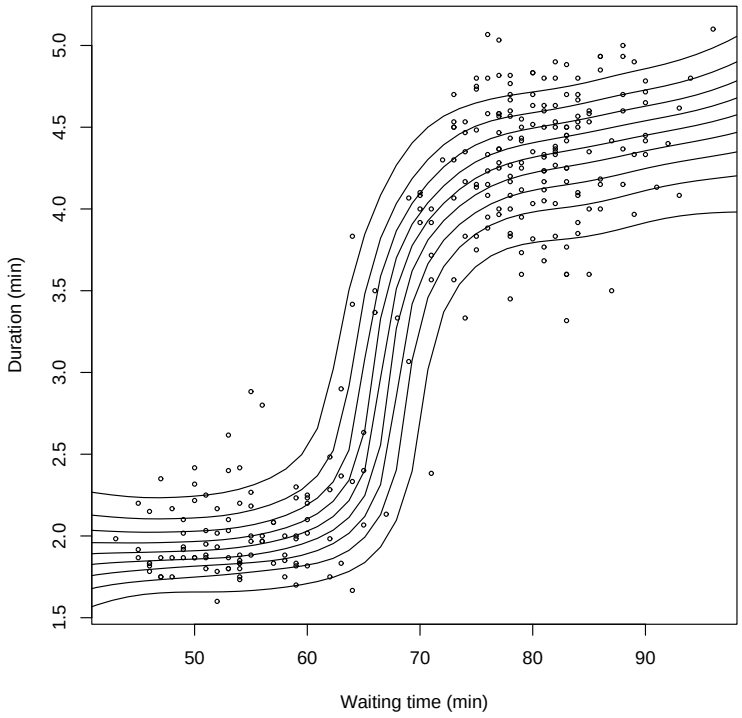


Figure 8: Waiting times between and durations of 272 eruptions of the Old Faithful geyser: conditional deciles of `Duration` for given `Waiting time`.

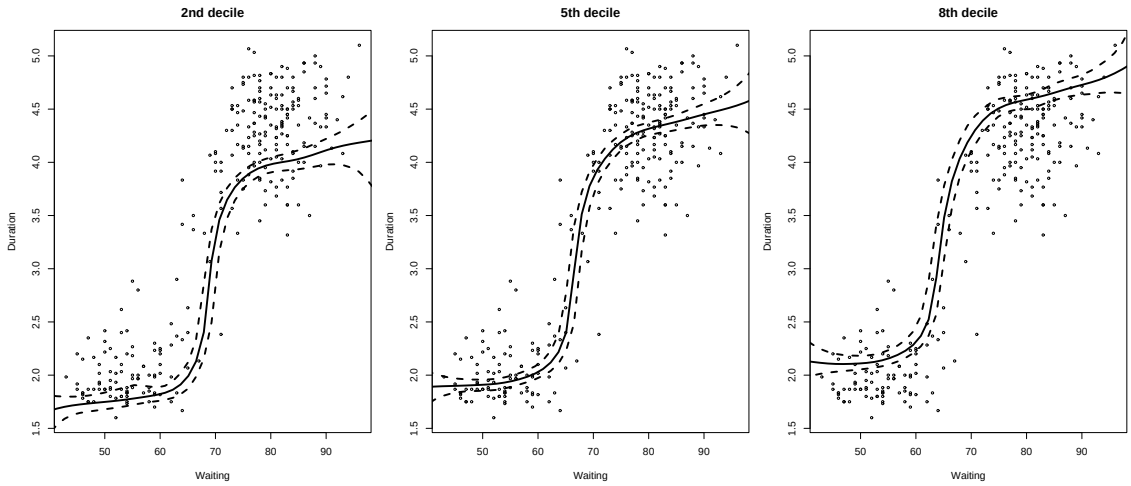


Figure 9: Waiting times between and durations of 272 eruptions of the Old Faithful geyser: 80% pointwise credible intervals of the 2nd, 5th and 8th conditional deciles of `Duration` for given `Waiting time`.

6.5 Implementation

Bayesian computation was performed using an R program interfacing a routine in C that implements the Langevin algorithm. It is extremely quick in the 1D case with, for the geyser data, 0.6 seconds required to build a chain of length 10,000 (see Section 6.1.1) on a Pentium IV 3.0 GHz. In the 2D example where 21 equidistant knots along both axes were considered (yielding 531 parameters, including the two roughness penalty parameters), it took about 5.3 seconds per 1,000 runs. This reduces to 1.1 seconds with 10 knots in each direction.

7 Discussion

We feel that our penalized spline approach to Bayesian density estimation has many attractive properties. It is effective and handles higher dimensions in a unified way. Computation time is a few seconds for one-dimensional application, including the determination of the optimal amount of smoothing, making this smoother attractive for everyday use. Optimal smoothing of a two-dimensional density takes around one minute.

Besides being interesting in its own right, Bayesian estimation of densities using McMC methods is a practically worthwhile alternative to existing frequentist approaches (where these exist) with, as a great addition, the availability of the generated chain(s) to estimate and produce credible regions of any function of the density. Moreover, the same principles can be used, whatever the dimension of the problem.

A interesting application is smooth quantile regression, formulated as a by-product of two-dimensional density estimation. It was illustrated in the case where a single continuous regressor is available: the so-obtained curves do not cross as it is often the case with nonparametric methods. In addition, credible regions can be constructed using the generated chain(s). Extension to larger dimensions (smooth quantile surfaces, based on smooth 3D densities) is feasible and will be reported elsewhere.

The basis of our model is a generalized linear model using (tensor products) of penalized splines. This means that extension to other non-normal data, like binary fields (using a logit link function and a binomial distribution) or variance fields (using a log link function and the Gamma distribution) is straightforward. See also Lambert and Eilers (2005) for an application in advanced survival analysis.

Acknowledgements

Financial support from the IAP research network nr P5/24 of the Belgian State (Federal Office for Scientific, Technical and Cultural Affairs) for Philippe Lambert

is gratefully acknowledged.

References

Atchadé, Y. F. (2005). An adaptive version for the Metropolis adjusted Langevin algorithm with a truncated drift. Technical report, University of Ottawa.

Atchadé, Y. F. and Rosenthal, J. S. (2005). On adaptive Markov chain Monte Carlo algorithms. *Bernoulli*, **11**, 815–828.

Berry, S. M., Carroll, R. J., and Ruppert, D. (2002). Bayesian smoothing and regression splines for measurement error problems. *Journal of the American Statistical Association*, **97**, 160–169.

Bolstad, W. (2004). *Introduction to Bayesian Statistics*. John Wiley & Sons, Hoboken, New Jersey.

Brezger, A. and Lang, S. (2006). Generalized structured additive regression based on Bayesian P-splines. *Journal of Computational Statistics and Data Analysis*, **50**, 967–991. (in press).

Currie, I. D., Durbán, M., and Eilers, P. H. C. (2004a). Efficient smoothing of d-dimensional arrays. In *Proceedings of the 19th International Workshop on Statistical Modelling (Florence, Italy)*, pages 139–143.

Currie, I. D., Durbán, M., and Eilers, P. H. C. (2004b). Smoothing and forecasting mortality rates. *Statistical Modelling*, **4**, 279–298.

Currie, I. D., Durbán, M., and Eilers, P. H. C. (2006). Generalized linear array models with applications to multidimensional smoothing. *Journal of the Royal Statistical Society, Series B*, **68**, 259–280.

Eilers, P. H. C. and Goeman, J. J. (2004). Enhancing scatterplots with smoothed densities. *Bioinformatics*, **20**, 632–638.

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

Eilers, P. H. C., Currie, I. D., and Durbán, M. (2006). Fast and compact smoothing on large multidimensional grids. *Journal of Computational Statistics and Data Analysis*, **50**, 61–76.

Gamerman, D. (1997). Efficient sampling from the posterior distribution in generalized linear models. *Statist. Comput.*, **7**, 57–68.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2004). *Bayesian Data Analysis, 2nd edition*. Chapman & Hall/CRC.

Haario, H., Saksman, E., and Tamminen, J. (2001). An adaptive Metropolis algorithm. *Bernoulli*, **7**, 223–242.

Hansen, M. H. and Kooperberg, C. (2002). Spline adaptation in extended linear models (with discussion). *Statistical Science*, **17**, 2–51.

Härdle, W. (1991). *Smoothing Techniques with Implementation in S*. New York: Springer.

Heikkinen, J. and Arjas, E. (1998). Non-parametric Bayesian estimation of spatial Poisson intensity. *Scandinavian Journal of Statistics*, **25**, 435–450.

Jullion, A. and Lambert, P. (2005). Robust specification of the roughness penalty prior distribution in spatially adaptive bayesian p-splines models. *Journal of Computational Statistics and Data Analysis*. (conditionally accepted).

Kooperberg, C. and Stone, C. J. (1991). A study of logspline density estimation. *Journal of Computational Statistics and Data Analysis*, **12**, 327–347.

Lambert, P. (2005). Archimedean copula estimation using Bayesian splines smoothing techniques. Discussion paper 05-27, Institut de Statistique, Université catholique de Louvain, Louvain-la-Neuve, Belgium. http://www.stat.ucl.ac.be/ISpub/ISdp.html.

Lambert, P. and Eilers, P. H. (2005). Bayesian proportional hazards model with time varying regression coefficients: a penalized Poisson regression approach. *Statistics in Medicine*, **24**, 3977–3989.

Lang, S. and Brezger, A. (2004). Bayesian P-splines. *Journal of Computational and Graphical Statistics*, **13**, 183–212.

Müller, P. and Vidakovic, B. (1998). Bayesian inference with wavelets: density estimation. *Journal of Computational and Graphical Statistics*, **7**, 456–468.

O'Sullivan, F. (1988). Fast computation of fully automated log-density and log-hazard estimators. *SIAM J. Sci. Stat. Comput.*, **9**, 363–379.

Roberts, G. O. and Rosenthal, J. S. (1998). Optimal scaling of discrete approximations to Langevin diffusions. *Journal of the Royal Statistical Society, Series B*, **60**, 225–268.

Roberts, G. O. and Tweedie, R. L. (1996). Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, **24**, 341–363.

Roeder, K. and Wasserman, L. (1997). Practical bayesian density estimation using mixtures of normals. *Journal of the American Statistical Association*, **92**, 894–902.

Sheather, S. J. and Jones, M. C. (1991). A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society, Series B*, **53**, 683–690.

Silverman, B. W. (1986). *Density Estimation for Statistics and Data analysis*. Chapman & Hall, London.

Stephens, M. (2000). Dealing with label switching in mixture models. *Journal of the Royal Statistical Society, Series B*, **62**, 795–809.