Statistical Science
2011, Vol. 26, No. 1, 130–149
DOI: 10.1214/11-STS354

Variable Selection for Nonparametric Gaussian Process Priors: Models and Computational Strategies

Terrance Savitsky, Marina Vannucci and Naijun Sha

Abstract. This paper presents a unified treatment of Gaussian process models that extends to data from the exponential dispersion family and to survival data. Our specific interest is in the analysis of data sets with predictors that have an a priori unknown form of possibly nonlinear associations to the response. The modeling approach we describe incorporates Gaussian processes in a generalized linear model framework to obtain a class of nonparametric regression models where the covariance matrix depends on the predictors. We consider, in particular, continuous, categorical and count responses. We also look into models that account for survival outcomes. We explore alternative covariance formulations for the Gaussian process prior and demonstrate the flexibility of the construction. Next, we focus on the important problem of selecting variables from the set of possible predictors and describe a general framework that employs mixture priors. We compare alternative MCMC strategies for posterior inference and achieve a computationally efficient and practical approach. We demonstrate performances on simulated and benchmark data sets.

Key words and phrases: Bayesian variable selection, generalized linear models, Gaussian processes, latent variables, MCMC, nonparametric regression, survival data.

1. INTRODUCTION

In this paper we present a unified modeling approach to Gaussian processes (GP) that extends to data from the exponential dispersion family and to survival data. With the advent of kernel-based methods, models utilizing Gaussian processes have become very common in machine learning approaches to regression and classification problems; see Rasmussen and Williams (2006). In the statistical literature GP regression models have been used as a nonparametric approach to model the nonlinear relationship between a response variable and a set of predictors; see, for example, O'Hagan (1978). Sacks, Schiller and Welch (1989) employed a stationary GP function of spatial locations in a regression model to account for residual spatial variation. Diggle, Tawn and Moyeed (1998) extended this construction to model the link function of the generalized linear model (GLM) construction of McCullagh and Nelder (1989). Neal (1999) considered linear regression and logit models.

We follow up on the literature cited above and introduce Gaussian process models as a class that broadens the generalized linear construction by incorporating fairly complex continuous response surfaces. The key idea of the construction is to introduce latent variables on which a Gaussian process prior is imposed. In the general case the GP construction replaces the linear relationship in the link function of a GLM. This results in a class of nonparametric regression models that

Terrance Savitsky is Associate Statistician, RAND Corporation, 1776 Main Street, Santa Monica, California 90401-3208, USA (e-mail: tds151@gmail.com). Marina Vannucci is Professor, Department of Statistics, Rice University, 6100 Main Street, Houston, Texas 77030, USA (e-mail: marina@rice.edu). Naijun Sha is Associate Professor, Department of Mathematical Sciences, University of Texas at El Paso, 500 W University Ave, El Paso, Texas 79968, USA (e-mail: nsha@utep.edu).

can accommodate linear and nonlinear terms, as well as noise terms that account for unexplained sources of variation in the data. The approach extends to latent regression models used for continuous, categorical and count data. Here we also consider a class of models that account for survival outcomes. We explore alternative covariance formulations for the GP prior and demonstrate the flexibility of the construction. In addition, we address practical computational issues that arise in the application of Gaussian processes due to numerical instability in the calculation of the covariance matrix.

Next, we look at the important problem of selecting variables from a set of possible predictors and describe a general framework that employs mixture priors. Bayesian variable selection has been a topic of much attention among researchers over the last few years. When a large number of predictors is available the inclusion of noninformative variables in the analysis may degrade the prediction results. Bayesian variable selection methods that use mixture priors were investigated for the linear regression model by George and McCulloch (1993, 1997), with contributions by various other authors on special features of the selection priors and on computational aspects of the method; see Chipman, George and McCulloch (2001) for a nice review. Extensions to linear regression models with multivariate responses were put forward by Brown, Vannucci and Fearn (1998b) and to multinomial probit by Sha et al. (2004). Early approaches to Bayesian variable selection for generalized linear models can be found in Chen, Ibrahim and Yiannoutsos (1999) and Raftery, Madigan and Volinsky (1996). Survival models were considered by Volinsky et al. (1997) and, more recently, by Lee and Mallick (2004) and Sha, Tadesse and Vannucci (2006). As for Gaussian process models, Linkletter et al. (2006) investigated Bayesian variable selection methods in the linear regression framework by employing mixture priors with a spike at zero on the parameters of the covariance matrix of the Guassian process prior.

Our unified treatment of Gaussian process models extends the line of work of Linkletter et al. (2006) to more complex data structures and models. We transform the covariance parameters and explore designs and MCMC strategies that aim at producing a minimally correlated parameter space and efficiently convergent sampling schemes. In particular, we find that Metropolis-within-Gibbs schemes achieve a substantial improvement in computational efficiency. Our results on simulated data and benchmark data sets show that GP models can lead to improved predictions without the requirement of pre-specifying higher order and nonlinear additive functions of the predictors. We show, in particular, that a Gaussian process covariance matrix with a single exponential term is able to map a mixture of linear and nonlinear associations with excellent prediction performance.

GP models can be considered part of the broad class of nonparametric regression models of the type $y = f(\mathbf{x}) + error$, with y an observed (or latent) response, f an unknown function and $\mathbf{x}$ a p-dimensional vector of covariates, and where the objective is to estimate the function f for prediction of future responses. Among possible alternative choices to GP models, one famous class is that of kernel regression models, where the estimate of f is selected from the set of functions contained in the reproducing kernel Hilbert space (RKHS) induced by a chosen kernel. Kernel models have a long and successful history in statistics and machine learning [see Parzen (1963), Wahba (1990) and Shawe-Taylor and Cristianini (2004)] and include many of the most widely used statistical methods for nonparametric estimation, including spline models and methods that use regularized techniques. Gaussian processes can be constructed with kernel convolutions and, therefore, GP models can be seen as contained in the class of nonparametric kernel regression with exponential family observations. Rasmussen and Williams (2006), in particular, note that the GP construction is equivalent to a linear basis regression employing an infinite set of Gaussian basis functions and results in a response surface that lies within the space of all mathematically smooth, that is, infinitely mean square differentiable, functions spanning the RKHS. Constructions of Bayesian kernel methods in the context of GP models can be found in Bishop (2006) and Rasmussen and Williams (2006).

Another popular class of nonparametric spline regression models is the generalized additive models (GAM) of Ruppert, Wand and Carroll (2003), that employ linear projections of the unknown function f onto a set of basis functions, typically cubic splines or B-splines, and related extensions, such as the structured additive regression (STAR) models of Fahrmeir, Kneib and Lang (2004) that, in addition, include interaction surfaces, spatial effects and random effects. Generally speaking, these regression models impose additional structure on the predictors and are therefore better suited for the purpose of interpretability, while Gaussian process models are better suited for prediction. Extensions of STAR models also enable variable selection based on spike and slab type priors; see, for example, Panagiotelis and Smith (2008).

Ensamble learning models, such as bagging, boosting and random forest models, utilize decision trees as basis functions; see Hastie, Tibshirani and Friedman (2001). Trees readily model interactions and nonlinearity subject to a maximum tree depth constraint to prevent overfitting. Generalized boosting models (GBMs), as an example, such as the AdaBoost of Freund and Schapire (1997), represent a nonlinear function of the covariates by simpler basis functions typically estimated in a stage-wise, iterative fashion that successively adds the basis functions to fit generalized or pseudo residuals obtained by minimizing a chosen loss function. GBMs accommodate dichotomous, continuous, event time and count responses. These models would be expected to produce similar prediction results to GP regression and classification models. We explore their behavior on one of the benchmark data sets in the application section of this paper. Notice that GBMs do not incorporate an explicit variable selection mechanism that allows to exclude nuisance covariates, as we do with GP models, although they do provide a relative measure of variable importance, averaged over all trees.

Regression trees partition the predictor space and fit independent models in different parts of the input space, therefore facilitating nonstationarity and leading to smaller local covariance matrices. "Treed GP" models are constructed by Gramacy and Lee (2008) and extend the constant and linear construction of Chipman, George and McCulloch (2002). A prior is specified over the tree process, and posterior inference is performed on the joint tree and leaf models. The effect of this formulation is to allow the correlation structure to vary over the input space. Since each tree region is composed of a portion of the observations, there is a computational savings to generate the GP covariance matrix from $m_r < n$ observations for region r. The authors note that treed GP models are best suited "...towards problems with a smaller number of distinct partitions...." So, while it is theoretically possible to perform variable selection in a forward selection manner, in applications these models are often used with single covariates.

The rest of the paper is organized as follows: In Section 2 we formally introduce the class of GP models by broadening the generalized linear construction. We also extend this class to include models for survival data. Possible constructions of the GP covariance matrix are enumerated in Section 3. Prior distributions for variable selection are discussed in Section 4 and posterior inference, including MCMC algorithms and prediction strategies, in Section 5. We include simulated data illustrations for continuous, count and survival data regression in Section 6, followed by benchmark applications in Section 7. Concluding remarks and suggestions for future research are in Section 8. Some details on computational issues and related pseudo-code are given in the Appendix.

2. GAUSSIAN PROCESS MODELS

We introduce Gaussian process models via a unified modeling approach that extends to data from the exponential dispersion family and to survival data.

2.1 Generalized Models

In a generalized linear model the monotone link function $g(\cdot)$ relates the linear predictors to the canonical parameter as $g(\eta_i) = \mathbf{x}_i'\boldsymbol{\beta}$, with η_i the canonical parameter for the ith observation, $\mathbf{x}_i = (x_1, \ldots, x_p)'$ a $p \times 1$ column vector of predictors for the ith subject and $\boldsymbol{\beta}$ the coefficient vector $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)'$. A broader class of models that incorporate fairly complex continuous response surfaces is obtained by introducing latent variables on which a Gaussian process prior is imposed. More specifically, the latent variables $z(\mathbf{x}_i)$ define the values of the link function as

$$g(\eta_i) = z(\mathbf{x}_i), \quad i = 1, \ldots, n, \tag{1}$$

and a Gaussian process (GP) prior on the $n \times 1$ latent vector is specified as

$$\mathbf{z}(\mathbf{X}) = (z(\mathbf{x}_1), \ldots, z(\mathbf{x}_n))' \sim N(\mathbf{0}, \mathbf{C}), \tag{2}$$

with the $n \times n$ covariance matrix $\mathbf{C}$ a fairly complex function of the predictors. This class of models can be cast within the model-based geostatistics framework of Diggle, Tawn and Moyeed (1998), with the dimension of the space being equal to the number of covariates.

The class of models introduced above extends to latent regression models used for continuous, categorical and count data. We provide some details on models for continuous and binary responses and for count data, since we will be using these cases in our simulation studies presented below. GP regression models are obtained by choosing the link function in (1) as the identity function, that is,

$$\mathbf{y} = \mathbf{z}(\mathbf{X}) + \boldsymbol{\varepsilon}, \tag{3}$$

with $\mathbf{y}$ the $n \times 1$ observed response vector, $\mathbf{z}(\mathbf{X})$ an n-dimensional realization from a GP as in (2), and

$\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \frac{1}{r}\mathbb{I}_n)$ with r a precision parameter. A Gamma prior can be imposed on r, that is, $r \sim \mathcal{G}(a_r, b_r)$. Linear models of type (3) were studied by Neal (1999) and Linkletter et al. (2006). One notices that, by integrating $\mathbf{z}(\mathbf{X})$ out, the marginalized likelihood is

$$\mathbf{y}|\mathbf{C}, r \sim \mathcal{N}\left(\mathbf{0}, \left[\frac{1}{r}\mathbb{I}_n + \mathbf{C}\right]\right), \tag{4}$$

that is, a regression model with the covariance matrix of the response depending on the predictors. Nonlinear response surfaces can be generated as a function of those covariates for suitable choices of the covariance matrix. We discuss some of the most popular in Section 3.

In the case of a binary response, class labels $t_i \in \{0, 1\}$ for $i = 1, \dots, n$ are observed. We assume $t_i \sim \text{Binomial}(1; p_i)$ and define $p_i = P(t_i = 1|z(\mathbf{x}_i))$ with $\mathbf{z}(\mathbf{X})$ as in (2). For logit models, for example, we have $p_i = \mathbf{F}(z(\mathbf{x}_i)) = 1/[1 + \exp(-z(\mathbf{x}_i))]$. Similarly, for binary probit we can directly define the inverse link function as $p_i = \Phi(z(\mathbf{x}_i))$, with $\Phi(\cdot)$ the cdf of standard normal distribution. However, a more common approach to inference in probit models uses data augmentation; see Albert and Chib (1993). This approach defines latent values y_i which are related to the response via a regression model, that is, in our latent GP framework, $y_i = z(\mathbf{x}_i) + \varepsilon_i$, with $\varepsilon_i \sim \mathcal{N}(0, 1)$, and associated to the observed classes, t_i, via the rule $t_i = 1$ if $y_i > 0$ and $t_i = 0$ if $y_i < 0$. Notice that the latent variable approach results in a GP on $\mathbf{y}$ with a covariance function obtained by adding a "jitter" of variance one to $\mathbf{C}$, with a similar effect of the noise component in the regression models (3) and (4). Neal (1999) argues that an effect close to a probit model can be produced by a logit model by introducing a large amount of jitter in its covariance matrix. Extensions to multivariate models for continuous and categorical responses are quite straightforward.

As another example, count data models can be obtained by choosing the canonical link function for the Poisson distribution as $\log(\boldsymbol{\lambda}) = \mathbf{z}(\mathbf{X})$ with $\mathbf{z}(\mathbf{X})$ as in (2). Over-dispersion, possibly caused from lack of inclusion of all possible predictors, is taken into account by modeling the extra variability via random effects, u_i, that is, $\tilde{\lambda}_i = \exp(z(\mathbf{x}_i) + u_i) = \exp(z(\mathbf{x}_i))\exp(u_i) = \lambda_i\delta_i$. For identifiability, one can impose $\mathbb{E}(\delta_i) = 1$ and marginalize over δ_i using a conjugate prior, $\delta_i \sim \mathcal{G}(\tau, \tau)$, to achieve the negative binomial likelihood as in Long (1997),

$$\begin{aligned} &\pi(s_i|\lambda_i, \tau) \\ &\quad = \frac{\Gamma(s_i + \tau)}{\Gamma(s_i + 1)\Gamma(\tau)}\left(\frac{\tau}{\tau + \lambda_i}\right)^{\tau}\left(\frac{\lambda_i}{\tau + \lambda_i}\right)^{s_i}, \end{aligned} \tag{5}$$

for $s_i \in \mathbb{N} \cup \{0\}$, with the same mean as the Poisson regression model, that is, $\mathbb{E}(s_i) = \lambda_i$, and $\text{Var}(s_i) = \lambda_i + \lambda_i^2/\tau$, with the added parameter τ capturing the variance inflation associated with over-dispersion.

2.2 Survival Data

The modeling approach via Gaussian processes exploited above extends to other classes of models, for example, those for survival data. In survival studies the task is typically to measure the effect of a set of variables on the survival time, that is, the time to a particular event or "failure" of interest, such as death or occurrence of a disease. The Cox proportional hazard model of Cox (1972) is an extremely popular choice. The model is defined through the hazard rate function $h(t|\mathbf{x}_i) = h_0(t)\exp(\mathbf{x}_i'\boldsymbol{\beta})$, where $h_0(\cdot)$ is the baseline hazard function, t is the failure time and $\boldsymbol{\beta}$ the p-dimensional regression coefficient vector. The cumulative baseline hazard function is denoted as $H_0(t) = \int_0^t h_0(u)\,du$ and the survivor function becomes $S(t|\mathbf{x}_i) = S_0(t)^{\exp(\mathbf{x}_i'\boldsymbol{\beta})}$, where $S_0(t) = \exp\{-H_0(t)\}$ is the baseline survivor function.

Let us indicate the data as $(t_1, \mathbf{x}_1, d_1), \dots, (t_n, \mathbf{x}_n, d_n)$ with censoring index $d_i = 0$ if the observation is right censored and $d_i = 1$ if the failure time t_i is observed. A GP model for survival data is defined as

$$h(t_i|z(\mathbf{x}_i)) = h_0(t_i)\exp(z(\mathbf{x}_i)), \quad i = 1, 2, \dots, n, \tag{6}$$

with $\mathbf{z}(\mathbf{X})$ as in (2). In this general setting, defining a probability model for Bayesian analysis requires the identification of a prior formulation for the cumulative baseline hazard function. One strategy often adopted in the literature on survival models is to utilize the partial likelihood of Cox (1972) that avoids prior specification and estimation of the baseline hazard, achieving a parsimonious representation of the model. Alternatively, Kalbfleisch (1978) employs a nonparametric gamma process prior on $H_0(t_i)$ and then calculates a marginalized likelihood. This "full" likelihood formulation tends to behave similarly to the partial likelihood one when the concentration parameter of the gamma process prior tends to 0, placing no confidence in the initial parametric guess. Sinha, Ibrahim and Chen (2003) extend this theoretical justification to time-dependent covariates and time-varying regression parameters, as well as to grouped survival data.

3. CHOICE OF THE GP COVARIANCE MATRIX

We explore alternative covariance formulations for the Gaussian process prior (2) and demonstrate the flexibility of the construction. In general, any plausible relationship between the covariates and the response can be represented through the choice of **C**, as long as the condition of positive definiteness of the matrix is satisfied; see Thrun, Saul and Scholkopf (2004). In the Appendix we further address practical computational issues that arise in the application of Gaussian processes due to numerical instability in the construction of the covariance matrix and the calculation of its inverse.

3.1 1-term vs. 2-term Exponential Forms

We consider covariance functions that include a constant term and a nonlinear, exponential term as

$$\mathbf{C} = \mathrm{Cov}(\mathbf{z}(\mathbf{X})) = \frac{1}{\lambda_a}\mathbf{J}_n + \frac{1}{\lambda_z}\exp(-\mathbf{G}), \tag{7}$$

with $\mathbf{J}_n$ an $n \times n$ matrix of 1's and $\exp(\mathbf{G})$ a matrix with elements $\exp(g_{ij})$, where $g_{ij} = (\mathbf{x}_i - \mathbf{x}_j)'\mathbf{P}(\mathbf{x}_i - \mathbf{x}_j)$ and $\mathbf{P} = \mathrm{diag}(-\log(\rho_1, \ldots, \rho_p))$, with $\rho_k \in [0, 1]$ associated to $x_k, k = 1, \ldots, p$. In the literature on Gaussian processes a noise component, called "jitter," is sometimes added to the covariance matrix **C**, in addition to the term $(1/\lambda)\mathbf{J}$, in order to make the matrix computations better conditioned; see Neal (1999). This is consistent with the belief that there may be unexplained sources of variation in the data, perhaps due to explanatory variables that were not recorded in the original study. The parametrization of **G** we adopt allows simpler prior specifications (see below), and it is also used by Linkletter et al. (2006) as a transformation of the exponential term used by Neal (1999) and Sacks, Schiller and Welch (1989) in their formulations. Neal (1999) notices that introducing an intercept in model (3), with precision parameter λ_a, placing a Gaussian prior on it and then marginalizing over the intercept produces the additive covariance structure (7). The parameter for the exponential term, λ_z, serves as a scaling factor for this term. In our empirical investigations we found that construction (7) is sensitive to scaling and that best results can be obtained by normalizing **X** to lie in the unit cube, $[0, 1]^p$, though standardizing the columns to mean 0 and variance 1 produces similar results.

The single-term exponential covariance provides a parsimonious representation that enables a broad class of linear and nonlinear response surfaces. Plots (a)–(c) of Figure 1 show response curves produced by utilizing a GP with the exponential covariance matrix (7) and three different values of ρ. One readily notes how higher order polynomial-type response surfaces can be generated by choosing relatively lower values for ρ, whereas the assignment of higher values provides lower order polynomial-type that can also include roughly linear response surfaces [plot (c)].

We also consider a two-term covariance obtained by adding a second exponential term to (7), that is,

$$\begin{aligned}\mathbf{C} &= \mathrm{Cov}(\mathbf{z}(\mathbf{X})) \\ &= \frac{1}{\lambda_a}\mathbf{J}_n + \frac{1}{\lambda_{1,z}}\exp(-\mathbf{G}_1) + \frac{1}{\lambda_{2,z}}\exp(-\mathbf{G}_2),\end{aligned} \tag{8}$$

where $\mathbf{G}_1$ and $\mathbf{G}_2$ are parameterized as $\mathbf{P}_1 = \mathrm{diag}(-\log(\rho_{1,1}, \ldots, \rho_{1,p}))$ and $\mathbf{P}_2 = \mathrm{diag}(-\log(\rho_{2,1}, \ldots, \rho_{2,p}))$, respectively. As noted in Neal (2000), adding multiple terms results in rougher, more complex, surfaces while retaining the relative computational efficiency of the exponential formulation. For example, plot (d) of Figure 1 shows examples of surfaces that can be generated by employing the 2-term covariance formulation with $(\rho_1, \rho_2) = (0.5, 0.05)$ and $(\lambda_{1,z} = 1, \lambda_{2,z} = 8)$.

3.2 The Matern Construction

An alternative choice to the exponential covariance term is the Matern formulation. This introduces an explicit smoothing parameter, ν, such that the resulting Gaussian process is k times differentiable for $k \le \nu$,

$$\begin{aligned}&\mathbf{C}(z(\mathbf{x}_i), z(\mathbf{x}_j)) \\ &\quad = \frac{1}{2^{\nu-1}\Gamma(\nu)}\big[2\sqrt{\nu d(\mathbf{x}_i, \mathbf{x}_j)}\big]^{\nu} K_\nu\big[2\sqrt{\nu d(\mathbf{x}_i, \mathbf{x}_j)}\big],\end{aligned} \tag{9}$$

with $d(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)'\mathbf{P}(\mathbf{x}_i - \mathbf{x}_j)$, $K_\nu(\cdot)$ the Bessel function and **P** parameterized as in (7). Banerjee et al. (2008) employ such a construction with ν fixed to 0.5 for modeling a spacial random effects process characterized by roughness. One recovers the exponential covariance term from the Matern construction in the limit as $\nu \to \infty$. However, Rasmussen and Williams (2006) point out that two formulations are essentially the same for $\nu > \frac{7}{2}$, as confirmed by our own simulations.

4. PRIOR MODEL FOR BAYESIAN VARIABLE SELECTION

The unified modeling approach we have described allows us to put forward a general framework for variable selection that employs Bayesian methods and mixture priors for the selection of the predictors. In particular, variable selection can be achieved within the GP modeling framework by imposing "spike-and-slab"

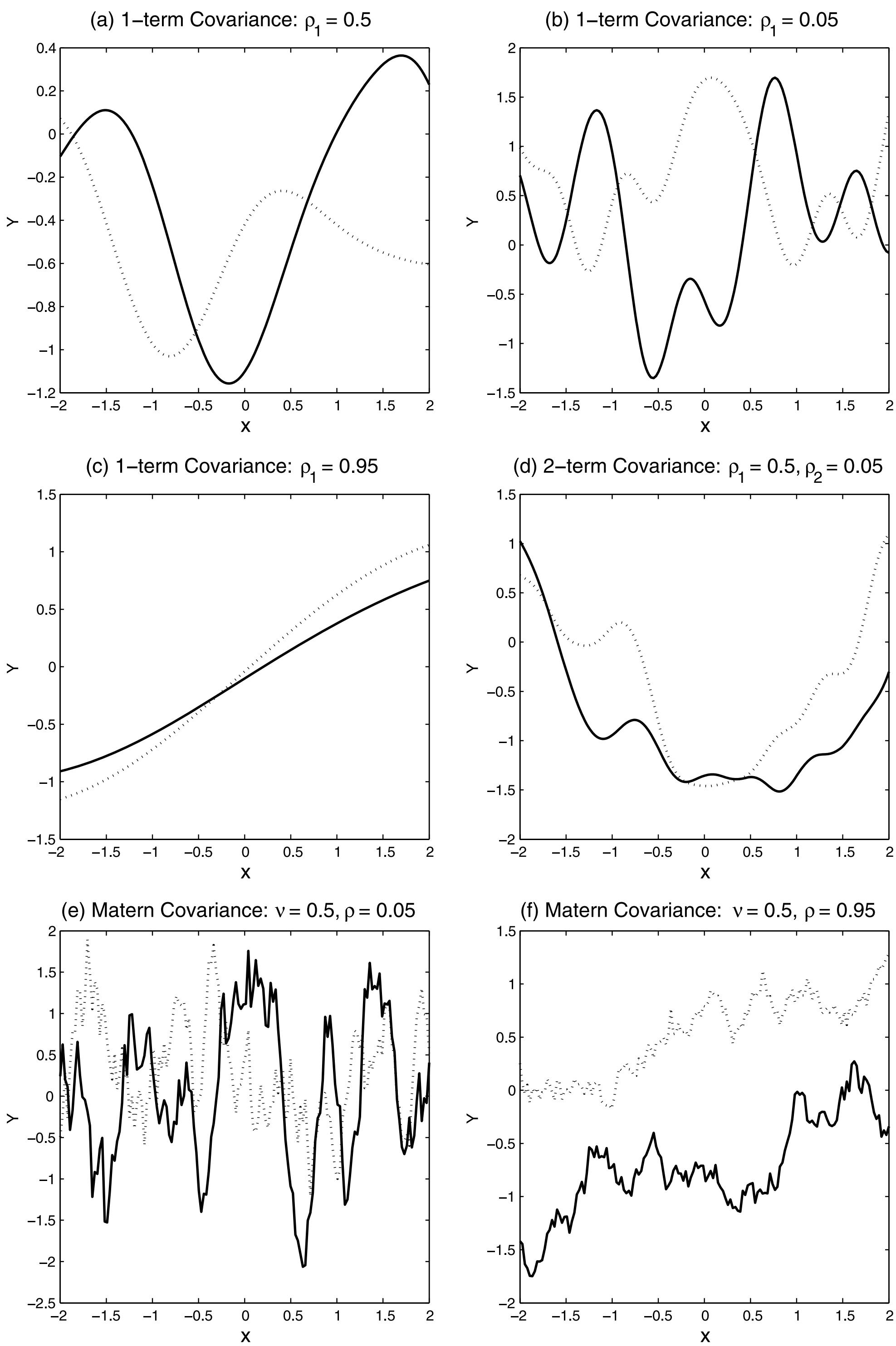


FIG. 1. *Response curves drawn from a GP. Each plot shows two* (*solid and dashed*) *random realizations. Plots* (a)–(c) *were obtained with the exponential covariance* (7) *and plot* (d) *with the* 2-*term formulation* (8). *Plots* (e) *and* (f) *show realizations from the matern construction. All curves employ a one-dimensional covariate.*

mixture priors on the covariance parameters in (7), that is,

$$\pi(\rho_k|\gamma_k) = \gamma_k \mathbb{I}[0 \le \rho_k \le 1] + (1-\gamma_k)\delta_1(\rho_k), \tag{10}$$

for $k = 1, \dots, p$, with $\delta_1(\cdot)$ a point mass distribution at one. Clearly, $\rho_k = 1$ causes the predictor x_k to have no effect on the computation for the GP covariance matrix. This formulation is similar in spirit to the use of selection priors for linear regression models and is employed by Linkletter et al. (2006) in the univariate GP regression framework (3). Further Bernoulli priors are imposed on the selection parameters, that is, $\gamma_k \sim \text{Bernoulli}(\alpha_k)$ and Gamma priors are specified on the precision terms (λ_a, λ_z).

Variable selection with a covariance matrix that employs two exponential terms as in (8) is more complex. In particular, one can select covariates separately for each exponential term by assigning a specific set of variable selection parameters to each term, that is, $(\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2)$ associated to $(\boldsymbol{\rho}_1, \boldsymbol{\rho}_2)$, and simply extending the single term formulation via independent spike-and-slab priors of the form

$$\begin{aligned} &\pi(\rho_{1,k}|\gamma_{1,k}) \\ &\quad = \gamma_{1,k}\mathbb{I}[0 \le \rho_{1,k} \le 1] + (1-\gamma_{1,k})\delta_1(\rho_{1,k}), \end{aligned} \tag{11}$$

$$\begin{aligned} &\pi(\rho_{2,k}|\gamma_{2,k}) \\ &\quad = \gamma_{2,k}\mathbb{I}[0 \le \rho_{2,k} \le 1] + (1-\gamma_{2,k})\delta_1(\rho_{2,k}), \end{aligned} \tag{12}$$

with $k = 1, \dots, p$. Assuming *a priori* independence of the two model spaces, Bernoulli priors can be imposed on the selection parameters, that is, $\gamma_{i,k} \sim \text{Bernoulli}(\alpha_{i,k}), i = 1, 2$. This variable selection framework identifies the association of each covariate, x_k, to one or both terms. Final selection can then be accomplished by choosing the covariates in the union of those selected by *either* of the two terms. An alternative strategy for variable selection may employ a common set of variable selection parameters, $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_p)$ for *both* $\boldsymbol{\rho}_1$ and $\boldsymbol{\rho}_2$, in a joint spike-and-slab (product) prior formulation,

$$\begin{aligned} &\pi(\rho_{1,k}, \rho_{2,k}|\gamma_k) \\ &\quad = \gamma_k \mathbb{I}[0 \le \rho_{1,k} \le 1]\mathbb{I}[0 \le \rho_{2,k} \le 1] \\ &\qquad + (1-\gamma_k)\delta_1(\rho_{1,k})\delta_1(\rho_{2,k}), \end{aligned} \tag{13}$$

where we assume a priori independence of the parameter spaces, $\boldsymbol{\rho}_1$ and $\boldsymbol{\rho}_2$. This prior choice focuses more on overall covariate selection, rather than simultaneous selection and assignment to each term in (8). While we lose the ability to align the $\rho_{i,k}$ to each covariance function term, we expect to improve computational efficiency by jointly sampling $(\boldsymbol{\gamma}, \boldsymbol{\rho}_1, \boldsymbol{\rho}_2)$ at each iteration of the MCMC scheme as compared to a separate joint sampling on $(\boldsymbol{\gamma}_1, \boldsymbol{\rho}_1)$ and $(\boldsymbol{\gamma}_2, \boldsymbol{\rho}_2)$. Some investigation is done in Savitsky (2010).

5. POSTERIOR INFERENCE

The methods for posterior inference we are going to describe apply to all GP formulations, even though we focus our simulation work on the continuous and count data models. We therefore express the posterior formulation employing a generalized notation. First, we collect all parameters of the GP covariance matrix in $\boldsymbol{\Theta}$ and write $\mathbf{C} = \mathbf{C}(\boldsymbol{\Theta})$. For example, for covariance matrix of type (7) we have $\boldsymbol{\Theta} = (\boldsymbol{\rho}, \lambda_a, \lambda_z)$. Next, we extend our notation to include the selection parameter $\boldsymbol{\gamma}$ by using $\boldsymbol{\Theta}_{\boldsymbol{\gamma}} = (\boldsymbol{\rho}_{\boldsymbol{\gamma}}, \lambda_a, \lambda_z)$ to indicate that $\rho_k = 1$ when $\gamma_k = 0$, for $k = 1, \dots, p$. For covariance of type (8) we write $\boldsymbol{\Theta}_{\boldsymbol{\gamma}} = \{\boldsymbol{\Theta}_{\boldsymbol{\gamma}_1}, \boldsymbol{\Theta}_{\boldsymbol{\gamma}_2}, \lambda_a\}$, where $\boldsymbol{\gamma} = (\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2)'$ and $\boldsymbol{\Theta}_{\boldsymbol{\gamma}_i} = (\boldsymbol{\rho}_{i\boldsymbol{\gamma}_i}, \lambda_{i,z}), i \in \{1, 2\}$ for prior of type (11)–(12) and $\boldsymbol{\Theta}_{\boldsymbol{\gamma}} = (\boldsymbol{\rho}_{1\boldsymbol{\gamma}}, \boldsymbol{\rho}_{2\boldsymbol{\gamma}}, \lambda_a, \lambda_{1,z}, \lambda_{2,z})$ for prior of type (13), and similarly for the Matern construction. Next, we define $D_i \in \{y_i, \{s_i, z(\mathbf{x}_i)\}\}$ and $\mathbf{D} := \{D_1, \dots, D_n\}$ to capture the observed data *augmented* by the unobserved GP variate, $\mathbf{z}(\mathbf{X})$, for the latent response models [such as model (5) for count data]. Finally, we set $\mathbf{h} := \{r, \tau\}$ to group unique parameters $\notin \boldsymbol{\Theta}_{\boldsymbol{\gamma}}$ and we collect hyperparameters in $\mathbf{m} := \{\mathbf{a}, \mathbf{b}\}$, with $\mathbf{a} = \{a_{\lambda_a}, a_{\lambda_z}, a_r, a_\tau\}$ and similarly for $\mathbf{b}$, where $\mathbf{a}$ and $\mathbf{b}$ include the shape and rate hyperparameters of the Gamma priors on the associated parameters. With this notation we can finally outline a generalized expression for the full conditional of $(\boldsymbol{\gamma}, \boldsymbol{\rho}_{\boldsymbol{\gamma}})$ as

$$\begin{aligned} &\pi(\boldsymbol{\gamma}, \boldsymbol{\rho}_{\boldsymbol{\gamma}}|\boldsymbol{\Theta}_{\boldsymbol{\gamma}} \backslash \boldsymbol{\rho}_{\boldsymbol{\gamma}}, \mathbf{D}, \mathbf{h}, \mathbf{m}) \\ &\quad \propto L^a(\boldsymbol{\gamma}, \boldsymbol{\rho}_{\boldsymbol{\gamma}}|\boldsymbol{\Theta}_{\boldsymbol{\gamma}} \backslash \boldsymbol{\rho}_{\boldsymbol{\gamma}}, \mathbf{D}, \mathbf{h}, \mathbf{m})\pi(\boldsymbol{\gamma}), \end{aligned} \tag{14}$$

with L^a the augmented likelihood. Notice that the term $\pi(\boldsymbol{\rho}_{\boldsymbol{\gamma}}|\boldsymbol{\gamma})$ does not appear in (14) since $\pi(\rho_k|\gamma_k) = 1$, for $k = 1, \dots, p$.

5.1 Markov Chain Monte Carlo—Scheme 1

We first describe a Metropolis–Hastings scheme within Gibbs sampling to jointly sample $(\boldsymbol{\gamma}, \boldsymbol{\rho}_{\boldsymbol{\gamma}})$, which is an adaptation of the MCMC model comparison (MC^3) algorithm originally outlined in Madigan and York (1995) and extensively used in the variable selection literature. As we are unable to marginalize over the parameter space, we need to modify the algorithm in a hierarchical fashion, using the move types outlined below. Additionally, we need to sample all the other nuisance parameters.

A generic iteration of this MCMC procedure comprises the following steps:

(1) *Update* (γ, ρ_γ): Randomly choose among three between-models transition moves:

(i) Add: set $\gamma_k' = 1$ and sample ρ_k' from a $\mathcal{U}(0, 1)$ proposal. Position k is randomly chosen from the set of k's where $\gamma_k = 0$ at the previous iteration.

(ii) Delete: set $(\gamma_k' = 0, \rho_k' = 1)$. This results in covariate x_k being excluded in the current iteration. Position k is randomly chosen from among those included in the model at the previous iteration.

(iii) Swap: perform both an *Add* and *Delete* move. This move type helps to more quickly traverse a large covariate space.

The proposed value $(\boldsymbol{\gamma}', \boldsymbol{\rho}'_{\boldsymbol{\gamma}'})$ is accepted with probability,

$$\alpha = \min\left\{1, \frac{\pi(\boldsymbol{\gamma}', \boldsymbol{\rho}'_{\boldsymbol{\gamma}'}|\boldsymbol{\Theta}_{\boldsymbol{\gamma}'}\backslash\boldsymbol{\rho}'_{\boldsymbol{\gamma}'}, \mathbf{D}, \mathbf{h}, \mathbf{m})q(\boldsymbol{\gamma}|\boldsymbol{\gamma}')}{\pi(\boldsymbol{\gamma}, \boldsymbol{\rho}_{\boldsymbol{\gamma}}|\boldsymbol{\Theta}_{\boldsymbol{\gamma}}\backslash\boldsymbol{\rho}_{\boldsymbol{\gamma}}, \mathbf{D}, \mathbf{h}, \mathbf{m})q(\boldsymbol{\gamma}'|\boldsymbol{\gamma})}\right\},$$

where the ratio of the proposals $q(\boldsymbol{\rho}_{\boldsymbol{\gamma}})/q(\boldsymbol{\rho}'_{\boldsymbol{\gamma}'})$ drops out of the computation since we employ a $\mathcal{U}(0, 1)$ proposal.

(2) Execute a Gibbs-type move, *Keep*, by sampling from a $\mathcal{U}(0, 1)$ all ρ_k''s such that $\gamma_k' = 1$. This move is not required for ergodicity, but it allows to perform a refinement of the parameter space within the existing model, for faster convergence.

(3) *Update* $\{\lambda_a, \lambda_z\}$: These are updated using Metropolis–Hastings moves with Gamma proposals centered on the previously sampled values.

(4) *Update* $\mathbf{h}$: Individual model parameters in $\mathbf{h}$ are updated using Metropolis–Hastings moves with proposals centered on the previously sampled values.

(5) *Update* $\mathbf{z}$: Jointly sample $\mathbf{z}$ for latent response models using the approach enumerated in Neal (1999) with proposal $\mathbf{z}' = (1 - \varepsilon^2)^{1/2}\mathbf{z} + \varepsilon\mathbf{L}\mathbf{u}$, where $\mathbf{u}$ is a vector of i.i.d. standard Gaussian values and $\mathbf{L}$ is the Cholesky decomposition of the GP covariance matrix. For faster convergence R consecutive updates are performed at each iteration.

Green (1995) introduced a Markov chain Monte Carlo method for Bayesian model determination for the situation where the dimensionality of the parameter vector varies iteration by iteration. Recently, Gottardo and Raftery (2008) have shown that the reversible jump can be formulated in terms of a mixture of singular distributions. Following the results given in their examples, it is possible to show that the acceptance probability of the reversible jump formulation is the same as in the Metropolis–Hastings algorithm described above, and therefore that the two algorithms are equivalent; see Savitsky (2010).

For inference, estimates of the marginal posterior probabilities of $\gamma_k = 1$, for $k = 1 \ldots, p$, can be computed based on the MCMC output. A simple strategy is to compute Monte Carlo estimates by counting the number of appearances of each covariate across the visited models. Alternatively, Rao–Blackwellized estimates can be calculated by averaging the full conditional probabilities of $\gamma_k = 1$. Although computationally more expensive, the latter strategy may result in estimates with better precision, as noted by Guan and Stephens (2011). In all simulations and examples reported below we obtained satisfactory results by estimating the marginal posterior probabilities by counts restricted to between-models moves, to avoid overestimation.

5.2 Markov Chain Monte Carlo—Scheme 2

Next we enumerate a Markov chain Monte Carlo algorithm to directly sample $(\boldsymbol{\gamma}, \boldsymbol{\rho}_{\boldsymbol{\gamma}})$ with a Gibbs scan that employs a Metropolis acceptance step. We formulate a proposal distribution of a similar mixture form as the joint posterior by extending a result from Gottardo and Raftery (2008) to produce a move to $(\gamma_k = 0, \rho_k = 1)$, as well as to $(\gamma_k = 1, \rho_k = [0, 1))$.

A generic iteration of this MCMC procedure comprises the following steps:

(1) For $k = 1, \ldots, p$ perform a joint update for (γ_k, ρ_k) with two moves, conducted in succession:

(i) Between-models: Jointly propose a new model such that if $\gamma_k = 1$, propose $\gamma_k' = 0$ and set $\rho_k' = 1$; otherwise, propose $\gamma_k' = 1$ and draw $\rho_k' \sim \mathcal{U}(0, 1)$. Accept the proposal for (γ_k', ρ_k') with probability,

$$\alpha = \min\left\{1, \frac{\pi(\gamma_k', \rho_k'|\boldsymbol{\gamma}'_{(k)}, \boldsymbol{\Theta}_{\boldsymbol{\gamma}'_{(k)}}, \mathbf{D}, \mathbf{h}, \mathbf{m})}{\pi(\gamma_k, \rho_k|\boldsymbol{\gamma}'_{(k)}, \boldsymbol{\Theta}_{\boldsymbol{\gamma}'_{(k)}}, \mathbf{D}, \mathbf{h}, \mathbf{m})}\right\},$$

where now $\boldsymbol{\gamma}'_{(k)} := (\gamma_1', \ldots, \gamma_{k-1}', \gamma_{k+1}, \ldots, \gamma_p)$ and similarly for $\boldsymbol{\rho}'_{(k)} \in \boldsymbol{\Theta}_{\boldsymbol{\gamma}'_{(k)}}$. The joint proposal ratio for (γ_k, ρ_k), reduces to 1 since we employ a $\mathcal{U}(0, 1)$ proposal for $\rho_k \in [0, 1]$ and a symmetric Dirac measure proposal for γ_k.

(ii) Within model: This move is performed only if we sample $\gamma_k' = 1$ from the between-models move, in which case we propose $\gamma_k'' = 1$ and, as before, draw $\rho_k'' \sim \mathcal{U}(0, 1)$. Similar to the between-models move, accept the joint proposal for (γ_k'', ρ_k'') with probability,

$$\alpha = \min\left\{1, \frac{\pi(\gamma_k'', \rho_k''|\boldsymbol{\gamma}'_{(k)}, \boldsymbol{\Theta}_{\boldsymbol{\gamma}'_{(k)}}, \mathbf{D}, \mathbf{h}, \mathbf{m})}{\pi(\gamma_k', \rho_k'|\boldsymbol{\gamma}'_{(k)}, \boldsymbol{\Theta}_{\boldsymbol{\gamma}'_{(k)}}, \mathbf{D}, \mathbf{h}, \mathbf{m})}\right\},$$

which further reduces to just the ratio of posteriors since we propose a move within the current model and utilize a $\mathcal{U}(0, 1)$ proposal for ρ_k.

(2) Sample the parameters $\{\lambda_a, \lambda_z, \mathbf{h}\}$ and latent responses $\mathbf{z}$ as outlined in scheme 1.

In simulations we also investigate performances of an adaptive scheme that employs a proposal with tuning parameters adapted based on "learning" from the data. In particular, we employ the method of Haario, Saksman and Tamminen (2001) for our Bernoulli proposal for $\gamma|\alpha$ to successively update the mean parameter, $\alpha_k, k = 1, \dots, p$, based on prior sampled values for γ_k. The construction does not require additional likelihood computations and it is expected to achieve more rapid convergence in the model space than the non-adaptive scheme. Roberts and Rosenthal (2007) and Ji and Schmidler (2009) note conditions under which adaptive schemes achieve convergence to the target posterior distribution.

Schemes 1 and 2 we enumerated above may be easily modified when employing the 2-term covariance formulation (8); see Savitsky (2010).

5.3 Prediction

Let $\mathbf{z}_f = \mathbf{z}(\mathbf{X}_f)$ be an $n_f \times 1$ latent vector of future cases. We use the regression model (3) to demonstrate prediction under the GP framework. The joint distribution over training and test sets is defined to be $\mathbf{z}_* := [\mathbf{z}', \mathbf{z}'_f]' \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_{n+n_f})$ with covariance,

$$\mathbf{C}_{n+n_f} := \begin{pmatrix} \mathbf{C}_{(\mathbf{X},\mathbf{X})} & \mathbf{C}_{(\mathbf{X},\mathbf{X}_f)} \\ \mathbf{C}_{(\mathbf{X}_f,\mathbf{X})} & \mathbf{C}_{(\mathbf{X}_f,\mathbf{X}_f)} \end{pmatrix},$$

where $\mathbf{C}_{(\mathbf{X},\mathbf{X})} := \mathbf{C}_{(\mathbf{X},\mathbf{X})}(\boldsymbol{\Theta})$. The conditional joint predictive distribution over the test cases, $\mathbf{z}_f|\mathbf{z}$, is also multivariate normal distribution with expectation $\mathbb{E}[\mathbf{z}_f|\mathbf{z}] = \mathbf{C}_{(\mathbf{X}_f,\mathbf{X})}\mathbf{C}^{-1}_{(\mathbf{X},\mathbf{X})}\mathbf{z}$. Estimation is based on the posterior MCMC samples. Here we take a computationally simple approach by first estimating $\hat{\mathbf{z}}$ as the mean of all sampled values of $\mathbf{z}$, defining

$$\mathbf{D}(\boldsymbol{\Theta}) := \mathbf{C}_{(\mathbf{X}_f,\mathbf{X})}\mathbf{C}^{-1}_{(\mathbf{X},\mathbf{X})}\hat{\mathbf{z}}, \tag{15}$$

and then estimating the response value as

$$\hat{\mathbf{y}}_f = \hat{\mathbf{z}}_f|\hat{\mathbf{z}} = \frac{1}{K}\sum_{t=1}^{K}\mathbf{D}(\boldsymbol{\Theta}^{(t)}), \tag{16}$$

with K the number of MCMC iterations and where calculations of the covariance matrices in (15) are restricted to the variables selected based on the marginal posterior probabilities of $\gamma_k = 1$. A more coherent estimation procedure, that may return more precise estimates but that is also computationally more expensive, would compute Rao–Blackwellized estimates by averaging the predictive probabilities over all visited models; see Guan and Stephens (2011). In the simulations and examples reported below we have calculated (16) using every 10th MCMC sampled value, to provide a relatively less correlated sample and save on computational time. In addition, when computing the variance product term in (15), we have employed the Cholesky decomposition $\mathbf{C} = \mathbf{L}\mathbf{L}'$, following Neal (1999), to avoid direct computation of the inverse of $\mathbf{C}_{(\mathbf{X},\mathbf{X})}$.

For categorical data models, we may predict the new class labels, $\mathbf{t}_f$, via the rule of largest probability in the case of a binary logit model, with estimated latent realizations $\hat{\mathbf{z}}_f$, and via data augmentation based on the values of $\hat{\mathbf{y}}_f$ in the case of a binary probit model.

5.3.1 *Survival Function Estimation.* For survival data it is of interest to estimate the survivor function for a new subject with unknown event time, T_i, and associated $z_{f,i} := z_{f,i}(\mathbf{x}_{f,i})$. This is defined as

$$\begin{aligned} P(T_i \geq t|z_{f,i}, \mathbf{z}) &= S_i(t|z_{f,i}, \mathbf{z}) \\ &= S_0(t|\mathbf{z})^{\exp(z_{f,i})}. \end{aligned} \tag{17}$$

When using the partial likelihood formulation an empirical Bayes estimate of the baseline survivor function, $S_0(t|\mathbf{z})$, must be calculated, since the model does not specifically enumerate the baseline hazard. Weng and Wong (2007), for example, propose a method that discretizes the likelihood to produce an estimator with the useful property that it cannot take negative values. Accuracy of this estimate may be potentially improved by Rao–Blackwellizing the computation by averaging over the MCMC runs.

6. SIMULATION STUDY

6.1 Parameter Settings

In all simulations and applications reported in this paper we set both priors on λ_a and λ_z as $\mathcal{G}(1, 1)$. We did not observe any strong sensitivity to this choice. In particular, we considered different choices of the two parameters of these Gammma priors in the range $(0.01, 1)$, keeping the prior mean at 1 but with progressively larger variances, and observed very little change in the range of posterior sampled values. We also experimented with prior mean values of 10 and 100, which produced only a small impact on the posterior. For model (3) we set $r \sim \mathcal{G}(a_r, b_r)$ with $(a_r, b_r) = (2, 0.1)$ to reflect our a priori expected residual variance. For the count model (5), we set $\tau \sim \mathcal{G}(1, 1)$. For survival data, when using the full likelihood from Kalbfleisch (1978) we specified a $\mathcal{G}(1, 1)$ prior for both the parameter of the exponential base distribution and the concentration parameter of the Gamma process prior on the baseline.

Some sensitivity on the Bernoulli priors on the γ_k's is, of course, to be expected, since these priors drive the sparsity of the model. Generally speaking, parsimonious models can be selected by specifying $\gamma_k \sim \text{Bernoulli}(\alpha_k)$ with $\alpha_k = \alpha$ and α a small percentage of the total number of variables. In our simulations we set α_k to 0.025. We observed little sensitivity in the results for small changes around this value, in the range of 0.01–0.05, though we would expect to see significant sensitivity for much higher values of α. We also investigated sensitivity to a Beta hyperprior on α; see below.

When running the MCMC algorithms independent chain samplers with $\mathcal{U}(0, 1)$ proposals for the ρ_k's have worked well in all applications reported in this paper, where we have always approximately achieved the target acceptance rate of 40–60% indicating efficient posterior sampling.

6.2 Use of Variable Selection Parameters

We first demonstrate the advantage of introducing selection parameters in the model. Figure 2 shows results with and without the inclusion of the variable selection parameter vector $\boldsymbol{\gamma}$ on a simulated scenario with a kernel that incorporates both linear and nonlinear associations. The observed continuous response, y, is constructed from a mix of linear and nonliner relationships to 4 variables, each generated from a $\mathcal{U}(0, 1)$,

$$y = x_1 + x_2 + \sin(3x_3) + \sin(5x_4) + \varepsilon,$$

with $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ and $\sigma = 0.05$. Additional variables are randomly generated, again from $\mathcal{U}(0, 1)$. In this simulation we used $(n, p) = (80, 20)$. We ran 70,000 MCMC iterations, of which 10,000 were discarded as burn-in.

Plot (a) of Figure 2 displays box plots of the MCMC samples for the $\rho_k' s, k = 1, \dots, 20$, for the case of no variable selection, that is, by using a simple "slab" prior on the ρ_k's. As both Linkletter et al. (2006) and Neal (2000) note, the single covariates demonstrate an association to the response whose strength may be assessed utilizing the distance of the posterior samples of the ρ_k's from 1. One notes that, according to this criterion, the true covariates are all selected. It is conceivable, however, for some of the unrelated covariates to be selected using the same criterion, since the ρ_k's all sample below 1, and that this problem would be compounded as p grows. Plot (b) of Figure 2, instead, captures results from employing the variable selection parameters $\boldsymbol{\gamma}$ and shows how the inclusion of these parameters results in the sampled values of the ρ_k's for variables unrelated to the response being all pushed up against 1.

This simple simulated scenario also helps us to illustrate a couple of other features. First, a single exponential term in (7) is able to capture a wide variety

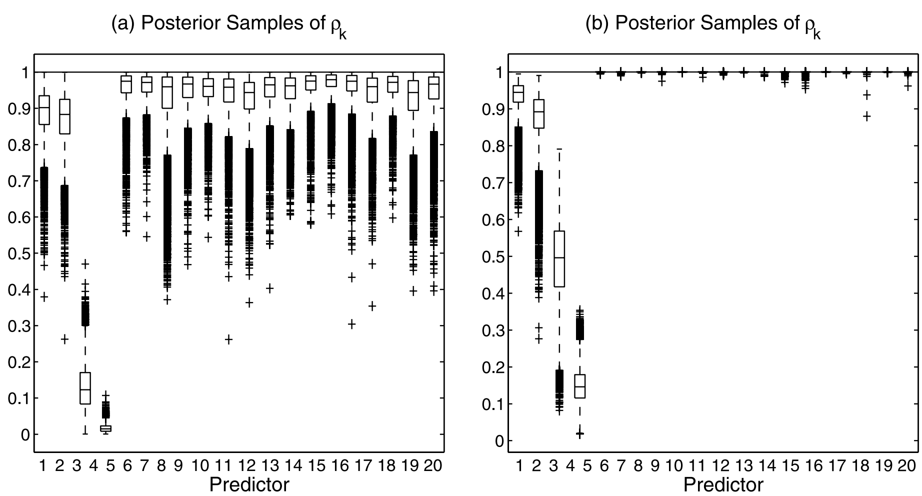


FIG. 2. *Use of variable selection parameters: Simulated data* $(n = 80, p = 20)$. *Box plots of posterior samples for* $\rho_k \in [0, 1]$. *Plots* (a) *and* (b) *demonstrate selection without and with, respectively, the inclusion of the selection parameter* $\boldsymbol{\gamma}$.

of continuous response surfaces, allowing a great flexibility in the shape of the response surface, with the linear fit being a subset of one of many types of surfaces that can be generated. Second, the effect of covariates with higher-order polynomial-like association to the response is captured by having estimates of the corresponding ρ_k's further away from 1; see, for example, covariate x_4 in Figure 2 which expresses the highest order association to the response.

6.3 Large p

Next we show simulation results on continuous, count and survival data models, for $(n, p) = (100, 1{,}000)$. We employ an additive term as the kernel for all models,

$$y = a_1x_1 + a_2x_2 + a_3x_3 + a_4x_4 + a_5\sin(a_6x_5) + a_7\sin(a_8x_6) + \varepsilon. \tag{18}$$

The functional form for the simulation kernel is designed so that the first four covariates express a linear relationship to the response while the next two express nonlinear associations. Model-specific coefficient values are displayed in Table 1. Methods employed to randomly generate the observed count and event time data from the latent response kernel are also outlined in the table. For example, the kernel captures the log-mean of the Poisson distribution used to generate count data, and it is used to generate the survivor function that is inverted to provide event time data for the Cox model. As in the previous simulation, all covariates are generated from $\mathcal{U}(0, 1)$.

We set the hyperparameters as described in Section 6.1. We used MCMC scheme 1 and increased the number of total iterations, with respect to the simpler simulation with only $p = 20$, to 800,000 iterations, discarding half of them for burn-in.

Results are reported in Table 1. While the continuous and count data GP models readily assigned high marginal posterior probabilities to the correct covariates (figures not shown), the Cox GP model correctly identified only 5 of 6 predictors; see Figure 3 for the posterior distributions of $\gamma_k = 1$ and the box plots for the posterior samples of ρ_k for this model (for readability, only the first 20 covariates are displayed). The predictive power for the continuous and count data models was assessed by normalizing the mean squared prediction error (MSPE) with the variance of the test set. Excellent results were achieved in our simulations. For the Cox GP model, the averaged survivor function estimated on the test set is shown in Figure 4, where we observe a tight fit between the estimated curve and the Kaplan–Meier empirical estimate constructed from the same test data.

Though for the Cox model we only report results obtained using the partial likelihood formulation, we conducted the same simulation study with the model based

TABLE 1
Large p: Simulations for continuous, count and survival data models with $(n, p) = (100, 1{,}000)$

	Continuous data	**Count data**	**Cox model**
Coefficients:			
a_1	1.0	1.6	3.0
a_2	1.0	1.6	−2.5
a_3	1.0	1.6	3.5
a_4	1.0	1.6	−3.0
a_5	1.0	1.0	1.0
a_6	3.0	3.0	3.0
a_7	1.0	1.0	−1.0
a_8	5.0	5.0	5.0
Model	Identity link	$\log(\lambda) = y$ $t \sim \mathrm{Pois}(\lambda)$	$S(t\|y) = \exp[-H_0(t)\exp(y)]$ $H_0(t) = \lambda t, \lambda = 0.2$ $t = M/(\lambda\exp(y)), M \sim \mathrm{Exp}(1)$ 5% uniform randomly censored, $t_{\mathrm{cens}} = \mathcal{U}(0, t_{\mathrm{event}})$
Train/test	100/20	100/20	100/60
Correctly selected	6 out of 6	6 out of 6	5 out of 6
False positives	0	0	0
MSPE (normalized)	0.0067	0.045	see Figure 4

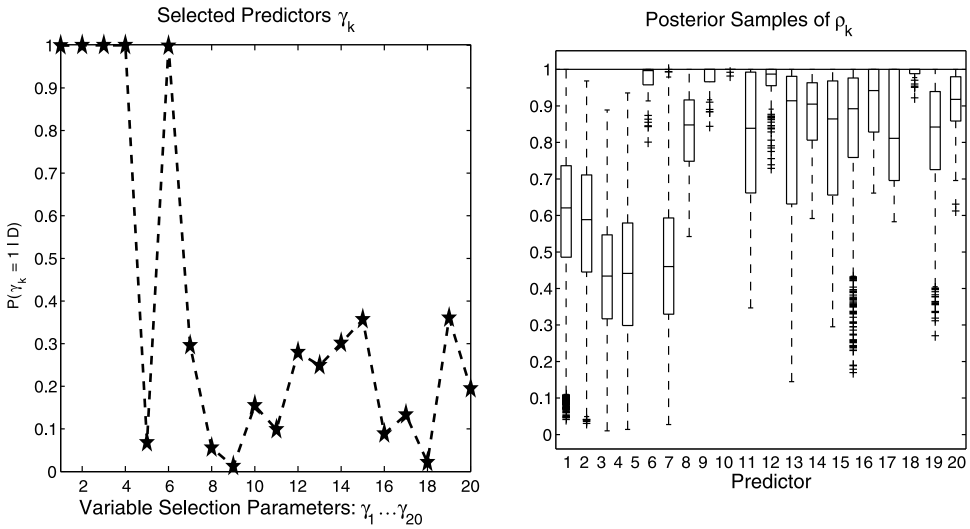


FIG. 3. *Cox GP model with large* p: *Simulated data* ($n = 100$, $p = 1{,}000$). *Posterior distributions for* $\gamma_k = 1$ *and box plots of posterior samples for* ρ_k.

on the full likelihood of Kalbfleisch (1978). The partial likelihood model formulation produced more consistent results across multiple chains, with the same data, and was able to detect much weaker signals. The Kalbfleisch (1978) model did, however, produce lower posterior values near 0 for nonselected covariates, unlike the partial likelihood formulation, which shows values typically from 10–40%, pointing to a potential bias toward false positives.

Additional simulations, including larger sample sizes cases, are reported in Savitsky (2010).

6.4 Comparison of MCMC Methods

We compare the 2 MCMC schemes previously described for posterior inference on $(\boldsymbol{\gamma}, \boldsymbol{\rho})$ on the basis of sampling and computational efficiency. We use the univariate regression simulation kernel

$$y = x_1 + 0.8x_2 + 1.3x_3 + \sin(x_4) + \sin(3x_5) + \sin(5x_6) + (1.5x_7)(1.5x_8) + \varepsilon,$$

with $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ and $\sigma = 0.05$. We utilize 1,000 covariates with all but the first 8 defined as nuisance. We use a training and a validation set of 100 observations each.

The two schemes differ in the way they update $(\boldsymbol{\gamma}, \boldsymbol{\rho})$. While scheme 1 samples either one or two positions in the model space on each iteration, scheme 2 samples (γ_k, ρ_k) for each of the p covariates. Because of this a good "rule-of-thumb" should employ a number of iterations for scheme 1 which is roughly p times the number of iterations employed for scheme 2. The use of the *Keep* move in scheme 1, however, reduces the need of scaling the number of iterations by exactly p, since all ρ_k's are sampled at each iteration. In our simulations we found stable convergence under moderate correlation among covariates for scheme 2 in 5,000 iterations and for scheme 1 in 500,000 iterations. For both schemes, we discarded half of the iterations as burn-in. The CPU run times we report in Table 2 are based on utilization of Matlab with a 2.4 GHz Quad Core (Q6600) PC with 4 GB of RAM running 64-bit Windows XP.

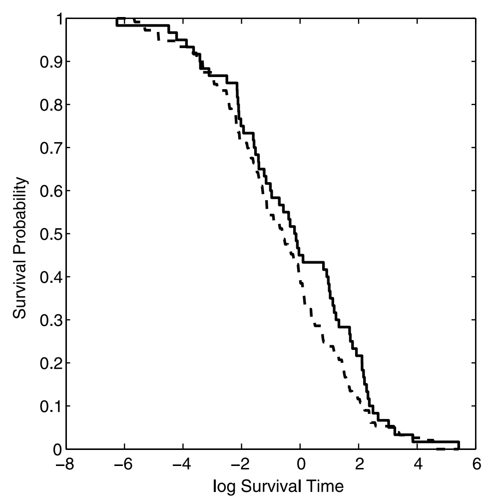


FIG. 4. *Cox GP model with large* p: *Simulated data* ($n = 100$, $p = 1{,}000$). *Average survivor function curve on the validation set* (*dashed line*) *compared to the Kaplan–Meier empirical estimate* (*solid line*).

TABLE 2
Efficiency comparison of GP MCMC methods

	MCMC scheme 2		MCMC scheme 1
	Adaptive	**Nonadaptive**	
Iterations (computation)	5,000	5,000	500,000
Autocorrelation time			
ρ_6	310	82	441
ρ_8	59	35	121
Computation			
CPU-time (sec)	980	4,956	10,224

We compared sampling efficiency looking at autocorrelation for selected ρ_k. The autocorrelation time is defined as one plus twice the sum of the autocorrelations at all lags and serves as a measure of the relative dependence for MCMC samples. We used the number of MCMC iterations divided by this factor as an "effective sample size." We followed a procedure outlined by Neal (2000) and ran first scheme 2 for 1,000 iterations, to obtain a state near the posterior distribution. We then employed this state to initiate a chain for each of the two schemes. We ran scheme 2 for an additional 2,000 iterations and scheme 1 for 200,000 (using the last 2,000 draws for each of the target ρ_k for final comparison). For scheme 2 we used both the adaptive and nonadaptive versions. Table 2 reports results for ρ_8, aligned to a covariate expressing a linear interaction, and for ρ_6, for a highly nonlinear interaction. We observe that both versions of scheme 2 express notable improvements in computational efficiency as compared to scheme 1. We note, however, that the adaptive scheme method produces draws of higher autocorrelation than the nonadaptive method.

6.5 Sensitivity Analysis

We begin with a sensitivity analysis on the prior for $\rho_k|\gamma_k = 1$. Table 3 shows results under a full factorial combination for hyperparameters (a, b) of a Beta prior construction, where we recall $\text{Beta}(1, 1) \equiv \mathcal{U}(0, 1)$. Results were obtained with the univariate regression simulation kernel

$$y = x_1 + x_2 + \sin(1.5x_3)\sin(1.5x_4) + \sin(3x_5) + \sin(3x_6) + (1.5x_7)(1.5x_8) + \varepsilon,$$

with $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ and where we employed a higher error variance of $\sigma = 0.28$. As before, we employ 1,000 covariates with all but the first 8 defined as nuisance. A training sample of 110 was simulated, along with a test set of 100 observations. We employed the adaptive scheme 2, with 5,000 iterations, half discarded as burn-in.

Figure 5 shows box plots of posterior samples for ρ_k for two symmetric alternatives, $1 : (a, b) = (0.5, 0.5)$ (U-shaped) and $2 : (a, b) = (2.0, 2.0)$ (symmetric unimodal). For scenario 2 we observe a reduction in posterior jitter on nuisance covariates and a stabilization of posterior sampling for associated covariates, but also a greater tendency to exclude x_3, x_4. One would expect the differences in posterior sampling behavior across prior hyperparameter values to decline as the sample size increases. Table 3 displays the number of nonselected true variables (false negatives), out of 8, along with the normalized MSPEs for all scenarios. There were no false positives to report across all hyperparameter settings. Overall, results are similar across the chosen settings for (a, b), with slightly better performances for $a < 1$ and $b \geq 1$, corresponding to strictly decreasing shapes that aid selection by pushing more mass away from 1, increasing the prior probability of the good variables to be selected, especially in the presence of a large number of noisy variables.

Next we imposed a Beta distribution on the hyperparameter α of the priors $\gamma_k \sim \text{Bernoulli}(\alpha)$ for covariate inclusion. We follow Brown, Vannucci and Fearn (1998a) to specify a vague prior by setting the mean of

TABLE 3
Prior sensitivity for $\rho_k|\gamma_k = 1 \sim \text{Beta}(a, b)$. Results are reported as (number of false negatives)/(normalized MSPE)

$b\backslash a$	0.5	1.0	2.0
0.5	2/0.18	2/0.15	2/0.18
1.0	1/0.14	1/0.16	2/0.18
2.0	1/0.15	2/0.16	2/0.17

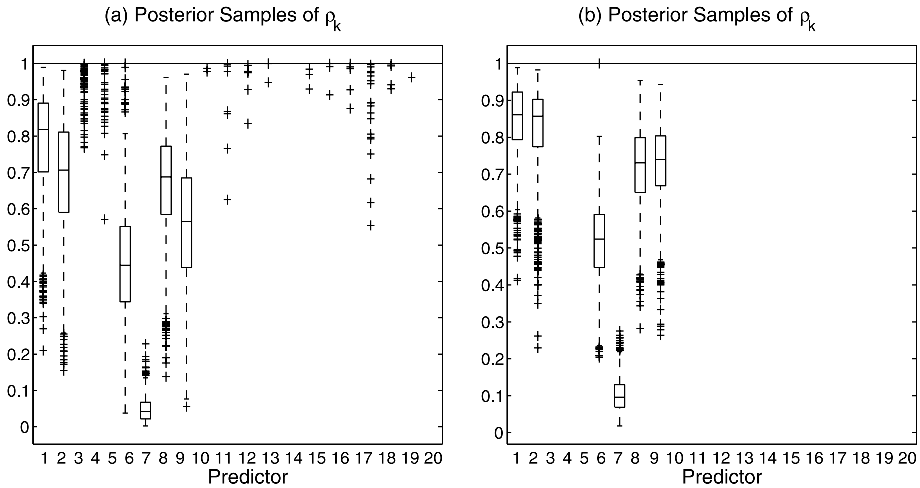


FIG. 5. *Prior Sensitivity for* $\rho_k|\gamma_k = 1 \sim \text{Beta}(a, b)$: *Box plots of posterior samples for* ρ_k *for* $(a, b) = (0.5, 0.5)$—*plot* (a)—*and* $(a, b) = (2.0, 2.0)$—*plot* (b).

the Beta prior to 0.025, reflecting a prior expectation for model sparsity, and the sum of the two parameters of the distribution to 2. We ran the same univariate regression simulation kernel as above with the hyperparameter settings for the Beta prior on ρ_k equal to $(1, 1)$ and obtained the same selection results as in the case of α fixed and a slightly lower normalized MSPE of 0.14.

Last, we explored performances with respect to correlation among the predictors. We utilized the same kernel as above with 8 true predictors from which to construct the response. We then induced a 70% correlation among 20 randomly chosen nuisance covariates and the true predictor x_6. We found 2 false negatives and 1 false positive, which demonstrates a relative selection robustness under correlation. We did observe a significant decline in normalized MSPE, however, to 0.33, as compared to previous runs.

7. BENCHMARK DATA APPLICATIONS

We now present results on two data sets often used in the literature as benchmarks. For both analyses we performed inference by using the MCMC—scheme 2, with 5,000 iterations and half discarded as burn-in.

7.1 Ozone data

We start by revisiting the ozone data, first analyzed for variable selection by Breiman and Friedman (1985) and more recently by Liang et al. (2008). This data set supplies integer counts for the maximum number of ozone particles per one million particles of air near Los Angeles for $n = 330$ days and includes an associated set of 8 meteorological predictors. We held out a randomly chosen set of 165 observations for validation.

Liang et al. (2008) use a linear regression model including all linear and quadratic terms for a total of $p = 44$ covariates. They achieve variable selection by imposing a mixture prior on the vector $\boldsymbol{\beta}$ of regression coefficients and specifying a g-prior of the type $\boldsymbol{\beta}_{\boldsymbol{\gamma}}|\phi \sim \mathcal{N}(\mathbf{0}, \frac{g}{\phi}(\mathbf{X}_{\boldsymbol{\gamma}}^T\mathbf{X}_{\boldsymbol{\gamma}})^{-1})$. Their results are reported in Table 4 with various formulations for g. In particular, the local empirical Bayes method offers a model-dependent maximizer of the marginal likelihood on g, while the hyper-g formulation with $a = 4$ is one member of a continuous set of hyper-prior distributions on the shrinkage factor, $g/(1 + g) \sim \text{Beta}(1, a/2 - 1)$. Since the design matrix expresses a high condition number, a situation that can at times induce poor results with g-priors, we additionally applied the method of Brown, Vannucci and Fearn (2002) who used a mixture prior of the type $\boldsymbol{\beta}_{\boldsymbol{\gamma}} \sim \mathcal{N}(\mathbf{0}, c\mathbf{I})$. Results shown in Table 4 were obtained from the Matlab code made available by the authors.

Though previous variable selection work on the ozone data all choose a Gaussian likelihood, a more precise approach employs a discrete Poisson or negative binomial formulation on data with low count values, or a log-normal approximation where counts are high. With a maximum value of 38 and a mean of 11 we chose to model the data with the negative-binomial count data model (5). We used the same hyperparameter settings as in our simulation study. Results are shown in Figure 6. By selecting, for example, the best 3 variables, we achieve a notable decrease in the root-MSPE as compared to the linear models. Also, by al-

TABLE 4
Ozone data: Results

Prior on g	$\mathcal{M}_\gamma$	p_γ	**RMSPE**
Local empirical Bayes	$X_5, X_6, X_7, X_6^2, X_7^2, X_3X_5$	6	4.5
Hyper-g ($a = 4$)	$X_5, X_6, X_7, X_6^2, X_7^2, X_3X_5$	6	4.5
Fixed (BIC)	$X_5, X_6, X_7, X_6^2, X_7^2, X_3X_5$	6	4.5
Brown, Vannucci and Fearn (2002)	$X_1X_6, X_1X_7, X_6X_7, X_1^2, X_3^2, X_7^2$	6	4.5
GP model	X_3, X_6, X_7	3	3.7

lowing an a priori unspecified functional form for how covariates relate to the response, we end up selecting a much more parsimonious model, although, of course, we lose in interpretability of the selected terms, with respect to linear formulations that specifically include linear, quadratic and interactions terms in the model.

7.2 Boston Housing data

Next we utilize the Boston Housing data set, also analyzed by Breiman and Friedman (1985), who used an additive model and employed an algorithm to empirically determine the functional relationship for each predictor. This data set relates $p = 13$ predictors to the median value of owner-occupied homes in each of $n = 506$ census tracts in the Boston metropolitan area. As with the previous data set, we held out a random set of 250 observations to assess prediction.

We employed the continuous data model (3) with the same hyperparameter settings as in our simulations. The four predictors chosen by Breiman and Friedman (1985), $(x_6, x_{10}, x_{11}, x_{13})$, had all marginal posterior probability of inclusion greater than 0.9 in our model. Other variables with high marginal posterior probability were (x_5, x_7, x_8, x_{12}). The adaptability of the GP response surface is illustrated with closer examination of covariate x_5, which measures the level of nitrogen oxide (NOX), a pollutant emitted by cars and factories. At low levels, indicating proximity to jobs, x_5 presents a positive association to the response, and at high levels, indicating overly industrialized areas, a negative association. This inverted parabolic association over the covariate range probably drove its exclusion in the model of Breiman and Friedman (1985). The GP formulation is, however, able to capture this strong nonlinear relationship as is noted in Figure 7. By using only the subset of the best eight predictors, we achieved a normalized MSE of 0.1 and a prediction R^2 of 0.9, very close to the value of 0.89 reported by Breiman and Friedman (1985) on the training data.

We also employed the Matern covariance construction (9), which we recall employs an explicit smoothing parameter, $\nu \in [0, \infty)$. While selection results were

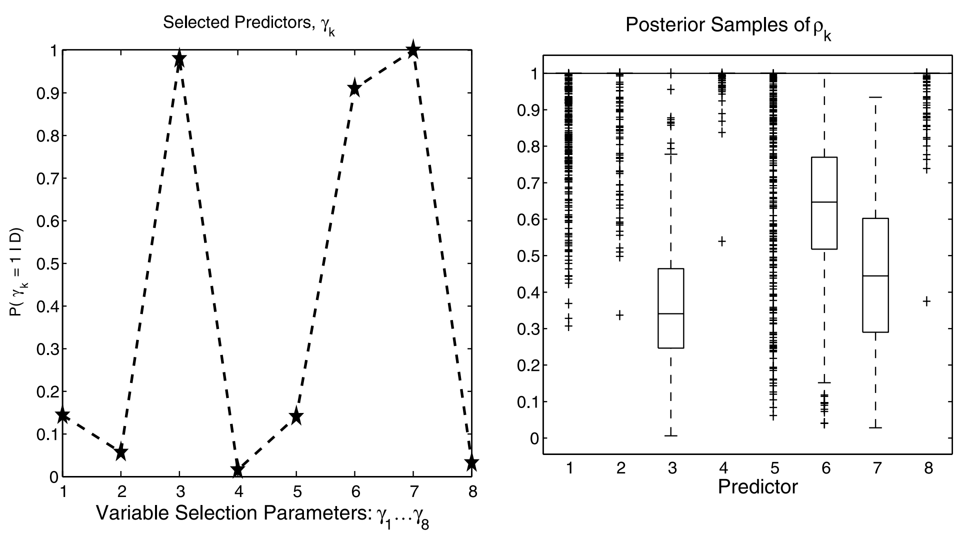


FIG. 6. *Ozone data: Posterior distributions for $\gamma_k = 1$ and box plots of posterior samples for ρ_k.*

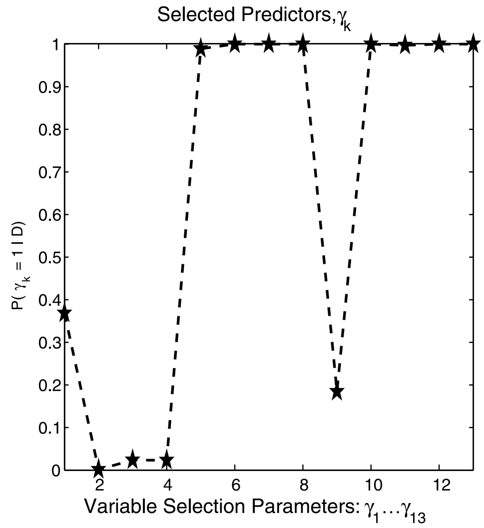


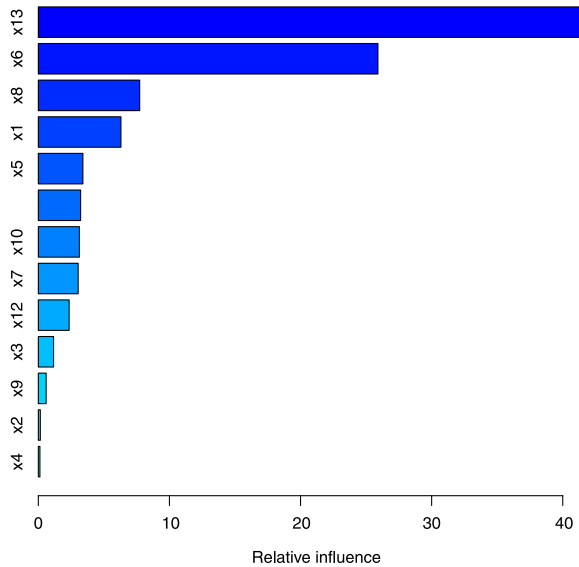
FIG. 7. *Boston housing data*: *Posterior distributions for $\gamma_k = 1$ and box plots of posterior samples for ρ_k*.

roughly similar, the prediction results for the Matern model were significantly worse than the exponential model, with a normalized MSPE of 0.16, probably due to overfitting. It is worth noticing that the more complex form for the Bessel function increases the CPU computation time by a factor of 5–10 under the Matern covariance as compared to the exponential construction.

For comparison, we looked at GBMs. We used version 3.1 of the **gbm** package for the R software environment. We utilized the same training and validation data as above. After experimentation and use of 10-fold cross-validation, we chose a small value for the input regularization parameter, $\nu = 0.0005$, to provide a smoother fit that prevents overfitting. Larger values of ν resulted in higher prediction errors. The GBM was run for 50,000 iterations to achieve minimum fit error. The result provided a normalized MSPE of 0.13 on the test set, similar to, though slightly higher than, the GP result. The left-hand chart of Figure 8 displays the relative covariate importance. Higher values correspond to (x_{13}, x_6, x_8), and agree with our GP results. A number of other covariates show similar importance values to one another, though lower than these top 3, making it unclear as to whether they are truly related or nuisance covariates. Similar conclusions are reported by other authors. For example, Tokdar, Zhu and Ghosh (2010) analyze a subset of the same data set with a Bayesian density regression model based on logistic Gaussian processes and subspace projections and found (x_{13}, x_6) as the most influential predictors, with a number of others having a mild influence as well. The right-hand plot supplies a partial dependence plot obtained by the GBM for variable x_{13} by averaging over the associations for the other covariates. We note that the nonlinear association is not constrained to be smooth under GBM.

8. DISCUSSION

In this paper we have presented a unified modeling approach via Gaussian processes that extends to data from the exponential dispersion family and to survival data. Such model formulation allows for nonlinear associations of the predictors to the response. We have considered, in particular, continuous, categorical and count responses and survival data. Next we have addressed the important problem of selecting variables from a set of possible predictors and have put forward a general framework that employs Bayesian variable selection methods and mixture priors for the selection of the predictors. We have investigated strategies for posterior inference and have demonstrated performances on simulated and benchmark data. GP models provide a parsimonious approach to model formulation with a great degree of freedom for the data to define the fit. Our results, in particular, have shown that GP models can achieve good prediction performances without the requirement of prespecifying higher order and nonlinear additive functions of the predictors. The benchmark data applications have shown that a GP formulation may be appropriate in cases of *heterogeneous* covariates, where the inability to employ an obvious transformation would require higher order polynomial terms in an additive linear fashion, or even in the case



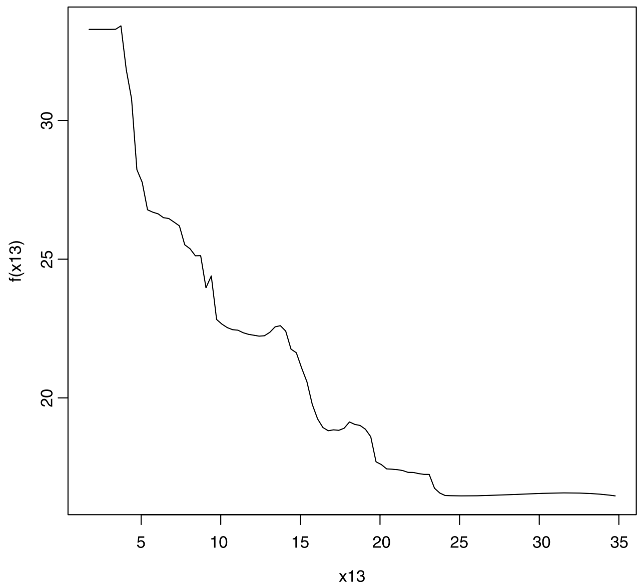


FIG. 8. *Boston housing data: GBM covariate analysis. Left-hand chart provides variables importance, normalized to sum up to* 100. *Right-hand plot enumerates partial association of* x_{13} *to the response.*

of a homogeneous covariate space where the transformation overly reduces structure in the data. Our simulation results have further highlighted the ability of the GP formulation to manage data sets with $p \gg n$.

A challenge in the use of variable selection methods in the GP framework is to manage the numerical instability in the construction of the GP covariance matrix. In the Appendix we describe a projection method to reduce the effective dimension of this matrix. Another practical limitation of the models we have described is the difficulty to use them with qualitative predictors. Qian, Wu and Wu (2008) provide a modification of the GP covariance kernel that allows for nominal qualitative predictors consisting of any number of levels. In particular, the authors model the covariance structure under a mixture of qualitative and quantitative predictors by employing a multiplicative factor against the usual GP kernel for each qualitative predictor to capture the by-level categorical effects.

Some generalization of the methods we have presented are possible. For example, as with GLM models, we may employ an additional set of variance inflation parameters in a similar construction to Neal (1999) and others to allow for heavier tailed distributions while maintaining the conjugate framework.

APPENDIX: COMPUTATIONAL ASPECTS

We focus on the exponential form (7) and introduce an efficient computational algorithm to generate $\mathbf{C}$. We also review a method of Banerjee et al. (2008) to approximate the inverse matrix that employs a random subset of observations and provide a pseudo-code.

A.1 Generating the Covariance Matrix C

Let us begin with the quadratic expression, $\mathbf{G} = \{g_{i,j}\}$ in (7). We rewrite $g_{i,j} = A'_{i,j}[-\log(\boldsymbol{\rho})]$ with $A_{i,j}$ constructed as a $p \times 1$ vector of term-by-term squared differences, $(x_{ik} - x_{jk})^2, k = 1, \dots, p$. We may directly employ the $p \times 1$ vector, $\boldsymbol{\rho}$, as $\mathbf{P}$ is diagonal. As a first step, we may then directly compute $\mathbf{G} = \mathbf{A}[-\log(\boldsymbol{\rho})]$, where $\mathbf{A}$ is $n \times n \times p$. We are, however, able to reduce the more complex structure of $\mathbf{A}$ to a two dimensional matrix form by simply stacking each $\{i, j\}$ row of dimension $1 \times p$ under each other such that our revised structure, $\mathbf{A}^*$, is of dimension $n^2 \times p$ and the computation, $\mathbf{G} = \mathbf{A}^*[-\log(\boldsymbol{\rho})]$, reduces to a series of inner products. Next, we note that $\log(\rho_k) = 0$ for $\rho_k = 1$. So we may reduce the dimension for each of the n^2 inner products by reducing the dimension of $\boldsymbol{\rho}$ to the $p_{\boldsymbol{\gamma}} < p$ nontrivial covariates. We may further improve efficiency by recognizing that since our resultant covariance matrix, $\mathbf{C}$, is symmetric positive definite, we need only compute the inner products for a reduced set of unique terms (by removing redundant rows from $\mathbf{A}^*$) and then "re-inflate" the result to a vector of the correct length. Finally, we exponentiate this vector, multiply the nonlinear weight $(1/\lambda_z)$, add the affine intercept term, $(1/\lambda_a)$, and then reshape this vector into the resulting $n \times n$ matrix, $\mathbf{C}$. The resulting improvement in computational efficiency at $n = 100$ from the naive approach that employs double loops of inner products is on the order of 500 times.

Our MCMC scheme 2 proposes a change to $\rho_k \in \boldsymbol{\rho}$, one-at-a-time, conditionally on $\boldsymbol{\rho}_{-k}$ and the other sampled parameters. Changing a single ρ_k requires updating only one column of the inner product computation of $\mathbf{A}^*$ and $[-\log(\boldsymbol{\rho})]$. Rather than conducting an entire recomputation for $\mathbf{C}$, we multiply the kth column of $\mathbf{A}^*$ (with number of rows reduced to only unique terms in $\mathbf{C}$) by $\log(\frac{\rho_{k,\text{prop}}}{\rho_{k,\text{old}}})$, where "prop" means the proposed value for ρ_k. This result is next exponentiated (to a covariance kernel), re-inflated and shaped into an $n \times n$ matrix, $\boldsymbol{\Delta}$. We then take the current value less the affine term, $\mathbf{C}_{\text{old}} - \frac{1}{\lambda_a}\mathbf{J}_n$, and multiply by $\boldsymbol{\Delta}$, term-by-term, and add back the affine term to achieve the new covariance matrix associated to the proposed value for ρ_k. So we may devise an algorithm to update an existing covariance matrix, C, rather than conducting an entire recomputation. At $p = 1{,}000$ with 6 nontrivial covariates and $n = 100$, this algorithm further reduces the computation time over recomputing the full covariance by a factor of 2. This efficiency grows nonlinearly with the number of nontrivial covariates.

A.2 Projection Method for Large *n*

In order to ease the computations, we have also adapted a dimension reduction method proposed by Banerjee et al. (2008) for spatial data. The method achieves a reduced-dimension computation of the inverse of the full ($n \times n$) covariance matrix. It can also help with the accuracy and stability of the posterior computations when working with possibly ill-conditioned GP covariance matrices, particularly for large n. To begin, randomly choose $m < n$ points (knots), sampled within fixed intervals on a grid to ensure relatively uniform coverage, and label these m points $\mathbf{z}^*$. Then define $\mathbf{z}_{m\to n}$ as the orthogonal projection of $\mathbf{z}$ onto the lower dimensional space spanned by $\mathbf{z}^*$, computed as the conditional expectation

$$\mathbf{z}_{m\to n} = \mathbb{E}(\mathbf{z}|\mathbf{z}^*) = \mathbf{C}'_{(\mathbf{z}^*,\mathbf{z})}\mathbf{C}^{-1}_{(\mathbf{z}^*,\mathbf{z}^*)}\mathbf{z}^*.$$

We use the univariate regression framework in (3) to illustrate the dimension reduction from constructing the *projection* model using $\mathbf{z}_{m\to n}$ in place of $\mathbf{z}(\mathbf{x})$. Recast the model from (3) to

$$\mathbf{y} = \mathbf{z}_{m\to n} + \boldsymbol{\varepsilon} = \mathbf{C}'_{(\mathbf{z}^*,\mathbf{z})}\mathbf{C}^{-1}_{(\mathbf{z}^*,\mathbf{z}^*)}\mathbf{z}^* + \boldsymbol{\varepsilon},$$

where $\varepsilon_i \sim \mathcal{N}(0, \frac{1}{r})$. Then derive $\boldsymbol{\Lambda}_n = \text{Cov}(\mathbf{y}) = \frac{1}{r}\mathbb{I}_n + \mathbf{C}'_{(\mathbf{z}^*,\mathbf{z})}\mathbf{C}^{-1}_{(\mathbf{z}^*,\mathbf{z}^*)}\mathbf{C}_{(\mathbf{z}^*,\mathbf{z})}$. Finally, employ the Woodbury matrix identity to transform the inverse computation, $\boldsymbol{\Lambda}_n^{-1} = r\mathbb{I} - r^2\mathbf{C}'_{(\mathbf{z}^*,\mathbf{z})}[\mathbf{C}_{(\mathbf{z}^*,\mathbf{z}^*)} + r\mathbf{C}_{(\mathbf{z}^*,\mathbf{z})} \cdot \mathbf{C}'_{(\mathbf{z}^*,\mathbf{z})}]^{-1}\mathbf{C}_{(\mathbf{z}^*,\mathbf{z})}$, where the quantity inside the square brackets, now being inverted, is $m \times m$, supplying the dimension reduction for inverse computation we seek. We note that, in the absence of the projection method, a large jitter term would be required to invert the GP covariance matrix, trading accuracy for stability. Though the projection method approximates a higher dimensional covariance matrix in a lower dimensional projection, we yet improve performance and avoid the accuracy/stability trade-off. We do, however, expect to use more iterations for MCMC convergence when employing a relatively lower projection ratio.

All results shown in this paper were obtained with $m/n = 0.35$, for simulated data, and with $m/n = 0.25$, for the benchmark applications, where we enhanced computation stability in the presence of the high condition number for the design matrix. We have also employed the Cholesky decomposition, in a similar fashion as in Neal (1999), in lieu of directly computing the resulting $m \times m$ inverse.

A.3 Pseudo-code

Procedure to Compute, $\mathbf{C} = \frac{1}{\lambda_a}\mathbf{J}_n + \frac{1}{\lambda_z}\exp(-\mathbf{G})$:

Input: data matrices;
  $(\mathbf{X}_1, \mathbf{X}_2)$ of dimension $(n_1, n_2) \times p$
Output: function, $[A^*, I_{\text{full}}] =$ **difference**$(\mathbf{X}_1, \mathbf{X}_2)$
% A^* is matrix of squared L_2 distances
  for 2 data matrices of p columns
% A^* size, $\ell \times p$, $\ell \le n_1 n_2$: only unique entries
% I_{full} re-inflates A^* with duplicate entries
% Key point: Compute A^*, once,
  and re-use in GP posterior computations
% Set counter to stack all (i, j) obs
  from $\mathbf{X}_1, \mathbf{X}_2$ in vectorized construction
    count = 1;
% Compute squared distances
    FOR $i = 1$ to n_1
      FOR $j = 1$ to n_2
        $A^*_{\text{full}}(\text{count},:) = (x_{1,i} - x_{2,j})^2$;
        count = count + 1;
      END
    END
% Reduce A^*_{full} to A^*
    $[A^*, I_{\text{full}}] = \text{unique}(A^*_{\text{full}}, \text{by row})$;
END FUNCTION

Input: Data = (A^*, I_{full}), $\boldsymbol{\Theta} = (\boldsymbol{\rho}, \lambda_a, \lambda_z)$
Output: function, $[\mathbf{C}] = \text{C}(A^*, I_{\text{full}}, \boldsymbol{\Theta})$
% An $n_1 \times n_2$ GP covariance matrix
% Only compute inner product
  for column k where $\rho_k < 1$

$sel_\rho = \{\rho_k < 1\};$
$\boldsymbol{\rho} = \boldsymbol{\rho}(sel_\rho)$
$A^* = A^*(:, sel_\rho);$
% Compute vector of unique values for **C**
$-G_{\text{vec}} = A^*[\log(\boldsymbol{\rho})]';$
$C_{\text{vec}} = \frac{1}{\lambda_a} + \frac{1}{\lambda_z}\exp(-G_{\text{vec}});$
% Re-inflate C_{vec} to include duplicate values
$C_{\text{vec}} = C_{\text{vec}}(I_{\text{full}});$
% Snap C_{vec} into matrix form, C
$C = \text{reshape}(C_{\text{vec}}, n_2, n_1)';$
END FUNCTION

Input: Previous covariance $= C_{\text{old}}$;
Data $= (A^*, I_{\text{full}})$; Position changed $= k$,
Parameters $= (\rho_{k,\text{new}}, \rho_{k,\text{old}})$, Intercept $= \lambda_a$
Output:$[\mathbf{C}_{\text{new}}] = C_{\text{partial}}(C_{\text{old}}, A^*, I_{\text{full}}, k, \lambda_a)$
% Compose new covariance matrix, $\mathbf{C}_{\text{new}}$,
from old, $\mathbf{C}_{\text{old}}$
% Compute inner products only for row k of A^*
% Produce matrix of multiplicative differences
from old to new
$-\boldsymbol{\Delta}\mathbf{G}_{\text{vec}} = \mathbf{A}^*(:, k) \times \log(\frac{\rho_{k,\text{new}}}{\rho_{k,\text{old}}});$
% Re-inflate $\exp(-\boldsymbol{\Delta}\mathbf{G}_{\text{vec}})$
$\exp(-\boldsymbol{\Delta}\mathbf{G}_{\text{vec}}) = \exp(-\boldsymbol{\Delta}\mathbf{G}_{\text{vec}})(I_{\text{full}});$
% Re-shape $-\boldsymbol{\Delta}\mathbf{G}_{\text{vec}}$ to matrix, $\boldsymbol{\Delta}$
$\boldsymbol{\Delta} = \text{reshape}(\exp[-\boldsymbol{\Delta}\mathbf{G}_{\text{vec}}], n_2, n_1)';$
% Compute C_{new}
$\mathbf{C}_{\text{new}} = \frac{1}{\lambda_a}\mathbf{J}_n + (\mathbf{C}_{\text{old}} - \frac{1}{\lambda_a}\mathbf{J}_n) \odot \boldsymbol{\Delta};$
END FUNCTION
Procedure to Compute Inverse of $\boldsymbol{\Lambda}_\mathbf{n} = \frac{1}{r}\mathbb{I}_n + \mathbf{C}$:
Input: Number of sub-sample $= m$, Data $= \mathbf{X}$,
Error precision $= r$
Covariance parameters = $\boldsymbol{\Theta} = (\boldsymbol{\rho}, \lambda_a, \lambda_z)$
Output: $\boldsymbol{\Lambda}_\mathbf{n}^{-1}$
% Randomly select $m < n$ observations
on which to project $n \times 1$, $z(x)$
$ind = \text{random.permutations.latin.hypercube}(n);$
% space-filling
$\mathbf{X}_m = \mathbf{X}(ind(1:m), :);$
% Compute squared distances, A_m^*, A^*
$[A_m^*, I_{m,\text{full}}] = \text{difference}(\mathbf{X}_m, \mathbf{X}_m);$ % $m \times n$
$[A^*, I_{\text{full}}] = \text{difference}(\mathbf{X}_m, \mathbf{X});$ % $n \times n$
% Compose associated covariance matrices
$C_{(m,m)} = C(A_m^*, I_{m,\text{full}}, \boldsymbol{\Theta});$
$C_{(m,n)} = C(A^*, I_{\text{full}}, \boldsymbol{\Theta});$
% Compute $\boldsymbol{\Lambda}_\mathbf{n}$
$\boldsymbol{\Lambda}_\mathbf{n} = \frac{1}{r}\mathbb{I}_n + C'_{(m,n)}C^{-1}_{(m,m)}C_{(m,n)};$
% Compute $\boldsymbol{\Lambda}_\mathbf{n}^{-1}$ employing
term-by-term multiplication
$\boldsymbol{\Lambda}_\mathbf{n}^{-1} = r\mathbb{I}_n - r^2 C'_{(m,n)}[C_{(m,m)}$
$+rC_{(m,n)}C'_{(m,n)}]^{-1}C_{(m,n)};$
END

ACKNOWLEDGMENTS

Marina Vannucci supported in part by NIH-NHGRI Grant R01-HG003319 and by NSF Grant DMS-10-07871. Naijum Sha supported in part by NSF Grant CMMI-0654417. Terrance Savitsky was supported under NIH Grant NCI T32 CA096520 while at Rice University.

REFERENCES

ALBERT, J. H. and CHIB, S. (1993). Bayesian analysis of binary and polychotomous response data. *J. Amer. Statist. Assoc.* **88** 669–679. MR1224394

BANERJEE, S., GELFAND, A. E., FINLEY, A. O. and SANG, H. (2008). Gaussian predictive process models for large spatial data sets. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **70** 825–848. MR2523906

BISHOP, C. M. (2006). *Pattern Recognition and Machine Learning. Information Science and Statistics*. Springer, New York. MR2247587

BREIMAN, L. and FRIEDMAN, J. H. (1985). Estimating optimal transformations for multiple regression and correlation (with discussion). *J. Amer. Statist. Assoc.* **80** 580–619. MR0803258

BROWN, P. J., VANNUCCI, M. and FEARN, T. (1998a). Bayesian wavelength selection in multicomponent analysis. *J. Chemometrics* **12** 173–182.

BROWN, P. J., VANNUCCI, M. and FEARN, T. (1998b). Multivariate Bayesian variable selection and prediction. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **60** 627–641. MR1626005

BROWN, P. J., VANNUCCI, M. and FEARN, T. (2002). Bayes model averaging with selection of regressors. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **64** 519–536. MR1924304

CHEN, M.-H., IBRAHIM, J. G. and YIANNOUTSOS, C. (1999). Prior elicitation, variable selection and Bayesian computation for logistic regression models. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **61** 223–242. MR1664057

CHIPMAN, H., GEORGE, E. and MCCULLOCH, R. (2001). Practical implementation of Bayesian model selection. In *Model Selection* (P. Lahiri, ed.) 65–134. IMS, Beachwood, OH. MR2000752

CHIPMAN, H., GEORGE, E. and MCCULLOCH, R. (2002). Bayesian treed models. *Machine Learning* **48** 303–324.

COX, D. R. (1972). Regression models and life-tables (with discussion). *J. Roy. Statist. Soc. Ser. B* **34** 187–220. MR0341758

DIGGLE, P. J., TAWN, J. A. and MOYEED, R. A. (1998). Model-based geostatistics (with discussion). *J. Roy. Statist. Soc. Ser. C* **47** 299–350. MR1626544

FAHRMEIR, L., KNEIB, T. and LANG, S. (2004). Penalized structured additive regression for space-time data: A Bayesian perspective. *Statist. Sinica* **14** 731–761. MR2087971

FREUND, Y. and SCHAPIRE, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. System Sci.* **55** 119–139. MR1473055

GEORGE, E. I. and MCCULLOCH, R. E. (1993). Variable selection via Gibbs sampling. *J. Amer. Statist. Assoc.* **88** 881–889.

GEORGE, E. I. and MCCULLOCH, R. E. (1997). Approaches for Bayesian variable selection. *Statist. Sinica* **7** 339–373.

GOTTARDO, R. and RAFTERY, A. E. (2008). Markov chain Monte Carlo with mixtures of mutually singular distributions. *J. Comput. Graph. Statist.* **17** 949–975. MR2479418

GRAMACY, R. B. and LEE, H. K. H. (2008). Bayesian treed Gaussian process models with an application to computer modeling. *J. Amer. Statist. Assoc.* **103** 1119–1130. MR2528830

GREEN, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika* **82** 711–732. MR1380810

GUAN, Y. and STEPHENS, M. (2011). Bayesian variable selection regression for genome-wide association studies, and other large-scale problems. *Ann. Appl. Stat.* To appear.

HAARIO, H., SAKSMAN, E. and TAMMINEN, J. (2001). An adaptive metropolis algorithm. *Bernoulli* **7** 223–242. MR1828504

HASTIE, T., TIBSHIRANI, R. and FRIEDMAN, J. (2001). *The Elements of Statistical Learning*: *Data Mining*, *Inference*, *and Prediction*. Springer, New York. MR1851606

JI, C. and SCHMIDLER, S. (2009). Adaptive Markov chain Monte Carlo for Bayesian variable selection. Technical report.

KALBFLEISCH, J. D. (1978). Non-parametric Bayesian analysis of survival time data. *J. Roy. Statist. Soc. Ser. B* **40** 214–221. MR0517442

LEE, K. E. and MALLICK, B. K. (2004). Bayesian methods for variable selection in survival models with application to DNA microarray data. *Sankhyā* **66** 756–778. MR2205819

LIANG, F., PAULO, R., MOLINA, G., CLYDE, M. A. and BERGER, J. O. (2008). Mixtures of g priors for Bayesian variable selection. *J. Amer. Statist. Assoc.* **103** 410–423. MR2420243

LINKLETTER, C., BINGHAM, D., HENGARTNER, N., HIGDON, D. and YE, K. Q. (2006). Variable selection for Gaussian process models in computer experiments. *Technometrics* **48** 478–490. MR2328617

LONG, J. (1997). *Regression Models for Categorical and Limited Dependent Variables*. Sage, Thousand Oaks, CA.

MADIGAN, D. and YORK, J. (1995). Bayesian graphical models for discrete data. *Internat. Statist. Rev.* **63** 215–232.

MCCULLAGH, P. and NELDER, J. A. (1989). *Generalized Linear Models*, 2nd ed. Chapman & Hall, London.

NEAL, R. M. (1999). Regression and classification using Gaussian process priors. In *Bayesian Statistics 6* (A. P. Dawid, J. M. Bernardo, J. O. Berger and A. F. M. Smith, eds.) 475–501. Oxford Univ. Press, New York. MR1723510

NEAL, R. M. (2000). Markov chain sampling methods for Dirichlet process mixture models. *J. Comput. Graph. Statist.* **9** 249–265. MR1823804

O'HAGAN, A. (1978). Curve fitting and optimal design for prediction. *J. Roy. Statist. Soc. Ser. B* **40** 1–42. MR0512140

PANAGIOTELIS, A. and SMITH, M. (2008). Bayesian identification, selection and estimation of semiparametric functions in high-dimensional additive models. *J. Econometrics* **143** 291–316. MR2389611

PARZEN, E. (1963). Probability density functionals and reproducing kernel Hilbert spaces. In *Proc. Sympos. Time Series Analysis* (*Brown Univ.*, 1962) (M. Rosenblatt, ed.) 155–169. Wiley, New York. MR0149634

QIAN, P. Z. G., WU, H. and WU, C. F. J. (2008). Gaussian process models for computer experiments with qualitative and quantitative factors. *Technometrics* **50** 383–396. MR2457574

RAFTERY, A. E., MADIGAN, D. and VOLINSKY, C. T. (1996). Accounting for model uncertainty in survival analysis improves predictive performance. In *Bayesian Statistics 5*. *Oxford Sci. Publ.* 323–349. Oxford Univ. Press, New York. MR1425413

RASMUSSEN, C. E. and WILLIAMS, C. K. I. (2006). *Gaussian Processes for Machine Learning*. *Adaptive Computation and Machine Learning*. MIT Press, Cambridge, MA. MR2514435

ROBERTS, G. O. and ROSENTHAL, J. S. (2007). Coupling and ergodicity of adaptive Markov chain Monte Carlo algorithms. *J. Appl. Probab.* **44** 458–475. MR2340211

RUPPERT, D., WAND, M. P. and CARROLL, R. J. (2003). *Semiparametric Regression*. *Cambridge Series in Statistical and Probabilistic Mathematics* **12**. Cambridge Univ. Press, Cambridge. MR1998720

SACKS, J., SCHILLER, S. B. and WELCH, W. J. (1989). Designs for computer experiments. *Technometrics* **31** 41–47. MR0997669

SAVITSKY, T. D. (2010). Generalized Gaussian process models with Bayesian variable selection. Ph.D. thesis, Dept. Statistics, Rice Univ. MR2782375

SHA, N., TADESSE, M. G. and VANNUCCI, M. (2006). Bayesian variable selection for the analysis of microarray data with censored outcomes. *Bioinformatics* **22** 2262–2268.

SHA, N., VANNUCCI, M., TADESSE, M. G., BROWN, P. J., DRAGONI, I., DAVIES, N., ROBERTS, T. C., CONTESTABILE, A., SALMON, M., BUCKLEY, C. and FALCIANI, F. (2004). Bayesian variable selection in multinomial probit models to identify molecular signatures of disease stage. *Biometrics* **60** 812–828. MR2089459

SHAWE-TAYLOR, J. and CRISTIANINI, N. (2004). *Kernel Methods for Pattern Analysis*. Cambridge Univ. Press, Cambridge.

SINHA, D., IBRAHIM, J. G. and CHEN, M.-H. (2003). A Bayesian justification of Cox's partial likelihood. *Biometrika* **90** 629–641. MR2006840

THRUN, S., SAUL, L. K. and SCHOLKOPF, B. (2004). *Advances in Neural Information Processing Systems*. MIT Press, Cambridge.

TOKDAR, S., ZHU, Y. and GHOSH, J. (2010). Bayesian density regression with logistic Gaussian process and subspace projection. *Bayesian Anal.* **5** 319–344.

VOLINSKY, C., MADIGAN, D., RAFTERY, A. and KRONMAL, R. (1997). Bayesian model averaging in proportional hazard models: Assessing the risk of stroke. *Appl. Statist.* **46** 433–448.

WAHBA, G. (1990). *Spline Models for Observational Data*. *CBMS-NSF Regional Conference Series in Applied Mathematics* **59**. SIAM, Philadelphia, PA. MR1045442

WENG, Y. P. and WONG, K. F. (2007). Baseline Survival Function Estimators under Proportional Hazards Assumption, Ph.D. thesis, Institute of Statistics, National Univ. Kaohsiung, Taiwan.