Towards a Faster Implementation of Density Estimation with Logistic Gaussian Process Priors

Surya T. Tokdar*

Department of Statistics, Carnegie Mellon University, Pittsburgh, PA 15213

Abstract

A novel method is proposed to compute the Bayes estimate for a logistic Gaussian process prior for density estimation. The method gains speed by drawing samples from the posterior of a finite dimensional surrogate prior, which is obtained by imputation of the underlying Gaussian process. We establish that imputation results in quite accurate computation. Simulation studies show that accuracy and high speed can be combined. This fact, along with known flexibility of the logistic Gaussian priors for modeling smoothness and recent results on their large support, makes these priors and the resulting density estimate very attractive.

Keywords and Phrases: Bayesian nonparametrics, Imputation, Markov Chain Monte Carlo.

1 Introduction

Logistic Gaussian process priors for density estimation were introduced and studied by Leonard (1978) and Lenk (1988, 1991). These nonparametric priors are easy to specify. One simply needs to elicit the mean and the covariance functions of the underlying Gaussian process. The (logistic transform of the)

*Surya T Tokdar is Morris H. DeGroot Visiting Assistant Professor, Department of Statistics, Carnegie Mellon University, Pittsburgh, PA 15213 (email: stokdar@stat.cmu.edu).

mean reflects the prior guess and the covariance determines the differentiability and smoothness properties of the sample paths (see Lenk 1988). In this sense, a logistic Gaussian prior has greater flexibility in modeling smoothness directly - a property not shared by other popular nonparametric priors such as the Dirichlet mixture of normals or the Polya tree.

Theoretical properties of logistic Gaussian priors have been recently investigated by Tokdar and Ghosh (2006). They show that in many cases the posterior is consistent for estimating densities supported on a closed bounded interval in $\mathbb{R}^d$. For example, standard choices of the covariance with a suitable prior on its (smoothing) parameters produce a consistent posterior when data arise from any continuous density.

However, a long standing problem with logistic Gaussian priors has been the difficulty in computing the posterior. The reason for this appears to be an intractable integration term that features in the likelihood. The presence of this term and the high spatial correlation of the sample paths make MCMC or importance sampling or a combination thereof difficult to implement. Lenk (1988) suggests a couple of methods to approximate the Bayes estimate, but these methods have not been well validated.

Lenk (1991) and Verdinelli and Wasserman (1998) use orthogonal series expansion of the underlying Gaussian process to devise feasible computing algorithms. Such expansions can be easily incorporated into a Gibbs sampling scheme to generate samples from the posterior. Theoretically, a series expansion can be derived from the Karhunen-Loéve representation of the process. However, an explicit representation is rarely available - making the applicability of this approach substantially restricted.

Our aim is to develop a method of computation that applies to any logistic Gaussian prior and produces fast and reasonably accurate samples from the posterior. In this pursuit, we propose a novel technique based on imputation of the underlying Gaussian process. The idea is to retain the process only at a fixed grid of nodes and then impute the rest analytically. This leads to a finite dimensional surrogate prior for which on line samples can be drawn from the posterior using a Metropolis-Hastings MCMC algorithm.

Our method of approximation crucially depends on the grid of nodes used for imputation. Theoretically, one does better by making the grid finer; we prove this result formally in Theorem 3.1. However, using an arbitrarily fine grid may be practically infeasible and even unsatisfactory for reasons discussed in Section 4.1. This motivates us to consider the grid of nodes as a part of the model parameters and let the data select the best set of grids. We implement this data driven selection of nodes by introducing a reversible jump step in our MCMC algorithm.

Logistic Gaussian priors are generally defined for densities supported on a bounded interval to avoid certain integrability problems (see Tokdar and Ghosh 2005). In the univariate case, Verdinelli and Wasserman (1998) use a simple parametric transformation to induce a prior on densities defined on the entire real line. We extend this technique to higher dimensions. Our imputation based computing scheme easily adapts to such extensions. However, the time taken for computation increases substantially, from a few seconds in the univariate case to a few minutes in the bivariate case.

The rest of this paper is organized as follows. Section 2 gives a basic introduction to logistic Gaussian process priors. In Section 3 we propose the idea of an approximation based on imputation. The resulting computing algorithms are discussed in Section 4. A detailed example is presented in Section 5 with discussions on choice of hyperparameters, comparison with exact posterior and convergence diagnosis of the proposed MCMC sampler. In Section 6 we outline how to define a logistic Gaussian prior on an arbitrary interval by using a family of transformations. Some interesting examples are worked out toward the end of this section. The last example of this section highlights an advantage of using the transformations even when the support is known to be bounded. We conclude the paper with a discussion in Section 7.

2 Logistic Gaussian Process Priors

We start with the definition of a logistic Gaussian process prior. For ease of illustration, we would stick to a simple version of the prior, for more general definitions, see Lenk (1991).

Take $\mathcal{I} = [0,1]^d$ and let $\sigma_0(\cdot,\cdot)$ be a fixed positive definite function on $\mathbb{R}^d \times \mathbb{R}^d$. Define a real valued process f_W on $\mathcal{I}$ as follows,

$$f_W(t) = \frac{e^{W(t)}}{\int_{\mathcal{I}} e^{W(s)} ds},\ t \in \mathcal{I} \tag{1}$$

where, given $\gamma = (\tau, \beta) \in \mathbb{R}^+ \times (\mathbb{R}^+)^d$, $W(\cdot)$ is a separable, zero mean Gaussian process on $\mathcal{I}$ with covariance

$$\sigma_\gamma(s,t) = \tau^2 \sigma_0(\beta s, \beta t),\ s, t, \in \mathcal{I} \tag{2}$$

and $\gamma \sim H$, a distribution on $\mathbb{R}^+ \times (\mathbb{R}^+)^d$, with density h. In (2), βs denotes the vector of coordinatewise products of β and s when $d > 1$.

Clearly f_W defines a stochastic process on $\mathcal{I}$ whose realizations (sample paths) satisfy $f_W \geq 0$ and $\int_0^1 f_W(t)dt = 1$ - properties which define a density function over $\mathcal{I}$. We let Π denote the probability measure governing this stochastic process. Then Π can be thought of as a prior distribution on the space of densities over $\mathcal{I}$. Such a prior would be called a logistic Gaussian process prior.

Remark The definition of f_W implicitly involves an integrability assumption. But for many choices of σ_γ this is not a problem. We would tacitly assume this to be true in the rest of this paper (see Tokdar and Ghosh (2006) for more details).

Remark The use of a family of covariance functions $\{\sigma_\gamma\}$, instead of a single covariance function, adds to the flexibility of the prior Π. To justify the choice of the particular family given in (2), we note that,

$$W \sim GP_{\mathcal{I}}(0, \sigma_\gamma) \iff W(\cdot) \overset{\mathcal{L}}{=} \tau W_0(\beta \cdot) \text{ with } W_0 \sim GP_{\beta\mathcal{I}}(0, \sigma_0), \tag{3}$$

where the notation $GP_{\mathcal{K}}(0, \sigma)$ is used for the distribution of a separable, zero mean Gaussian process defined on a set $\mathcal{K}$ with covariance σ. Hence, with $\{\sigma_\gamma\}$ as in (2), small β results in smooth sample paths of f_W and large β produces oscillating sample paths. In other words, β acts like a (inverted) smoothing window in this model. The parameter τ controls the overall variability of f_W from its prior guess; we elaborate more on this in Section 6. The base covariance kernel σ_0 determines the degree of differentiability of the sample paths of W and can be selected appropriately to reflect prior expectations.

3 An Approximation through Imputation

The integral that appears in the denominator of (1) involves the entire sample path of W, which is an infinite dimensional object. The presence of this term makes it infeasible to carry out any likelihood based computation. Therefore, from the perspective of efficient computing, we seek a good, easy to use, finite dimensional approximation to the process W. We achieve this by retaining W only at a finite set of nodes $T = \{t_1, \cdots, t_m\} \subset S$ and imputing the rest by using conditional expectation. More formally, we approximate W by a new process Z

defined as,

$$Z(t) = E(W(t)|\mathbf{W}_m, \gamma),\ t \in \mathcal{I} \tag{4}$$

where $\mathbf{W}_m = (W(t_1), \cdots, W(t_m))$. This leads to an approximation of f_W by f_Z, the logistic transform of Z given by,

$$f_Z(t) = \frac{e^{Z(t)}}{\int_{\mathcal{I}} e^{Z(s)} ds}, t \in \mathcal{I}. \tag{5}$$

The process f_Z is much easier to deal with than the original process f_W. This is because Z can be written in closed form as a function of the finite dimensional vector $(\mathbf{W}_m, \gamma)$:

$$Z(t) = \mathbf{W}_m^{\mathrm{T}} \mathbf{\Sigma}_\gamma^{-1} \boldsymbol{\sigma}_\gamma(t) \tag{6}$$

where,

$$\mathbf{\Sigma}_\gamma = \begin{pmatrix} \sigma_\gamma(t_1, t_1) & \sigma_\gamma(t_1, t_2) & \cdots & \sigma_\gamma(t_1, t_m) \\ \sigma_\gamma(t_2, t_1) & \sigma_\gamma(t_2, t_2) & \cdots & \sigma_\gamma(t_2, t_m) \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_\gamma(t_m, t_1) & \sigma_\gamma(t_m, t_2) & \cdots & \sigma_\gamma(t_m, t_m) \end{pmatrix},\ \boldsymbol{\sigma}_\gamma(t) = \begin{pmatrix} \sigma_\gamma(t_1, t) \\ \sigma_\gamma(t_2, t) \\ \vdots \\ \sigma_\gamma(t_m, t) \end{pmatrix}.$$

Therefore Π^*, the distribution of f_Z, can be viewed as a surrogate prior that provides a finite dimensional approximation to Π.

Note that Π^* can be equivalently expressed as the distribution of the process

$$f_{\mathbf{X}^{\mathrm{T}}\mathbf{A}_\gamma}(\cdot) = \frac{e^{\mathbf{X}^{\mathrm{T}}\mathbf{A}_\gamma(\cdot)}}{\int_{\mathcal{I}} e^{\mathbf{X}^{\mathrm{T}}\mathbf{A}_\gamma(s)} ds} \tag{7}$$

where $(\mathbf{X}, \gamma) \sim N_m(\mathbf{0}, \mathbf{I}_m) \times H$ and $\mathbf{A}_\gamma = \mathbf{\Sigma}_\gamma^{-1/2} \boldsymbol{\sigma}_\gamma(\cdot)$. This is because, given γ, $\mathbf{W}_m \sim N_m(\mathbf{0}, \mathbf{\Sigma}_\gamma)$ and hence Z has the same distribution as the process $\mathbf{X}^{\mathrm{T}}\mathbf{A}_\gamma$. Here $N_m(\boldsymbol{\mu}, \mathbf{\Sigma})$ denotes the m-variate normal distribution with mean

$\boldsymbol{\mu}$ and covariance $\mathbf{\Sigma}$ and $\mathbf{I}_m$ is the $m \times m$ identity matrix.

The representation displayed in (7) gives an explicit and easy to use formulation of the finite dimensional surrogate prior Π^*. This formulation would be very useful for the computing scheme that we discuss in Section 4. But before that, we provide a brief discussion on why we selected this particular type of approximation to the the process f_W.

3.1 Motivation behind Imputation: A Result

Our motivation behind this particular type of imputation is the following. Tokdar and Ghosh (2006) show that, under some conditions on the family σ_γ, the original process W realizes its sample paths in the sup norm closure of the following set

$$\mathcal{A} = \{w = \sum_{i=1}^{k} a_i \sigma_\gamma(s_i, \cdot) : \ k \geq 1, \ s_i \in \mathcal{I}, \ a_i \in \mathbb{R}, \gamma \in \text{support}(H)\}. \tag{8}$$

Because of the identity in (6), the new process Z also realizes its sample paths in $\mathcal{A}$. Moreover, by definition, at each node t_i, $Z(t_i) = W(t_i)$. Therefore, Z provides an a priori interpolation-type approximation to W, staying within the target set $\mathcal{A}$. This gives the assurance that one does not lose much by replacing W with Z.

While the above a priori approximation property of Z is aesthetically satisfying, our interest lies in what happens a posteriori. Let, $\mathbf{y} = (y_!, \cdots, y_n)^{\mathrm{T}}$ denote a sample of observations from some unknown density f. We can either use an actual logistic Gaussian process model $f = f_W$ or use its imputation based approximation $f = f_Z$. The imputation method would be really useful if the posterior distribution of f_Z given $\mathbf{y}$ continues to well resemble the posterior distribution of f_W given $\mathbf{y}$. A high degree of resemblance between these two distributions is reflected in a small value of the Kullback-Leibler distance $K(\hat{f}_W, \hat{f}_Z)$

between the predictive densities under these two models: $\hat{f}_W = E(f_W|\mathbf{y})$ and $\hat{f}_Z = E(f_Z|\mathbf{y})$. The next theorem establishes that this distance converges to 0 as more tightly packed nodes are used.

Theorem 3.1 *Assume that, $\exists c > 0, q > 0$ such that for all $s, t \in \mathbb{R}^d$ and $\gamma \in \text{support}(H)$,*

$$\sqrt{\text{Var}[W(s) - W(t)|\gamma]} < c\|s - t\|^q. \tag{9}$$

Let $\delta(T) = \sup_{t \in \mathcal{I}} \min_{t_j \in T} \|t - t_j\|$ denote the fineness of the nodes. Then, $K(\hat{f}_W, \hat{f}_Z) \to 0$ as $\delta(T) \to 0$.

Remark The assumption in (9) ensures continuous sample paths of the process W given γ. Such a requirement is indispensable given the method of interpolation used for constructing Z from W. However, the above theorem demands this condition in a strong form; a fixed constant c appears on the right hand side, instead of a more flexible $c(\gamma)$. The reason for this is a technical complication that arises when c varies arbitrarily with γ. For the particular formulation of σ_γ given in (2), the condition (9) necessitates that $c(\gamma) := \tau \max_l \beta_l$ be uniformly bounded from above. Under such strong assumptions, the convergence actually holds uniformly over all samples $\mathbf{y}$ of size n. However, during implementation, we would meet this technical requirement only approximately, by choosing H with exponentially decaying tails; uniform convergence should not be expected for this case.

In Section 4.1 we discuss two major problems with using an arbitrarily fine grid of nodes and recommend using a moderately sized, dynamic, data dependent choice of this grid. In this sense the above theorem does not provide an irrefutable theoretical validation to the approximation scheme introduced in this paper. Nevertheless, the above theorem lends considerable motivational support to the use of this particular approximation and in Section 5.1 we demonstrate that in reality this approximation leads to quite accurate computation.

4 Computation of Posterior via MCMC

Let $\mathbf{y} = (y_1, \cdots, y_n)^{\mathrm{T}}$ be a sample of observations from an unknown density f. We use the surrogate prior Π^* and model f as $f = f_{\mathbf{X}^{\mathrm{T}}\mathbf{A}_\gamma}$. Then, to obtain the posterior distribution of f given $\mathbf{y}$, it suffices to know the posterior distribution of $(\mathbf{X}, \gamma)$ given $\mathbf{y}$. A simple application of the Bayes rule shows that the posterior density of $(\mathbf{X}, \gamma)$ (with respect to the product of the Lebesgue measures on $\mathbb{R}^m$ and $\mathbb{R}^+ \times (\mathbb{R}^+)^d$) given the data $\mathbf{y} = (y_1, \cdots, y_n)$ can be written as,

$$p^*((\mathbf{x}, \gamma) \mid \mathbf{y}) \propto \left\{ \prod_{j=1}^{n} f_{\mathbf{x}^{\mathrm{T}}\mathbf{A}_\gamma}(y_j) \right\} \phi_m(\mathbf{x}) h(\gamma), \tag{10}$$

where $\phi_m(\cdot)$ is the pdf of the m-variate standard normal distribution.

Since the posterior density can be written explicitly (up to a constant factor), it is possible to use a Metropolis-Hastings MCMC sampler to draw samples from it. The following algorithm gives a possible construction of such a sampler.

Algorithm I:

Initialize: Generate $\mathbf{X} \sim N_m(\mathbf{0}, \mathbf{I}_m)$ and $\gamma \sim H$.

Update: Iteratively update each coordinate of the vector $(\mathbf{X}, \gamma)$. Let $\mathbf{x} = (x_1, \cdots, x_m)^{\mathrm{T}}$ be the current value of $\mathbf{X}$ and suppose we want to update the first coordinate. We propose to move $\mathbf{X}$ to $\mathbf{x}' = (x_1', x_2, \cdots, x_m)^{\mathrm{T}}$ where x_1' is generated according to a $N(x_1, \sigma_{x1}^2)$ distribution with some variance σ_{x1}^2. Following the principles of the Metropolis-Hastings algorithm, we accept this move with a probability

$$\alpha = \min\left\{1, \frac{\phi_1(x_1') \prod_{j=1}^n f_{\mathbf{X}'^{\mathrm{T}}\mathbf{A}_\gamma}(y_j)}{\phi_1(x_1) \prod_{j=1}^n f_{\mathbf{X}^{\mathrm{T}}\mathbf{A}_\gamma}(y_j)}\right\}.$$

Derivation of α follows directly from (10). similar moves are considered for the coordinates of γ as well. The move parameters, such as σ_{x1}, σ_{x2} etc., are tuned so that about 50% of the moves are accepted. This tuning can be achieved on a trial and error basis with multiple runs of the chain. One can also design an automatic and iterative tuning of these parameters until the burn-in stage.

Store: Store $f = f_{\mathbf{X}^{\mathrm{T}}\mathbf{A}_\gamma}$ from every k-th sweep of the MCMC sampler after an initial burn-in period.

The MCMC sample obtained as above can be summarized in different ways to estimate different features of the posterior. The most important of these is the predictive density which is estimated simply by taking an average of the stored f values.

Remark In order to compute $\prod_{j=1}^n f_{\mathbf{X}^{\mathrm{T}}\mathbf{A}_\gamma}(y_j)$, one needs to evaluate the process $\mathbf{X}^{\mathrm{T}}\mathbf{A}_\gamma(\cdot)$ at the points $y_1, \cdots, y_n$ and also needs to calculate $\int e^{\mathbf{X}^{\mathrm{T}}\mathbf{A}_\gamma(t)}dt$. For the latter term, one can evaluate the process on a plotting grid $G \subset [0, 1]$ and then carry out the integration numerically. This grid would be essentially the same as the grid on which f is evaluated and stored. For our computation we would take $G = \{0.00, 0.01, \cdots, 1.00\}$ as the plotting grid. The quality of the estimate does not improve much by taking a finer grid.

4.1 Choice of Nodes and Difficulties

The choice of nodes plays a crucial role in the above computation. A default strategy would be to take T equal to an uniform grid over $\mathcal{I}$ with a small

$\delta(T)$. For example, when $d = 1$, one can take $T = \{0, \frac{1}{m-1}, \cdots, \frac{m-2}{m-1}, 1\}$ with $\delta(T) = \frac{1}{2(m-1)}$. However, a default, uniform grid T, with small $\delta(T)$ may not always lead to a better computation. We discuss two of the major difficulties below.

First, the computation time increases substantially. The time Algorithm I takes to run can be expressed as $a + (b + cmp(m))M$, where m is the size of T, $p(m)$ is the computation time for calculating $A^{-1/2}$ of a square matrix A of dimension $m \times m$, M is the total number of iterations and a, b and c are constants which depend on the dimension d and the sample size n. For most software packages $p(m)$ is a polynomial of degree 2 or 3. Therefore the time required increases polynomially with the number of nodes used in the algorithm. Now, a finer choice of the grid would result in an increased value of m and consequently an increased running time for the algorithm. The situation would be worse for higher dimensions where a k-fold reduction in $\delta(T)$ would amount to at least a $\approx k^{3d}$ fold increase in m.

The second problem relates to the quality of estimation. Note that Algorithm I requires a numerical computation of $\mathbf{\Sigma}_\gamma^{-1/2}$. However, it would be infeasible to carry out this computation if β is smaller than some number L_T which would depend on T as well as the software used for computing. For example, `MATLAB 6.5.1` may calculate $\mathbf{\Sigma}_\gamma^{-1/2}$ inaccurately if β is smaller than $L_T \approx 1.15$ when $T = \{0, .1, \cdots, .9, 1\}$. Therefore, it would be necessary to use a truncated version H_T of H that ensures $\Pr(\beta > L_T) = 1$. But, for a default grid T, L_T would be quite large when $\delta(T)$ is small. This would lead to overfitting and result in a very inaccurate approximation to the exact posterior.

Remark A conservative bound of L_T is obtained as the minimum β for which the conditioning number (ratio of the largest to the smallest eigen value) of $\Sigma^{-1/2}$ is smaller than the inverse of the machine accuracy (i.e., relative rounding off error, generally $\approx 10^{-6}$). This minimum value can be quickly found by searching over a grid of β values. For higher dimensions we recommend searching over vectors of the form $\beta = (b, \cdots, b)$.

4.2 Data driven Selection of Nodes

The above discussion points out that it is impractical to take T simply as a default grid with a small $\delta(T)$. On the other hand, a default T with a large $\delta(T)$ is also unappealing, since it may result in a poor approximation. A possible way out is to use a small, *customized* grid T which is tight at the *right* places but sparse everywhere else. One can think of these *right* places as the regions where the true density f undergoes rapid changes. Such a customized grid can be derived only in a data driven manner, since the right places are unknown *a priori*. We achieve this by treating T as an unknown parameter and let the data select a good set of Ts. The exact model is described below.

Fix a default, tight grid of nodes S of size k. For each nonempty subset $T \subset S$, let Π^*_T denote the distribution of the imputed process $f_{\mathbf{X}^{\mathrm{T}}\mathbf{A}_\gamma}$ where $(\mathbf{X}, \gamma) \sim N_{|T|}(\mathbf{0}, \mathbf{I}_{|T|}) \times H_T$ for some distribution H_T that ensures $\Pr(\beta > L_T) = 1$ and with A_γ constructed by using only the nodes in T. Instead of working with any single Π^*_T, we combine all of these into the following mixture:

$$\Pi_m = \sum_{T \subset S, T \neq \emptyset} \omega_T \Pi^*_T$$

where $\boldsymbol{\omega} = (\omega_T : T \subset S, T \neq \emptyset)$ form a system of weights: $\omega_T \geq 0$ and $\sum_T \omega_T = 1$. One may select ω_T according to the (rescaled) binomial probabilities:

$$\omega_T \propto p^{|T|}(1-p)^{k-|T|}$$

where p is some fixed number in $(0, 1)$. A practical choice of p can be made ensuring that the expected number of nodes, kp, be small.

Under this mixture prior, the model on the unknown density f can be described as: $f = f_{\mathbf{X}^{\mathrm{T}} A_\gamma}$ where,

$$(\mathbf{X}, \gamma) \mid T \sim N_{|T|}(\mathbf{0}, \mathbf{I}_{|T|}) \times H_T,\ T \sim w_T.$$

One should write A_γ as $A_{\gamma,T}$, since its construction is specific to the subset T, but we shall drop this extra subscript to keep the notation simple.

One can draw samples of $(\mathbf{X}, \gamma)$ from its posterior by using an MCMC sampler. However, now we need to update the set T in every sweep of the MCMC. Such an update would alter the dimension of the parameter space whenever the size of T changes. Below we present an algorithm for an MCMC sampler which copes with this change in dimension through a reversible jump move.

Algorithm II:

Initialize Begin with some subset $T = \{t_1, \cdots, t_m\}$. Generate $\mathbf{X} \sim N_m(\mathbf{0}, \mathbf{I}_m)$ and $\gamma \sim H_T$.

Update Update $\mathbf{X}$ and γ, keeping the nodes fixed, as in Algorithm I.

Update nodes Propose a birth or a death or a shuffle according to some probability vector $(b_m, d_m, s_m = 1 - b_m - d_m)$. In case of birth, propose to increase T by attaching one more node to it. One also needs to expand $\mathbf{X}$ suitably. For death, remove a member of T and accordingly change $\mathbf{X}$. For shuffle, simply replace a member of T with an outside node. The vector $\mathbf{X}$ remains fixed in this case. The proposals for these three moves are given below:

Birth: Propose $T' = T \cup \{t_{m+1}\}$ and $\mathbf{X}' = (\mathbf{X}^{\mathrm{T}}, x_{m+1})^{\mathrm{T}}$ where t_{m+1} is chosen randomly from $S \setminus T$ and $x_{m+1} \sim N(0, \sigma_B^2)$.

Death: Propose $T' = T \backslash \{t_i\}$ and $\mathbf{X}' = \mathbf{X}_{-i} = (x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_m)^{\mathrm{T}}$ where t_i is chosen randomly from T.

Shuffle: Propose $T' = T \cup \{t_{m+1}\} \setminus \{t_i\}$ and $\mathbf{X}' = \mathbf{X}$ where t_{m+1} is chosen randomly from $S \setminus T$ and t_i is chosen randomly from T.

Acceptance: One can easily calculate the acceptance probability for each of the above moves by following the principle of detailed balance discussed in Richardson and Green (1997). For example, once a birth move is proposed, along with T' and $\mathbf{X}'$, it is accepted with the probability

$$\alpha = \min \left\{ 1, \frac{d_{m+1} \frac{1}{m+1} \phi(x_{m+1}) p \prod_{j=1}^{n} f_{\mathbf{X}'^{\mathrm{T}} \mathbf{A}'_{\gamma}}(y_j) h_{T'}(\gamma)}{b_m \frac{1}{k-m} \phi(\frac{x_{m+1}}{\sigma_B})(1-p) \prod_{j=1}^{n} f_{\mathbf{X}^{\mathrm{T}} \mathbf{A}_{\gamma}}(y_j) h_T(\gamma)} \right\}$$

where $\mathbf{A}'_\gamma$ corresponds to T' and h_T is the density function of H_T. When H_T is derived from H by truncating the support of β to $\{\beta > L_T\}$, the ratio $\frac{h_{T'}(\gamma)}{h_T(\gamma)}$ simplifies to $I(\beta > L_{T'})\frac{c_T}{c_{T'}}$ where $c_T = \Pr(\beta > L_T)$ under the distribution H.

Store: Store $f = f_{\mathbf{X}^{\mathrm{T}}\mathbf{A}_\gamma}$ as in Algorithm I. As before, the estimate of the predictive density is obtained by averaging the stored values of f.

The birth, death or shuffle probabilities are user specified. We recommend the following choices:

$$\begin{aligned} s_m &= c f_{Beta}\left(l(m); (1-p)^{-1}, p^{-1}\right) \\ b_m &= (1-s_m)(1-l(m)^a) \\ d_m &= (1-s_m)l(m)^a \end{aligned}$$

where $l(m) = \frac{m-1}{k-1}$ and $f_{Beta}(x; b_1, b_2)$ denotes the Beta density with parameters b_1 and b_2, evaluated at x. The constants c and a are chosen so that at $m = \lceil kp \rceil$, $s_m = b_m = d_m = \frac{1}{3}$. With these choices, a birth is encouraged when T is small, a death is encouraged when T is large and a shuffle is given an equal preference to a birth or a death around the target size $\lceil kp \rceil$. A suitable combination of a death and a birth is actually equivalent to a shuffle. But we feel that a direct shuffle move for moderately sized $|T|$s make it more efficient to explore the important regions where a tight set of nodes is required. If p is too close to 0 or 1, one may replace p by some moderately valued $\tilde{p}$ in the definition of s_m. We use $\tilde{p} = 0.3$ if $p < 0.3$ and $\tilde{p} = 0.7$ if $p > 0.7$.

5 An Example

We illustrate the proposed computing scheme with a simulation. Following Lenk (1991), we choose $\mathcal{I} = [0, 1]$ and draw 50 independent observations from the true density

$$f_0(t) \propto \frac{3}{4} 3e^{-3t} + \frac{1}{4}\sqrt{32/\pi} e^{-32(t-0.75)^2}. \tag{11}$$

This density is a mixture of an exponential and a normal truncated to the interval $\mathcal{I}$. As Lenk points out, this density has a peak at the boundary and a small bump in the interior. These two features make its estimation quite challenging.

To specify the prior, we use $\sigma_0(s, t) = \exp(-(s - t)^2)$ which results in infinitely differentiable sample paths of f_W. The prior H on $\gamma = (\tau, \beta)$ is chosen as follows: Under H,

$$\tau^2 \sim Gamma(r = 5, \lambda = 4)$$

$$\beta \sim EV(r = 3, \lambda = \sqrt{10})$$

and these are taken to be independent. Here, $Gamma(r, \lambda)$ denotes the gamma distribution with shape r and scale λ. The notation $EV(r, \lambda)$ stands for a type of extreme value distribution on the positive real line with its density function given by,

$$f(\beta) \propto I(\beta > 0)\frac{1}{\lambda^r}\beta^{r-1}e^{1+\frac{\beta}{\lambda}-e^{\beta/\lambda}}.$$

A special feature of this type of extreme value distribution is that it has a very rapidly decaying tail: $\Pr(\beta > b) \leq c_1 e^{-e^{c_2\beta}}$ for some constants c_1, c_2.

The exponential tail on τ and the extreme value distribution type tail on β are motivated by the theoretical study on consistency properties of a logistic Gaussian process prior. The specific choices made above are recommended by Ghosal and Roy (2006, see Section 2.4). For a finite sample, the need of such rapidly decaying tails can be understood from a different perspective. The exponential tails strongly discourage large values of τ and β and hence act as strong penalties against undersmoothing and overfitting. It is also important that the shape parameters for both the gamma prior on τ^2 and the extreme value prior on β be moderately large. Such a choice makes sure that the logistic Gaussian process prior does not concentrate too much around its prior guess - the uniform density. The choices made above work fairly well for many different applications. However, the resulting estimates are not alarmingly sensitive to the exact choice.

Figure 1 shows an estimate (solid line) produced by Algorithm II with $S = \{0, .1, \cdots, 1\}$, $p = 0.5$. We executed 20,000 sweeps of the MCMC sampler, from which every 20th sample is stored after an initial burn-in of 10,000 sweeps. This took about 72 seconds to run. The histogram of the data is also plotted.

5.1 Comparison with Exact Posterior

It would be interesting to compare our estimate with an estimate obtained from the exact posterior. But this requires computation of the Bayes estimate starting with the actual prior. Fortunately, for the σ_0 used in our example, it is feasible to derive this estimate through a large scale importance sampling.

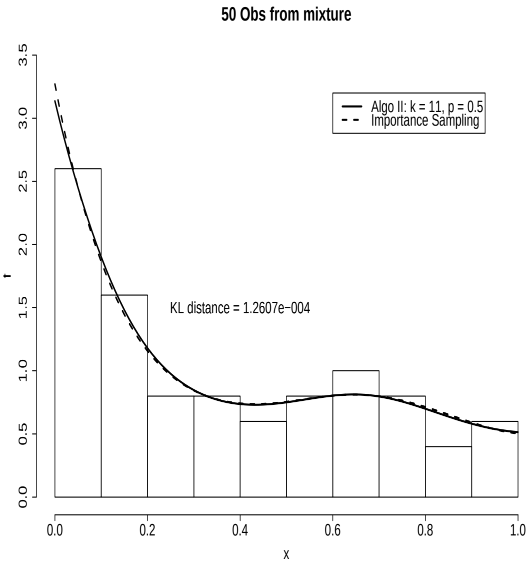

Figure 1: Bayes estimates obtained with Algorithm II: $S = \{0, .1, \cdots, 1\}$, $p = 0.5$ (solid line) and from importance sampling (broken line). The estimates agree well with each other with a Kullback-Leibler divergence $= 0.00012$.

We obtain an importance sampling estimate of the actual Bayes estimate by generating a large sample of W (actually a finite dimensional profile on the plotting grid G that was used for Algorithm II) from the prior Π and then taking an weighted average of the resulting f_W. The likelihood function is used as the weight. To simulate W, we make use of a Cholesky decomposition of the covariance matrix which can be derived in a closed form for the particular choice of σ_0 used in our example. We superimpose the graph of this estimate (broken line) on Figure 1 which also shows the imputation based estimate obtained earlier. Evidently, these two estimates agree really well with each other, with a small Kullback-Leibler divergence of 0.00012. This ensures that one can indeed obtain good approximations by using the surrogate prior through Algorithm II.

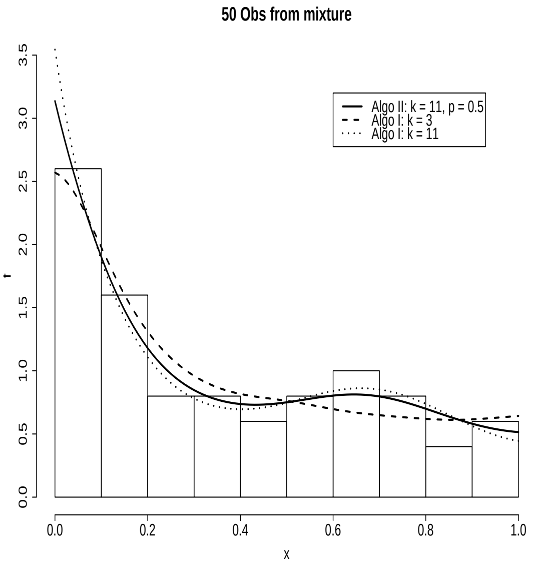

Figure 2: Computation with Algorithm II: $S = \{0, .1, \cdots, 1\}$, $p = 0.5$ (solid line), with Algorithm I: $T = \{0.0, 0.5, 1.0\}$ (broken line) and Algorithm I: $T = S$ (dotted line).

5.2 Comparison with Algorithm I

In Figure 2, we show two estimates obtained from Algorithm I along with the estimate produced by Algorithm II (solid line). One estimate (dashed line) from Algorithm I is obtained with $T = \{0.0, 0.5, 1.0\}$ and the other (dotted line) is obtained with $T = S$. Note that the latter estimate slightly overfits the data compared to the estimate obtained with Algorithm II. This is an artifact of the severe left truncation ($L_T \approx 1.15$) on the prior of β that was necessary for computing this estimate.

5.3 Convergence Diagnosis

In this section we briefly overview a possible diagnosis of the convergence of our reversible jump MCMC (rj-MCMC) sampler described by Algorithm II. Convergence diagnosis of rj-MCMC sampler based on parallel chains have been

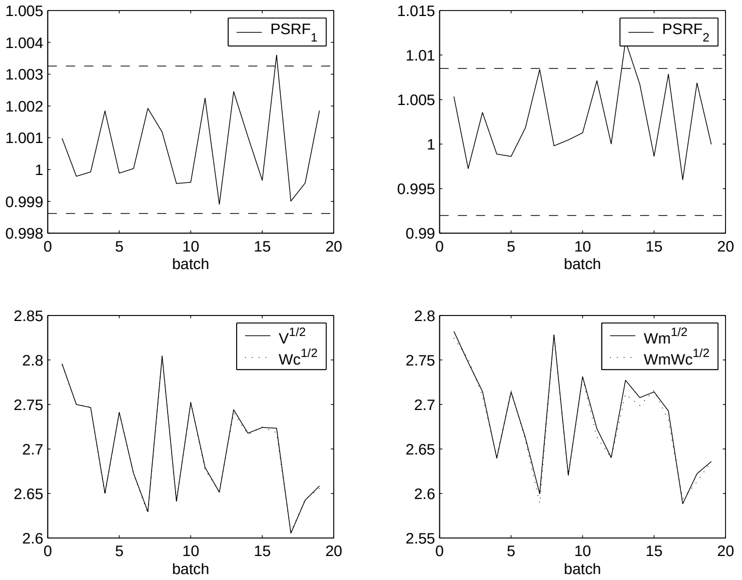

Figure 3: Convergence diagnosis of the sampler of Algorithm II. Left: ratio ($PSRF_1$) of total variance (Vhat) to within chain variance (Wm). Right: ratio ($PSRF_2$) of within model variance (Wm) to within model within chain (WmWc) variance. Both ratios stay close to 1 - indicating convergence.

proposed and discussed in Brooks and Giudici (2000) (see also Gelman and Rubin (1992) for the basic ideas). An improved version appears in Castelloe and Zimmerman (2002) to handle the *unbalanced* case where some components of the parameter space are less visited than the others. We find this to be appealing for our sampler, especially in higher dimensions, where models corresponding to large $T \subset S$ are less likely to be visited.

To test for convergence of our rj-MCMC sampler (for the dataset used in the previous example), we obtain 5 chains of MCMC samples obtained from 5 parallel runs of the sampler. Each chain contains 9,500 samples obtained from 200,000 sweeps of the sampler with every 20th sweep stored after a burn in of 10,000 sweeps. Each chain is split into 19 batches that consist of 500 successive samples. For each sample, we compute deviance = two times the negative log likelihood. We compute two PSRF (potential scale reduction factor) statistics using deviance for each of the 19 batches. The top panel of Figure 3 shows the plots of these two statistics across the batches. The lower panel shows the plots of the two pairs of statistics that are used for computing the PSRFs; see Brooks and Giudici (2000) and Castelloe and Zimmerman (2002) for the definitions of all these quantities.

It is apparent that the two PSRF graphs remain very close to the target value 1. For each plot on the top panel, two broken, horizontal lines show estimated 2.5% and 97.5% percentiles of the corresponding PSRF statistics under the assumption of convergence and mixing. These estimates are obtained through a bootstrap Monte Carlo with 10,000 bootstrap samples drawn from the pool of all chains combined. These plots give clear indication that the proposed sampler indeed achieves rapid mixing and converges. Similar results are obtained if one monitors the parameters β or τ.

6 Densities with Unbounded Support

So far, the logistic Gaussian process f_W (or f_Z) has been defined only on bounded intervals. To define such processes on $\mathbb{R}^d$ (or some other unbounded interval), one can make use of a suitable transformation of an f_W defined on $\mathcal{I}$ as in (1). A more flexible approach is to use a parametric family of transformations with a suitable prior on the parameters. Verdinelli and Wasserman (1998) and Lenk (2003) use this approach for the case $d = 1$ and successfully compute the posterior using orthogonal series expansion. This can be done with our imputation based approach as well. An extension to the case $d > 1$ is also possible. We briefly discuss the necessary constructions before taking up the issue of computation.

6.1 When Support is $\mathbb{R}$

Let $g_\theta(\cdot)$ be a family of continuous, positive valued density functions on $\mathbb{R}$ with a parameter $\theta \in \Theta$, some Euclidean space. For example, one can take $g_\theta = N(\mu, \xi^2)$ with $\theta = (\mu, \xi^2) \in \mathbb{R} \times \mathbb{R}^+$. Let G_θ denote the CDF of g_θ. Define a new process f_W^θ on $\mathbb{R}$ using a transformation T_θ of f_W as,

$$f_W^\theta(y) = T_\theta f_W(y) = f_W(G_\theta(y))g_\theta(y),\ y \in \mathbb{R}, \tag{12}$$

where f_W is a logistic Gaussian process on $[0,1]$ and $\theta \sim Q$ independently of W. The process f_W^θ realizes its sample paths in the space of densities over $\mathbb{R}$, inducing a prior distribution $\tilde{\Pi}$ on this space.

Remark A little algebraic simplification shows that f_W^θ can also be expressed as,

$$f_W^\theta(y) = \frac{e^{\mu_\theta(y) + \tilde{W}(y)}}{\int_{\mathbb{R}} e^{\mu_\theta(t) + \tilde{W}(t)} dt}$$

where $\mu_\theta(\cdot) = \log g_\theta(\cdot)$ and $\tilde{W}$, given (θ, γ), is a mean zero Gaussian process with covariance $\sigma_\gamma(G_\theta(\cdot), G_\theta(\cdot))$. Therefore, f_W^θ itself is a logistic Gaussian process. The indirect construction simply avoids the technical difficulty in showing that the integral in the denominator is finite almost surely. Note that this construction also matches with the models proposed by Wahba (1978) in the context of penalized likelihood estimation.

Remark The use of a family of transformations $T_\theta : f_W \mapsto f_W^\theta$ results in a semi-parametric model, where the nonparametric part builds around the parametric *prior guess* $\{g_\theta\}$. To justify the last statement, note that, $E(\log f_W^\theta | \gamma, \theta) = \mu_\theta + \mathit{const.}$, whose logistic transform is nothing but g_θ. The variance of the process W controls the weight the prior places on neighborhoods of the parametric guess. An excellent control on this is offered by the parameter $\gamma = (\tau, \beta)$ that appears in the particular specification of σ_γ outlined in (2).

6.2 When support is $\mathbb{R}^d$

As before, we take g_θ to be a family of continuous, positive density functions on $\mathbb{R}^d$. But, for $d > 1$, a little more work is needed to get G_θ. Fix a θ and let $Y = (Y_1 \cdots, Y_d) \sim g_\theta(\cdot)$. For any $y \in \mathbb{R}^d$, define

$$G_\theta(y) = (G_{1,\theta}(y_1), G_{2,\theta}(y_1, y_2) \cdots, G_{d,\theta}(y_1, \cdots, y_d)) \tag{13}$$

where,

$$\begin{aligned} G_{1,\theta}(y_1) &= P(Y_1 \leq y_1) \\ G_{2,\theta}(y_1, y_2) &= P(Y_2 \leq y_2 | Y_1 = y_1) \\ \vdots \quad & \quad \vdots \\ G_{d,\theta}(y_1, \cdots, y_d) &= P(Y_d \leq y_d | Y_1 = y_1, \cdots, Y_{j-1} = y_{j-1}). \end{aligned}$$

Note that the function G_θ maps $\mathbb{R}^d$ onto $(0,1)^d$ with its Jacobian determinant given by g_θ. As before, we define f_W^θ as,

$$f_W^\theta = T_\theta f_W(y) = f_W(G_\theta(y)) g_\theta(y),\ y \in \mathbb{R}^d, \tag{14}$$

where f_W is a logistic Gaussian process on $[0,1]^d$ and $\theta \sim Q$ independently of W. The distribution $\tilde{\Pi}$ of the process f_W^θ defines a logistic Gaussian process prior on the space of densities supported on $\mathbb{R}^d$.

6.3 Choice of g_θ

One can show that G_θ defined in (13) admits a differentiable inverse under strict positivity and continuity conditions on g_θ. This renders T_θ invertible on the space of densities over $\mathbb{R}^d$ which satisfy a tail condition with respect to the family g_θ. It can be shown that all these densities belong to the Kullback-Leibler support of $\tilde{\Pi}$ and hence weak posterior consistency would be achieved at these densities. Strong posterior consistency can also be achieved for this class of densities under some regularity conditions on the family $\{g_\theta\}$.

The above discussion suggests that a choice of g_θ can be obtained from our prior beliefs about the tails of the unknown density. For example, if one anticipates the target density to be a location mixture of normals, or a location-scale mixture of normals with a bounded support on the scale mixing, then it is safe to choose g_θ as the normal density. However, if it is suspected that the unknown density could have tails heavier than any normal density, then one needs to take g_θ with heavy tails as well. In this case, one could take g_θ as the family of t densities.

The prior Q on θ depends on the family g_θ. One could take Q same as what would have been selected for a parametric analysis with g_θ. For example, the normal-inverse Gamma prior is a convenient choice when g_θ is the normal density. However, the conjugacy property of the parametric analysis has no positive or negative influence on our computation.

6.4 Computation of Posterior

Suppose a sample $\mathbf{y} = (y_1, \cdots, y_n)$ has been obtained from a density f which is to be modeled semiparametrically as $f = f_W^\theta$. Like before, we instead use the imputation based surrogate model: $f = f^\theta_{\mathbf{X}^{\mathrm{T}}\mathbf{A}_\gamma} = T_\theta f_{\mathbf{X}^{\mathrm{T}}\mathbf{A}_\gamma}$ where,

$$(\mathbf{X}, \gamma, \theta) \mid T \sim N_{|T|}(\mathbf{0}, \mathbf{I}_{|T|}) \times H_T \times Q, \ T \sim \boldsymbol{\omega}.$$

We would draw samples of $(\mathbf{X}, \gamma, \theta)$ from its posterior distribution given $\mathbf{y}$ by using a rj-MCMC sampler. The exact steps are given by the following algorithm.

Algorithm III:

Initialize Begin with some subset $T = \{t_1, \cdots, t_m\}$. Generate $\mathbf{X} \sim N_m(\mathbf{0}, \mathbf{I}_m)$, $\gamma \sim H_T$ and $\theta \sim Q$.

Update Update X and γ, keeping T and θ fixed. These updates are to be done exactly as in algorithm I but by treating $\mathbf{y}^\theta = (G_\theta(y_1), \cdots, G_\theta(y_n))^{\mathrm{T}}$ as the observed data.

Update nodes Update T as in Algorithm II by treating $\mathbf{y}^\theta$ as the observed data.

Update θ Update θ coordinatewise. For example, suppose we want to update the first coordinate of $\theta = (\theta_1, \theta_{-1})$. We propose $\theta' = (\theta_1', \theta_{-1})$ where $\theta_1' \sim N(\theta_1, \sigma_{\theta 1}^2)$ and accept this move with a probability:

$$\alpha = \min\left\{1, \frac{q(\theta') \prod_{j=1}^n [f_{\mathbf{X}^{\mathrm{T}}\mathbf{A}_\gamma}(G_\theta'(y_j)) g_\theta'(y_j)]}{q(\theta) \prod_{j=1}^n [f_{\mathbf{X}^{\mathrm{T}}\mathbf{A}_\gamma}(G_\theta(y_j)) g_\theta(y_j)]}\right\}$$

where q denotes the density of Q.

Store: Store $f^\theta_{\mathbf{X}^{\mathrm{T}}\mathbf{A}_\gamma}$ from every k-th sweep after an initial burn in of b sweeps.

6.5 Examples

We illustrate the proposed method of computation with two examples. These examples cover the cases $d = 1$ and $d = 2$ with data arising from a mixture of univariate normals and a mixture of bivariate normals.

Example 1 (Mixture of univariate normals): Mixture densities are standard test examples for nonparametric methods. Our first example includes a sample of 200 independent observations from the normal mixture: $f_0 = .4N(-3, 1.5^2) + .6N(2, 1^2)$.

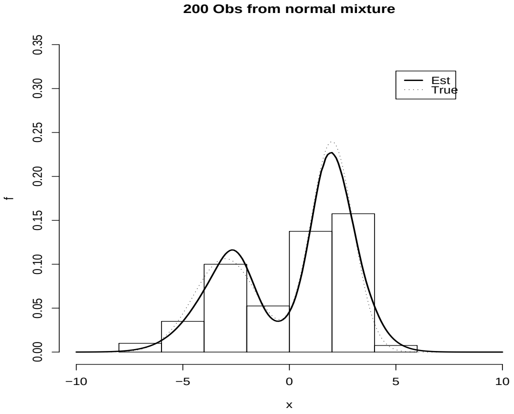

Figure 4: Predictive density (solid line) estimated from a sample of 200 observations from the true density (dotted line) $.4N(-3, 1.5^2) + .6N(2, 1^2)$. Estimate obtained using Algorithm III with $S = \{0, .1, \cdots, 1\}$ and $p = 0.5$. The normal family is used as g_θ. The L_1 distance between the true and the estimated density is 0.1015.

The family g_θ is taken to be the normal family $N(\mu, \xi^2)$ with $\theta = (\mu, \xi^2)$. The distribution Q of θ is given as:

$$\mu|\xi \sim N(0, \xi^2) \text{ and } \xi^{-2} \sim Gamma(r = 5, \lambda = 4).$$

Recall that this is a conjugate prior for the normal family in the parametric setting. Other priors could also be used.

We take $f_{\mathbf{X}^{\mathrm{T}}\mathbf{A}_\gamma}$ exactly as in the example of Section 5. Figure 4 shows the plot of the estimated predictive density (solid line) and the plot of the true density f_0 (dotted line). The estimate is obtained using 20,000 sweeps of the MCMC with every 20th sample stored after a burn-in of 10,000 sweeps. It takes about 155 seconds.

Example 2 (A skewed, infinite mixture of normals): Many Bayesian and non-Bayesian kernel based methods are specifically designed to estimate mixture densities well. These methods, however, may do poorly for small sam-

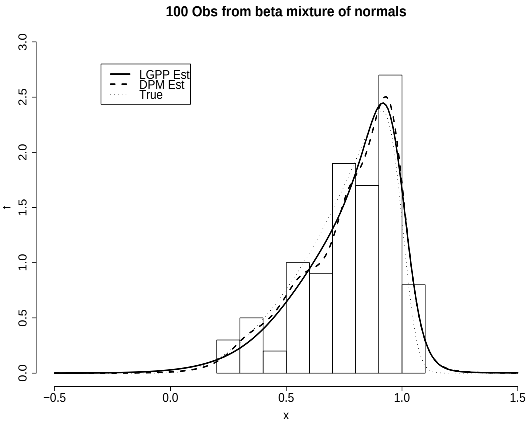

Figure 5: A comparison between the estimates obtained from logistic Gaussian process prior (Algorithm III) and Dirichlet mixture prior (see text for details). The latter finds it difficult to capture mixture densities that are difficult to approximate with a finite mixture involving only a few components.

ples when the true density involves an infinite mixture which cannot be well approximated by a finite mixture with a few components. On the other hand, we believe that a logistic Gaussian process prior has more shape flexibility to capture these mixtures even with moderate sample sizes. In this example, we illustrate this within a simulation setting, where we obtain 100 observations from the mixture density $\int_0^1 N(\mu, 0.05^2)(\mu^2/3)d\mu$ – a location mixture of normals, mixed according to the $Beta(3,1)$ distribution. This density (dotted line) and histogram of the observations are shown in Figure 5. The solid line in this figure gives the estimated predictive density obtained using our method where the prior is specified exactly as in the previous example.

The dashed line on Figure 5 shows the predicite density estimated by a Dirichlet location scale mixture of normal model. We use the model specified in Escobar and West (1995), namely, the observations y_i's are deemed as independent draws from $N(\mu_i, \sigma_i^2)$ and a hierercchical prior is placed on (μ_i, σ_i^2) as follows. We consider, a priori, (μ_i, σ_i^2) are independent draws from a random measure G arising from $Dir(\alpha G_0)$ - a Dirichlet process prior with precision constant α and base probability measure G_0 on the space of (μ, σ^2). The base measure is taken as follows: under G_0, $(\sigma^2)^{-1} \sim Gamma(s/2, S/2)$ and $\mu|\sigma^2 \sim N(m, \tau\sigma^2)$. We further take $\tau^{-1} \sim Gamma(w/2, W/2)$ and $m \sim N(a, A)$. We fix $\alpha = 1$ - a conventional choice. We take $s = 2$ and $S = 0.01$ so that the distribution of σ under G_0 has the central 95% coverage interval $= [0.037, 0.444]$ with median $= 0.085$. The parameter A is fixed at ∞, for which a can be fixed at any arbitrary value – we choose 0. We take $w = 1$ and $W = 1$ so that the central 95% coverage interval of τ is $[0.199, 1018.258]$ with median $= 2.198$. The last choice is motivated by consideration of the *modality issue* discussed in Escobar and West (1995). In particular, with $\tau = 2.0$ and $\sigma = 0.085$, the prior strongly supports a unimodal predictive density (with 100 observations) - a priori, there is a 81.2% chance of a single mode and a 99.7% chance of atmost two modes. Note that, these choices of the hyperparameters are well matched with the true density which we use to generate the observations.

We learn from our simulations that despite the favorable choice of the hyperparameters, the Dirichlet process mixture tends to slightly undersmooth (see Figure 5). In fact on repeating the above experiment on 50 independently generated data sets, we find that about 70% of the times the estimate obtained from the logistic Gaussian process prior has a smaller L_1 error compared to the estimate obtained from the Dirichlet mixture prior. In all these cases, the former produces a smoother estimate than the latter.

We also note that the kernel based method used by the `KernSmooth` package of the software `R` produces estimates inferior to both the Bayesian procedures discussed above - more often than not it finds multiple modes. We omit this estimate from Figure 5 for purpose of clarity.

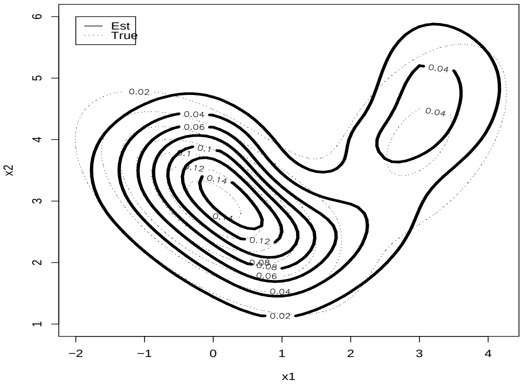

Figure 6: Level curves (solid line) of the estimated predictive density from a sample of 200 observations from a mixture of bivariate normals distribution. The level curves of the true density is shown in the background as the thin dotted lines. Estimate obtained with Algorithm III with $S = \{0, .1, \cdots, 1\}^2$ and $p = 0.1$. The bivariate normal family is used as g_θ. The L_1 distance between the true and the estimated density is 0.2265.

Example 2 (Mixture of bivariate normals): Next we consider the situation $d = 2$. A sample of 200 independent observations is generated from the mixture density

$$f_0 = 0.3N\left(\begin{pmatrix} 3 \\ 4 \end{pmatrix}, \begin{bmatrix} 1 & .6 \\ .6 & 1.5 \end{bmatrix}\right) + 0.7N\left(\begin{pmatrix} 0 \\ 3 \end{pmatrix}, \begin{bmatrix} 1 & -.5 \\ -.5 & .8 \end{bmatrix}\right) \quad (15)$$

The family g_θ is taken to be the bivariate normal family. We use a slightly different formulation to suit our computations best. Take $\theta = (\mu_1, \xi_1^2, \alpha_0, \alpha_1, \xi_2^2)$ and let g_θ stand for the joint distribution of $Y = (Y_1, Y_2)$ given by,

$$Y_1 \sim N(\mu_1, \xi_1^2),\ Y_2 | Y_1 = y_1 \sim N(\alpha_0 + \alpha_1 y_1, \xi_2^2) \quad (16)$$

The distribution Q of θ is taken as follows,

$$\mu_1|\xi_1 \sim N(0,\xi_1^2), \xi_1^{-2} \sim Gamma(r=5,\lambda=4),$$

$$\alpha_0,\alpha_1|\xi_2 \overset{ind}{\sim} N(0,\xi_2^2), \xi_2^{-2} \sim Gamma(r=5,\lambda=4)$$

and the two sets are independent.

The covariance function is chosen as, $\sigma_\gamma(s,t) = \tau^2 \exp(-\beta_1^2(s_1-t_1)^2-\beta_2^2(s_2-t_2)^2)$ with $\gamma=(\tau,\beta_1,\beta_2)$. The distribution H on γ is specified as the following: $\tau^2 \sim Gamma(r=5,\lambda=4)$, $\beta_1 \sim EV(r=3,\lambda=\sqrt{10})$, $\beta_2 \sim EV(r=3,\lambda=\sqrt{10})$ and these are independent. We use $S=\{0.0,0.1,\cdots,1.0\}^2$ and $p=0.1$. Note that $k=|S|=121$ in this case and hence a smaller p is needed to ensure that the average number of nodes used remains low.

Figure 6 shows the level plots (solid lines) of the estimated predictive density superimposed over the level plots (dotted line) of the true density f_0. Both contour plots are obtained by using levels $=\{0.00,0.02,\cdots,0.20\}$. The estimated predictive density is obtained by running Algorithm III for 20,000 sweeps where every 20th sweep was saved after a burn-in of 10,000 sweeps. The sampler took approximately 62 minutes to run. It is evident that the computation time increases substantially with an increase in the dimensionality.

6.6 Back to Bounded Support

Previously, we used a nonparametric model $f=f_{\mathbf{X}^{\mathrm{T}}\mathbf{A}_\gamma}$ when it was known that the support of f is $\mathcal{I}=[0,1]$. However, it is quite possible to use a semiparametric model $f=f^\theta_{\mathbf{X}^{\mathrm{T}}\mathbf{A}_\gamma}$ by using a family $\{g_\theta\}$ with support $\mathcal{I}$. We recommend to use a semiparametric model over a nonparametric one even in the case of a bounded support. The reason for this is a particular advantage in computation gained by using the family g_θ. We elaborate on this below.

We would make a comparison between the estimates provided by the two models $f = f_{\mathbf{X}^{\mathrm{T}}\mathbf{A}_\gamma}$ and $f = f^{\theta}_{\mathbf{X}^{\mathrm{T}}\mathbf{A}_\gamma}$. To make a fair comparison, assume that the specification of $f_{\mathbf{X}^{\mathrm{T}}\mathbf{A}_\gamma}$ remains the same across the two models. This means, in particular, that both computations are to be carried out using the same grid of nodes S.

Now, suppose that the true density f has two sharp modes in an interval $(a, b) \subset (0, 1)$ which does not intersect with S. Although a sample of observations $\mathbf{y} = (y_1, \cdots, y_n)^{\mathrm{T}}$ from f would exhibit this bimodality in (a, b), the nonparametric model would fail to recognize it. The model $f = f_{\mathbf{X}^{\mathrm{T}}\mathbf{A}_\gamma}$ learns from the data through the subset of nodes T that is being used. But, no subset T of S can place a node inside (a, b), and hence cannot capture the rapid changes that take place inside this interval. As a result the MCMC would be driven to sample fs which would average out the features in that interval. The resulting estimate would be poor.

Now consider the semiparametric model $f = f^{\theta}_{\mathbf{X}^{\mathrm{T}}\mathbf{A}_\gamma}$. Here, one essentially models the density of the transformed observations $\mathbf{y}^{\theta} = (G_\theta(y_1), \cdots, G_\theta(y_n))^{\mathrm{T}}$ by $f_{\mathbf{X}^{\mathrm{T}}\mathbf{A}_\gamma}$. The original interval now gets transformed to $(G_\theta(a), G_\theta(b))$. If the family $\{g_\theta\}$ is sufficiently flexible, then it is possible that for some values of θ the interval $(G_\theta(a), G_\theta(b))$ would have a substantial overlap with some subsets T of S. The MCMC would eventually be driven to these pairs of θ and T and a good estimate should come out.

We illustrate the above heuristic argument with an example. We obtain 50 observations from $f = 0.5N(0.62, 0.005^2) + 0.5N(0.68, 0.005^2)$ and model it as $f = f_{\mathbf{X}^{\mathrm{T}}\mathbf{A}_\gamma}$ and $f = f^{\theta}_{\mathbf{X}^{\mathrm{T}}\mathbf{A}_\gamma}$ where $f_{\mathbf{X}^{\mathrm{T}}\mathbf{A}_\gamma}$ is exactly the same as in the example of Section 5 and $g_\theta = Beta(\alpha_1, \alpha_2)$ with $\theta = (\alpha_1, \alpha_2)$. We take $\theta \sim Q = \xi \times \xi$ where ξ admits a density proportional to $I(x > 0)\frac{x/\lambda}{1+(x/\lambda)^3}$, where λ is so chosen that the median of ξ is 1. Such a choice makes sure that both the nonparametric

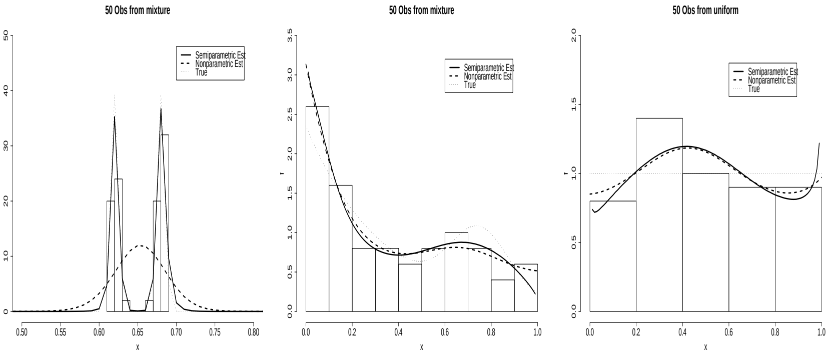

Figure 7: Comparison between the estimates from the nonparametric (dashed) and the semiparametric (solid) models for the case of bounded support ($\mathcal{I} = [0,1]$). Left: True density $= 0.5N(0.62, 0.005^2) + 0.5N(0.68, 0.005^2)$; Middle: true density $= f_0$ of example in Section 5; Right: true density $= U[0,1]$. Histogram of the data and the true densities (dotted line) are also shown. Semiparametric model does much better when the data are spiked (Left panel); otherwise, the two estimates are almost identical.

model and the semiparametric model have the same center, namely the uniform density. Note that the chosen $f_{\mathbf{X}^{\mathrm{T}}\mathbf{A}_\gamma}$ uses $S = \{0.0, 0.1, \cdots, 1.0\}$ and the true f has two sharp modes in the interval $(0.61, 0.69)$ which does not intersect with S.

The left plot of Figure 7 shows the two estimates obtained from these two models via Algorithm II and Algorithm III respectively. The estimate from the nonparametric model (broken line) completely misses the two modes of f. On the other hand the semiparametric estimate (solid line) captures the rapid changes. The histogram of the data and the true density (dotted line) are also produced.

What happens if f is such that the nonparametric model already provides a good estimate? In such cases the semiparametric estimate should be very close to the nonparametric estimate, since the latter is a particular case of the former. We illustrate this fact with two more simulations, one with the data set that was used in Section 5 and the other with a data set obtained from the uniform density on $[0, 1]$. The second and the third plots of Figure 7 show that the semiparametric (solid line) and the nonparametric (dashed line) estimates are almost the same. The only differences occur at the tails. This is unavoidable, since we have seen that the family $\{g_\theta\}$ has a strong influence on the tails of the estimates.

7 Discussion

We have proposed a new method to compute the posterior of a logistic Gaussian process prior for estimating densities. This method does not require expanding the underlying process as an orthogonal series and, therefore, can be applied to a large class of Gaussian processes. Our method of computation employs a surrogate prior which is an imputation based approximation of the actual prior. We illustrated through a theoretical result and simulations that the error of this approximation is not severe.

The speed and accuracy of the proposed algorithm depend on the choice of the nodes used for imputation. We have discussed a data driven selection of nodes by implementation of a reversible jump MCMC sampler. This is particularly helpful when data are multivariate, since any default, tight set of nodes would be very large in a high dimensional space.

The introduction of the semiparametric model further strengthens our computation by imparting a certain degree of mobility to the nodes. With this model, the data adjusts itself with respect to the nodes so that maximum information is learned. This way, one not only let the data select the number of nodes, but also their (relative) positions.

The computation time required by our method is within a reasonable limit, and is comparable to other Bayesian, nonparametric methods. Moreover, a nice feature of the proposed method is that the computation time does not increase rapidly with the sample size. For example, in simulation setting detailed in the Example of Section 5, the average computation time with 50 observations is 71 seconds, with 500 observations it is 84 seconds and with 5000 observations it is 215 seconds. This amounts to a roughly linear increase by 0.028 seconds per extra observation. Therefore a logistic Gaussian prior has an advantage over other nonparametric priors when large data sets are to be used.

Appendix: Proof

Proof of Theorem 3.1 Let $X(\cdot)$ denote the difference $W(\cdot)-Z(\cdot)$ with $\|X\| = \sup_t |X(t)|$ as its sup-norm. A simple calculation produces,

$$\sup_{t\in I} \frac{\hat{f}_W(t)}{\hat{f}_Z(t)} \le \frac{\sup_\gamma E(e^{2(n+1)\|X\|} \mid \gamma)}{\inf_\gamma E(e^{-2n\|X\|} \mid \gamma)}. \tag{17}$$

Therefore it suffices to show that the upper bound in the above display tends to 1 as $\delta(T)$ diminishes.

Note that, for each γ, the process $X(\cdot)$ given γ is a zero mean Gaussian process. The assumption of this theorem then leads to the conclusion that, $\exists \lambda > 0$ such that for all $\gamma \in \text{support}(H)$,

$$\Pr(\|X\| > x|\gamma) \le \exp(-\lambda x^2/\delta(T)).$$

This sub-Gaussian tail behavior is a common feature of the supremum of a Gaussian process. The above result can be proved rigorously starting from Corollary 2.2.8 of van der Vaart and Wellner (1996) using the Orlicz norm with respect to the function $\psi(x) = e^{x^2}/5$. For more general theory see Adler (1990).

Using the above property one can find $a, b > 0$, such that for all $\gamma \in$

support(H),

$$\begin{aligned} E(e^{2(n+1)\|X\|} \mid \gamma) &= 1 + 2(n+1)\int_0^\infty e^{2(n+1)x}\,\Pr(\|X\| > x|\gamma)dx \\ &\leq 1 + a\sqrt{\delta(T)}e^{b\delta(T)} \end{aligned}$$

and

$$\begin{aligned} E(e^{-2n\|X\|} \mid \gamma) &\geq E(I\{\|X\| < \delta(T)^{1/4}e^{-2n\|X\|} \mid \gamma) \\ &\geq e^{-2n\delta(T)^{1/4}}(1 - e^{-\lambda/\sqrt{\delta(T)}}). \end{aligned}$$

This completes the proof as both these bounds tend to 1 as $\delta(T) \to 0$.

Acknowledgments

We thank Prof. Jayanta K. Ghosh for many helpful suggestions which led to a better exposition of this work. We also thank an Associate Editor and three referees for their many constructive suggestions on an earlier version of the paper.

References

Adler, R. J. (1990), “An Introduction to Continuity, Extrema, and Related Topics for General Gaussian Processes”, *IMS Lecture Notes-Monograph Series.*

Brooks, S. P. and Giudici P. (2000), “MCMC Convergence Assessment via Two-way ANOVA”, *Journal of Computational and Graphical Statistics*, 9, 266-285.

Castelloe, J. M. and Zimmerman, D. L. (2002), "Convergence Assessment for Reversible Jump MCMC Samplers", *Department of Statistics and Actuarial Science, University of Iowa*, Technical Report #313.

Escobar, M. and West, M. (1995), "Bayesian density estimation and inference using mixtures". *Journal of American Statistical Assocication* 90, 577-588.

Gelman, A. and Rubin, D. B. (1992), "Inference from Iterative Simulations using Multiple Sequences", *Statistical Science*, 7, 457-511.

Ghosal, S. and Roy, A. (2006), "Posterior Consistency of Gaussian Process Prior for Nonparametric Binary Regression", *Annals of Statistics*, To appear.

Lenk, P. J. (1988), "The Logistic Normal Distribution for Bayesian, Nonparametric, Predictive Densities", *Journal of American Statistical Association*, 83(402), 509-516.

Lenk, P. J. (1991), "Towards a Practicable Bayesian Nonparametric Density Estimator", *Biometrika*, 78(3), 531-543.

Lenk, P. J. (2003), "Bayesian Semiparametric Density Estimation and Model Verification Using a Logistic Gaussian Process", *Journal of Computational and Graphical Statistics*, 12(3), 548-565.

Leonard, T. (1978), "Density Estimation, Stochastic Processes, and Prior Information", *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 40(2), 113-146.

Richardson, S. and Green, P. J. (1997), "On Bayesian Analysis of Mixtures with an Unknown Number of Components", *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(4), 731-792.

Tokdar, S. T. and Ghosh, J. K. (2006), "Posterior Consistency of Logistic Gaussian Process Priors in Density Estimation". *Journal of Statistical Planning* *and Inference*, To appear.

Van der Vaart, A. W. and Wellner, J. A. (1996), “Weak Convergence and Empirical Processes”, Springer-Verlag, New York.

Verdinelli, I. and Wasserman, L. (1998), “Bayesian Goodness of Fit Testing using Infinite Dimensional Exponential Families”, *Annals of Statistics*, 20, 1203-1221.

Wahba, G., (1978). “Improper Priors, Spline Smoothing and the Problem of Guarding Against Model Errors in Regression”, *Journal of Royal Statistical Society: Series B (Statistical Methodology)*, 40(3), 364-372.