Taylor & Francis
Taylor & Francis Group

Journal of the American Statistical Association

ISSN: 0162-1459 (Print) 1537-274X (Online) Journal homepage: https://www.tandfonline.com/loi/uasa20

Bayesian Treed Gaussian Process Models With an Application to Computer Modeling

Robert B Gramacy & Herbert K. H Lee

To cite this article: Robert B Gramacy & Herbert K. H Lee (2008) Bayesian Treed Gaussian Process Models With an Application to Computer Modeling, Journal of the American Statistical Association, 103:483, 1119-1130, DOI: 10.1198/016214508000000689

To link to this article: https://doi.org/10.1198/016214508000000689

Published online: 01 Jan 2012.

Bayesian Treed Gaussian Process Models With an Application to Computer Modeling

Robert B. GRAMACY and Herbert K. H. LEE

Motivated by a computer experiment for the design of a rocket booster, this article explores nonstationary modeling methodologies that couple stationary Gaussian processes with treed partitioning. Partitioning is a simple but effective method for dealing with nonstationarity. The methodological developments and statistical computing details that make this approach efficient are described in detail. In addition to providing an analysis of the rocket booster simulator, we show that our approach is effective in other arenas as well.

KEY WORDS: Computer simulator; Heteroscedasticity; Nonparametric regression; Nonstationary spatial model; Recursive partitioning.

1. INTRODUCTION

Much of modern engineering design is now done through computer modeling, which is both faster and more cost-effective than building small-scale models, particularly in the earlier stages of design, when more scope for changes is desired. As design proceeds, increasingly sophisticated simulators may be created. Our work here was motivated by a simulator of a proposed rocket booster. NASA relies heavily on simulators for design, because wind tunnel experiments are quite expensive and still not fully realistic in terms of the range of flight experiences. In particular, one of the highly critical points in time for a rocket booster is the moment that it reenters the atmosphere. Such conditions are difficult to recreate in a wind tunnel, and it obviously is impossible to run a standard physical experiment. Thus NASA uses computer simulation to learn about the behavior of the proposed rocket booster.

Simulators can involve large amounts of physical modeling, requiring the numerical solution of complex systems of differential equations. The NASA simulator was no exception, typically requiring between 5 and 20 hours for a single run. Thus NASA was interested in creating a statistical model of the simulator itself, known as an *emulator* or *surrogate model*, in the terminology of computer modeling. The standard approach for emulation in the literature is to model the simulator output with a stationary smooth Gaussian process (GP) (Sacks, Welch, Mitchell, and Wynn 1989; Kennedy and O'Hagan 2001; Santner, Williams, and Notz 2003). But this approach proved inadequate for the NASA data. In particular, we were faced with problems of nonstationarity, heteroscedasticity, and the size of the data set. Thus here we introduce an expansion of GPs, based on the idea of Bayesian partition models (Chipman, George, and McCulloch 2002; Denison, Adams, Holmes, and Hand 2002), which can address these issues.

GPs are conceptually straightforward, can easily accommodate prior knowledge in the form of covariance functions, and can return estimates of predictive confidence, all features were desired by NASA. But here we highlight three disadvantages of the standard form of a GP that we had to confront on this data set and would expect to encounter in a wide range of other applications. First, inference on the GP scales poorly with the number of data points, N, typically requiring computing time in $O(N^3)$ for calculating inverses of $N \times N$ covariance matrixes. Second, GP models are usually stationary in that the same covariance structure is used throughout the entire input space, which may be too strong a modeling assumption. Third, the estimated predictive error (as opposed to the predictive mean value) of a stationary model does not depend directly on the locally observed response values; rather, the predictive error at a point depends only on the locations of the nearby observations and on a global measure of error that uses all of the discrepancies between observations and predictions without regard for their distance from the point of interest (because of the stationarity assumption). In Section 4.3 we provide more details, noting in particular that eq. (12) does not depend on $\mathbf{z}$. In many real-world spatial and stochastic problems, such a uniform modeling of uncertainty is not desirable. Instead, some regions of the space will tend to exhibit larger variability than others. On the other hand, fully nonstationary Bayesian GP models (e.g., Higdon, Swall, and Kern 1999; Schmidt and O'Hagan 2003) can be difficult to fit and are not computationally tractable for more than a relatively small number of data points. Further discussion of nonstationary models is deferred until the end of Section 3.2.

All of these shortcomings can be addressed by partitioning the input space into regions and fitting separate stationary GP models within each region (e.g., Kim, Mallick, and Holmes 2005). Partitioning provides a relatively straightforward mechanism for creating a nonstationary model and can ameliorate some of the computational demands by fitting models to less data. A Bayesian model-averaging approach allows for the explicit estimation of predictive uncertainty, which now can vary beyond the constraints of a stationary model. Finally, an R package with implementation of all of the models discussed in this article is available at *http://www.cran.r-project.org/web/packages/tgp/index.html*. We note that by partitioning, we do not have any theoretical guarantee of continuity in the fitted function; however, as we demonstrate in several examples, Bayesian model averaging yields mean fitted functions that typically are quite smooth in practice, giving fits that are indistinguishable from continuous functions except

Robert Gramacy is Lecturer, Statistical Laboratory, University of Cambridge, Cambridge, U.K. (E-mail: *bobby@statslab.cam.ac.uk*). Herbert Lee is Associate Professor, Department of Applied Mathematics and Statistics, University of California, Santa Cruz, CA 95064 (E-mail: *herbie@ams.ucsc.edu*).
The authors thank William Macready for originating the collaboration with NASA and for his help with the project, Thomas Pulliam and Edward Tejnil for their help with the NASA data, Tamara Broderick for her careful reading and detailed comments, and the editor, associate editor, and two anonymous referees for their helpful comments and suggestions. This work was supported in part by NASA awards 08008-002-011-000 and SC 2003028 NAS2-03144, Sandia National Labs grant 496420, and National Science Foundation grants DMS 0233710 and 0504851.

Journal of the American Statistical Association
September 2008, Vol. 103, No. 483, Applications and Case Studies
DOI 10.1198/016214508000000689

when the data call for the contrary. Indeed the ability to accurately model possible discontinuities is a side benefit of this approach.

The rest of the article is organized as follows. Section 2 describes the motivating data in further detail. Section 3 provides some background material. Section 4 combines stationary GPs and treed partitioning to create treed GPs, implementing a tractable nonstationary model for nonparametric regression. Section 5 returns to the analysis of the rocket booster data, and Section 6 concludes with a discussion.

2. THE LANGLEY GLIDE–BACK BOOSTER SIMULATION

The Langley glide-back booster (LGBB) is a proposed rocket booster currently under design at NASA. Standard rocket boosters are created to be reusable, assisting in the launch process and then parachuting back to Earth after their fuel has been exhausted. Their return path is planned so that they fall into the ocean, where they can be recovered and reused. The LGBB represents a new direction in booster design, having wings and a tail, appearing somewhat like the space shuttle. The idea is that it will gracefully glide back down rather than plummet into the ocean.

The booster is being developed primarily through the use of computer simulators. The particular model with which we are involved (Rogers et al. 2003) is based on computational fluid dynamics simulators that numerically solve the relevant inviscid Euler equations over a mesh of 1.4 million cells. Each run of the Euler solver for a given set of parameters can take 5–20 hours on the NASA computers. The simulator is theoretically deterministic, but the solver is typically started with random initial conditions and does not always numerically converge. There is an automated check for convergence that is mostly accurate, but some runs are marked as accepted despite their false convergence, or else they converge to a clearly inferior local mode. For those runs that fail the automated convergence check, the solver is restarted at a different set of randomly chosen initial conditions. Our NASA collaborators have commented that input configurations arbitrarily close to one another can fail to achieve the same estimated convergence, even after satisfying the same stopping criterion. Thus neither simple interpolation of the data nor a GP model without an error term is adequate, because smoothing is necessary to reduce the impact of the inaccurate runs.

The simulator models the forces felt by the vehicle at the moment that it is reentering the atmosphere. As a free body in space, there are 6 degrees of freedom, so the six relevant forces are lift, drag, pitch, side force, yaw, and roll. For this project, the interest focused solely on the lift force, the most important force for keeping a vehicle aloft. The inputs to the simulator are the speed of the vehicle at reentry (measured by Mach number), the angle of attack (the alpha angle), and the sideslip angle (the beta angle). Thus the primary goal is to model the lift force as a function of speed, alpha, and beta. The sideslip angle is quantized in the experiments, so it is run only at six particular levels. Speed ranges from Mach 0 to Mach 6, and the angle of attack, alpha, varies from −5 to 30 degrees. The simulator was run at 3,041 locations, over a combination of 3 hand-designed grids. The first grid was relatively coarse and spaced equally over the whole region of interest. Two successively finer grids on smaller regions primarily around Mach 1 were then run, because the initial run showed that the most interesting part of the input space was generally around the sound barrier. This makes sense, because the physics in the simulator comes from two completely different regimes, a subsonic regime for speeds less than Mach 1 and a supersonic regime for speeds greater than Mach 1. What happens close to and along the boundary is the most difficult part of the simulation.

The upper-left plot in Figure 1 shows an interpolation of the simulator output for the lift surface as a function of speed and angle of attack when the sideslip angle is 0. The primary feature of this plot is the large ridge that appears at Mach 1 and larger angles of attack. The transition from subsonic to supersonic is sharp, and whether we would want to use a continuous model or introduce a discontinuity is not clear. Although much of the surface is quite smooth, parts of the surface, particularly around Mach 1, are less smooth. Thus the standard computer modeling assumption of a stationary process clearly will not work well here. We need a method that allows for a nonstationary formulation, yet can still produce uncertainty estimates and is computationally feasible to fit on a data set of this size.

One other feature of the data shown in Figure 1 is numerical convergence. In the upper-left corner of the upper-left plot (high angle of attack, low speed), there is a single spike that looks out of place. Our collaborators at NASA believe this to be a result of a false convergence by the simulator, so we would want our surrogate model to smooth out this one point. This stands in contrast to most computer modeling problems, where we want to interpolate the deterministic simulator without smoothing. Here we require smoothing to compensate for problems with the simulator.

The other plots in Figure 1 show the issue of false convergence more clearly for other sideslip settings. In the center plots (levels .5 and 3), there are noisy depressions in the surface for moderate speed and high angle of attack. Because this feature is not seen in the other plots by sideslip angle, this region may be suspected to be showing more numerical instability than signal. Thus there is a need to combine information across the levels to smooth out numerical problems with the simulator. Note that because no subsonic inputs were sampled for these slices, the ridge around Mach 1 does not appear in these two plots.

For sideslip levels of 1, 2, and 4, the surface again appears to be most interesting around Mach 1. But instead of a clean ridge at levels 1 and 4, it is noisy, especially at high angles of attack. Whether this variability is due to false convergence of the simulator or inadequate coverage in the design, or whether the boundary really is this complicated, is not clear. The NASA scientists have postulated that the instabilities are more likely to be numerical rather than structural, but we want our surrogate model to capture this uncertainty.

Also of concern are the deviations from the smooth trend at high speeds (particularly for level 4), with upward deviations at low angles of attack and downward deviations at high angles of attack. Again, these are suspected to be the result of false numerical convergence of the simulator, but we cannot rule out a priori the possibility that the physical system itself becomes unstable at higher speeds. Thus we desire smoothing, but with an appropriate local estimate of uncertainty. Fitting with a single

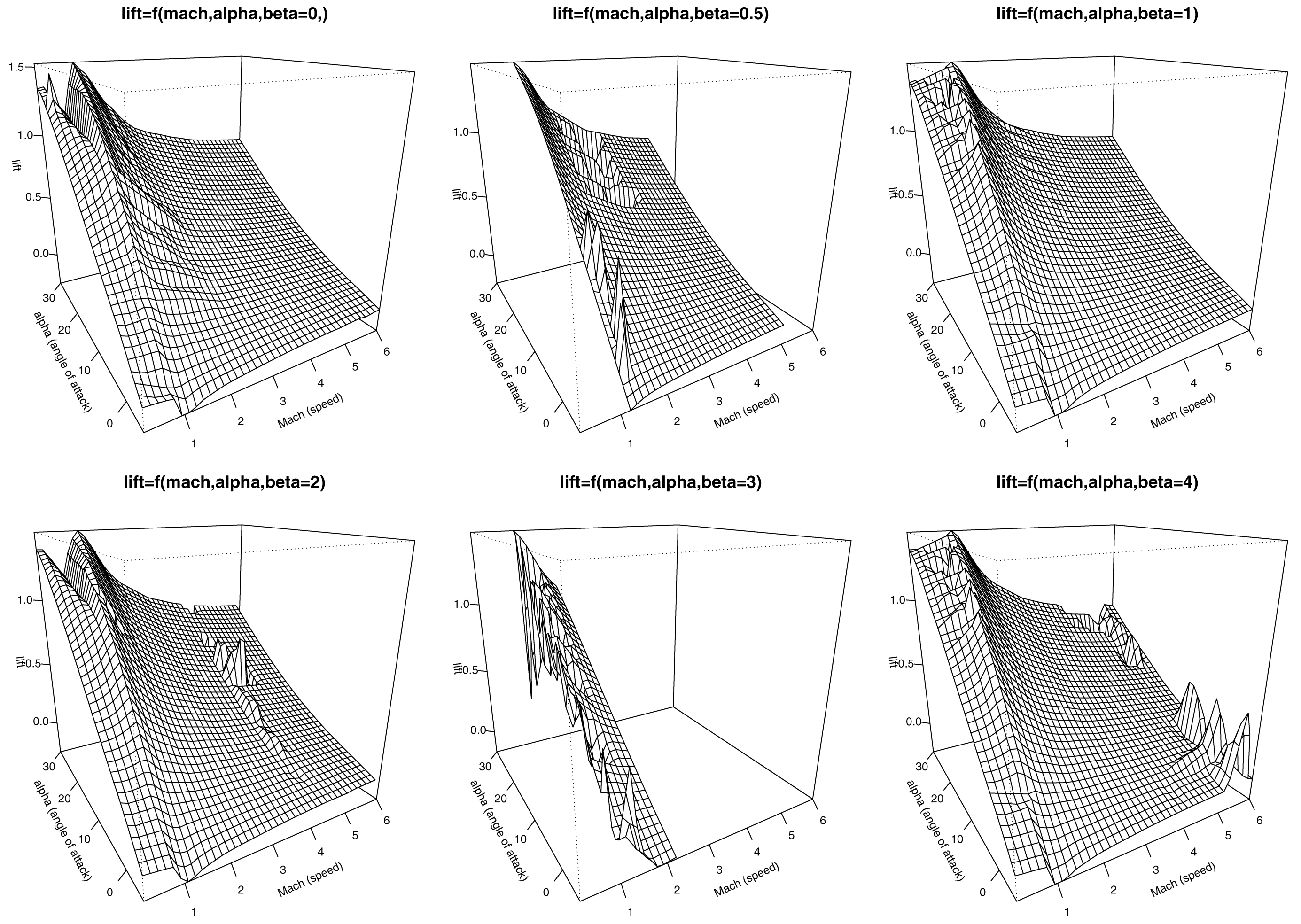


Figure 1. Interpolation of lift by speed and angle of attack for all sideslip levels. Note that for levels .5 and 3 (center), Mach ranges only in $(1, 5)$ and $(1.2, 2.2)$.

stationary GP would give uncertainty estimates that are fairly uniform across the space, due to the assumption of stationarity; therefore, we turn to a partitioning approach.

The engineers' understanding of the mean surface is important for several reasons. First, they may discover potential problems with the design, which could lead to structural changes in the design. Second, they need to determine the optimal flight conditions so they can plan the flight trajectories. Third, they need to be able to make contingency plans in the event a problem arises during a mission. If some of the stabilizing rockets fail and the vehicle must reenter at an unplanned angle or speed, they will need to be able to map out its new trajectory and adjust the process as necessary. The engineers are interested not just in the mean surface, but also in the uncertainty associated with prediction, because these uncertainties are not constant across the surface.

3. RELATED WORK

Our approach to nonparametric and semiparametric nonstationary modeling combines standard GPs and treed partitioning within the context of Bayesian hierarchical modeling and model averaging. We assume that the reader is familiar with the basic concepts of Bayesian inference through Markov chain Monte Carlo (MCMC) (e.g., Gilks, Richardson, and Spiegelhalter 1996). An introduction to GPs and treed partition modeling follows.

3.1 Stationary Gaussian Processes

A common specification of stochastic processes for spatial data, of which the stationary GP is a particular case, specifies that model outputs (responses), z, depend on multivariate inputs (explanatory variables), $\mathbf{x}$, as $z(\mathbf{x}) = \boldsymbol{\beta}^\top \mathbf{f}(\mathbf{x}) + w(\mathbf{x})$, where $\boldsymbol{\beta}$ are linear trend coefficients, $w(\mathbf{x})$ is a mean-0 random process with covariance $C(\mathbf{x}, \mathbf{x}') = \sigma^2 K(\mathbf{x}, \mathbf{x}')$, $\mathbf{K}$ is a correlation matrix, and $\boldsymbol{\beta}$ is independent of w in the prior. Low-order polynomials sometimes are used instead of the simple linear mean $\boldsymbol{\beta}^\top \mathbf{f}(\mathbf{x})$, or the mean process is specified generically, often as $\xi(\mathbf{x}, \boldsymbol{\beta})$ or $\xi(\mathbf{x})$.

Formally, a GP is a collection of random variables, $Z(\mathbf{x})$, indexed by $\mathbf{x}$, having a jointly Gaussian distribution for any finite subset of indexes (Stein 1999). It is specified by a mean function $\mu(\mathbf{x}) = E(Z(\mathbf{x}))$ and a correlation function $K(\mathbf{x}, \mathbf{x}') = \frac{1}{\sigma^2} E([Z(\mathbf{x}) - \mu(\mathbf{x})][Z(\mathbf{x}') - \mu(\mathbf{x}')]^\top)$. We assume that the correlation function can be written in the form

$$K(\mathbf{x}_j, \mathbf{x}_k | g) = K^*(\mathbf{x}_j, \mathbf{x}_k) + g\delta_{j,k}, \qquad (1)$$

where $\delta_{\cdot,\cdot}$ is the Kronecker delta function and K^* is a proper underlying parametric correlation function. The g term in (1) is referred to as the "nugget." The nugget always must be positive ($g > 0$), and it serves two purposes. First, it provides a mechanism for introducing measurement error into the stochastic process. This arises when considering a model of the form $Z(\mathbf{x}) = \xi(\mathbf{x}, \boldsymbol{\beta}) + w(\mathbf{x}) + \eta(\mathbf{x})$, where $w(\cdot)$ is a process with correlations governed by K^* and $\eta(\cdot)$ is simply Gaussian noise. Second, it helps prevent $\mathbf{K}$ from becoming numerically singular. Notational convenience and conceptual congruence motivates referral to $\mathbf{K}$ as a correlation matrix, even though the nugget term (g) forces $K(\mathbf{x}_i, \mathbf{x}_i) > 1$. There is an isomorphic model specification wherein $\mathbf{K}$ depicts proper correlations. Under both specifications, K^* indeed does define a valid correlation matrix $\mathbf{K}^*$.

The correlation functions, $K^*(\cdot,\cdot)$, typically are specified through a low-dimensional parametric structure, which also guarantees that they are symmetric and positive semidefinite. Here we focus on the power family, although our methods clearly can be extended to other families as well, such as the Matérn class (Matérn 1986). Further discussions of correlation structures have been given by Abrahamsen (1997) and Stein (1999). The power family of correlation functions includes the simple isotropic parameterization

$$K^*(\mathbf{x}_j, \mathbf{x}_k|d) = \exp\left\{-\frac{\|\mathbf{x}_j - \mathbf{x}_k\|^{p_0}}{d}\right\}, \qquad (2)$$

where $d > 0$ is a single range parameter and $p_0 \in (0, 2]$ determines the smoothness of the process. Thus the correlation of two points depends only on the Euclidean distance, $\|\mathbf{x}_j - \mathbf{x}_k\|$, between them. A straightforward enhancement to the isotropic power family is to use a separate range parameter, d_i, in each dimension ($i = 1, \ldots, m_X$). The resulting correlation function is still stationary but no longer isotropic,

$$K^*(\mathbf{x}_j, \mathbf{x}_k|\mathbf{d}) = \exp\left\{-\sum_{i=1}^{m_X} \frac{|x_{ij} - x_{ik}|^{p_0}}{d_i}\right\}. \qquad (3)$$

3.2 Treed Partitioning

Many spatial modeling problems require more flexibility than is offered by a stationary GP. One way to achieve a more flexible, nonstationary process is to use a partition model—a meta-model that divides up the input space and fits different base models to data independently in each region. Treed partitioning is one possible approach to this.

Treed partition models typically divide up the input space by making binary splits on the value of a single variable (e.g., $x_1 > .8$), so that partition boundaries are parallel to coordinate axes. Partitioning is recursive, so each new partition is a subpartition of a previous partition. For example, a first partition may divide the space in half by whether the first variable is above or below its midpoint. The second partition then divides only the space below (or above) the midpoint of the first variable, so that there are now three partitions (not four). Because variables may be revisited, using binary splits incurs no loss of generality, because multiple splits on the same variable will be equivalent to a nonbinary split. In each partition (leaf of the tree), an independent model is applied. A classification and regression tree (CART) (Breiman, Friedman, Olshen, and Stone 1984) is an example of a treed partition model. CART, which fits a constant surface in each leaf, has become popular because of its ease of use, clear interpretation, and ability to provide a good fit in many cases.

Applying the Bayesian approach to CART is straightforward (Chipman et al. 1998; Denison, Mallick, and Smith 1998), provided that a meaningful prior for the size of the tree can be specified. We follow Chipman et al. (1998), who specified the prior through a tree-generating process and enforced a minimum amount of data to infer the parameters in each partition. Starting with a null tree (i.e., all data in a single region), a leaf node $\eta \in \mathcal{T}$, representing a region of the input space, splits with probability $a(1 + q_\eta)^{-b}$, where q_η is the depth of $\eta \in \mathcal{T}$ and a and b are parameters chosen to give an appropriate size and spread to the distribution of trees. Further details are available in the works of Chipman et al. For our models, we have found that default values of $a = .5$ and $b = 2$ often work well in practice, although in any particular problem, prior knowledge may call for other values. The prior for the splitting process involves first choosing the splitting dimension, u, from a discrete uniform, and then choosing the split location, s, uniformly from a subset of the locations, $\mathbf{X}$, in the uth dimension. Integrating out dependence on the tree structure, $\mathcal{T}$, can be accomplished through reversible-jump (RJ) MCMC, as further described in Section 4.2.2.

Chipman et al. (2002) generalized Bayesian CART to create the Bayesian treed linear model (LM) by fitting hierarchical LMs at the leaves of the tree. In Section 4 we generalize further by proposing to fit stationary GPs in each leaf of the tree. This approach bears some similarity to that of Kim et al. (2005), who fit separate GPs in each element of a Voronoi tessellation. The treed GP approach is geared more toward problems with a smaller number of distinct partitions, leading to a simpler overall model. Voronoi tessellations allow intricate partitioning of the space but have the trade-off of added complexity and can produce a final model that is difficult to interpret. The tessellation approach also has the benefit of not being restricted to axis-aligned partitions, although in some cases a simple transformation, such as rotating the data, will suffice to allow axis-aligned partitions. A nice review of Bayesian partition modeling has been provided by Denison et al. (2002).

Other approaches to nonstationary modeling include those that use spatial deformations and process convolutions. The idea behind the spatial deformation approach is to map nonstationary inputs in the original, geographical space into a dispersion space in which the process is stationary. Sampson and Guttorp (1992) used thin-plate spline models and multidimensional scaling (MDS) to construct the mapping. Damian, Sampson, and Guttorp (2001) explored a similar methodology from a Bayesian perspective. Schmidt and O'Hagan (2003) also took the Bayesian approach but put a Gaussian process prior on the mapping. The process convolution approach (Higdon et al. 1999; Fuentes 2002; Paciorek 2003) proceeds by allowing the convolution kernels to vary smoothly in parameterization as an unknown function of their spatial location. A common theme of such nonstationary models is the introduction of meta-structure that ratchets up the flexibility of the model, which ratchets up the computational demands as well. This approach is in stark contrast to the treed approach, which introduces a structural mechanism—the tree, $\mathcal{T}$—that actually reduces the computational burden relative to the base model (e.g., a GP), because smaller correlation matrixes are inverted. A key difference is that these alternative approaches strictly enforce continuity of the process, which requires much more effort than the treed approach.

4. TREED GAUSSIAN PROCESS MODELS

Extending the partitioning ideas of Chipman et al. (1998, 2002) for simple Bayesian treed models, we fit stationary GP models with linear trends independently within each of R regions, $\{r_\nu\}_{\nu=1}^R$, depicted at the leaves of the tree $\mathcal{T}$, instead of constant (1998) or linear (2002) models. The tree is averaged out by integrating over possible trees using RJ–MCMC (Richardson and Green 1997), with the tree prior specified through a tree-generating process. Prediction is conditioned on the tree structure and is averaged over in the posterior to get a full accounting of uncertainty.

4.1 Hierarchical Model

A tree, $\mathcal{T}$, recursively partitions the input space into R nonoverlapping regions: $\{r_\nu\}_{\nu=1}^R$. Each region, r_ν, contains data, $D_\nu = \{\mathbf{X}_\nu, \mathbf{Z}_\nu\}$, consisting of n_ν observations. Let $m \equiv m_X + 1$ be the number of covariates in the design (input) matrix $\mathbf{X}$ plus an intercept. For each region r_ν, the hierarchical generative GP model is

$$\begin{aligned} \mathbf{Z}_\nu | \boldsymbol{\beta}_\nu, \sigma_\nu^2, \mathbf{K}_\nu &\sim \mathrm{N}_{n_\nu}(\mathbf{F}_\nu \boldsymbol{\beta}_\nu, \sigma_\nu^2 \mathbf{K}_\nu), \\ \boldsymbol{\beta}_0 &\sim \mathrm{N}_m(\boldsymbol{\mu}, \mathbf{B}), \\ \boldsymbol{\beta}_\nu | \sigma_\nu^2, \tau_\nu^2, \mathbf{W}, \boldsymbol{\beta}_0 &\sim \mathrm{N}_m(\boldsymbol{\beta}_0, \sigma_\nu^2 \tau_\nu^2 \mathbf{W}), \\ \tau_\nu^2 &\sim \mathrm{IG}(\alpha_\tau/2, q_\tau/2), \\ \sigma_\nu^2 &\sim \mathrm{IG}(\alpha_\sigma/2, q_\sigma/2), \\ \mathbf{W}^{-1} &\sim \mathrm{W}((\rho \mathbf{V})^{-1}, \rho), \end{aligned} \tag{4}$$

with $\mathbf{F}_\nu = (\mathbf{1}, \mathbf{X}_\nu)$, where $\mathbf{W}$ is an $m \times m$ matrix, and N, IG, and W are the (multivariate) normal, inverse-gamma, and Wishart distributions. Hyperparameters $\boldsymbol{\mu}, \mathbf{B}, \mathbf{V}, \rho, \alpha_\sigma, q_\sigma, \alpha_\tau$, and q_τ are treated as known. The model (4) specifies a multivariate normal likelihood with linear trend coefficients $\boldsymbol{\beta}_\nu$, variance σ_ν^2, and $n_\nu \times n_\nu$ correlation matrix $\mathbf{K}_\nu$. The coefficients $\boldsymbol{\beta}_\nu$ are believed to have come from a common unknown mean, $\boldsymbol{\beta}_0$, and region-specific variance, $\sigma_\nu^2 \tau_\nu^2$. Model (4) includes no explicit mechanism to ensure that the process near the boundary of two adjacent regions remains continuous across the partitions depicted by $\mathcal{T}$. But the model can capture smoothness through model averaging, as we discuss in Section 4.3. In our work with models for physical processes, we often encounter problems with phase transitions where the response surface is not smooth at the boundary between distinct physical regimes (e.g., subsonic versus supersonic flight of the rocket booster); thus we view the ability to fit a discontinuous surface as a feature of this model.

The GP correlation structure, $K_\nu(\mathbf{x}_j, \mathbf{x}_k) = K_\nu^*(\mathbf{x}_j, \mathbf{x}_k) + g_\nu \delta_{j,k}$, generating $\mathbf{K}_\nu$ for each partition r_ν takes K_ν^* to be from the isotropic power family (2) or separable power family (3), with fixed power p_0 but unknown (random) range and nugget parameters. But because most of the following discussion holds for K_ν^* generated by other families, as well as for unknown p_0, we refer to the correlation parameters indirectly through the resulting correlation matrix $\mathbf{K}$, or function $K(\cdot,\cdot)$; for example, $p(\mathbf{K}_\nu)$ can represent either $p(d_\nu, g_\nu)$ or $p(\mathbf{d}_\nu, g_\nu)$, and so on. Priors that encode a preference for a model with a nonstationary global covariance structure are chosen for parameters to K_ν^* and g_ν. In particular, we propose a mixture-of-gammas prior for d,

$$\begin{aligned} p(d,g) &= p(d) \times p(g) \\ &= p(g) \times \frac{1}{2}[\mathrm{G}(d|\alpha = 1, \beta = 20) \\ &\qquad + \mathrm{G}(d|\alpha = 10, \beta = 10)]. \end{aligned} \tag{5}$$

This prior gives roughly equal mass to small d representing a population of GP parameterizations for wavy surfaces and a separate population for surfaces that are quite smooth or approximately linear. We take the prior for g to be $\mathrm{Exp}(\lambda)$. Alternatively, we could encode the prior as $p(d,g) = p(d|g)p(g)$ and then use a reference prior (Berger, de Oliveira, and Sansó 2001) for $p(d|g)$. We prefer the more deliberate mixture specification both for its modeling implications and for its ability to interface well with limiting linear models (Gramacy and Lee 2008).

It also may be sensible to define the prior for $\{\mathbf{K}, \sigma^2, \tau^2\}_\nu$ hierarchically, depending on parameters $\boldsymbol{\gamma}$ (not indexed by ν), similar to how the population of $\boldsymbol{\beta}_\nu$ parameters is given a common prior in terms of $\mathbf{W}$ and $\boldsymbol{\beta}_0$ in (4).

4.2 Estimation

The data, $D_\nu = \{\mathbf{X}, \mathbf{Z}\}_\nu$, are used to update the GP parameters, $\boldsymbol{\theta}_\nu \equiv \{\boldsymbol{\beta}, \sigma^2, \mathbf{K}, \tau^2\}_\nu$, for $\nu = 1, \ldots, R$. Conditional on the tree $\mathcal{T}$, the full set of parameters is denoted as $\boldsymbol{\theta} = \boldsymbol{\theta}_0 \cup \bigcup_{\nu=1}^R \boldsymbol{\theta}_\nu$, where $\boldsymbol{\theta}_0 = \{\mathbf{W}, \boldsymbol{\beta}_0, \boldsymbol{\gamma}\}$ denotes upper-level parameters from the hierarchical prior that also are updated. Samples from the posterior distribution of $\boldsymbol{\theta}$ are gathered using MCMC by first conditioning on the hierarchical prior parameters $\boldsymbol{\theta}_0$ and drawing $\boldsymbol{\theta}_\nu | \boldsymbol{\theta}_0$ for $\nu_1, \ldots, \nu_R$, and then drawing $\boldsymbol{\theta}_0$ as $\boldsymbol{\theta}_0 | \bigcup_{\nu=1}^R \boldsymbol{\theta}_\nu$. Section 4.2.1 gives the details. All parameters can be sampled with Gibbs steps, except those that parameterize the covariance function $K(\cdot,\cdot)$, such as $\{d, g\}_\nu$, which require Metropolis–Hastings (MH) draws. Section 4.2.2 shows how RJ–MCMC is used to gather samples from the joint posterior of $(\boldsymbol{\theta}, \mathcal{T})$ by alternately drawing $\boldsymbol{\theta}|\mathcal{T}$ and $\mathcal{T}|\boldsymbol{\theta}$.

4.2.1 Gaussian Process Parameters Given a Tree ($\mathcal{T}$). Conditional conjugacy allows Gibbs sampling for most parameters. Full derivations of the following equations have been given by Gramacy (2005). The linear regression parameters $\boldsymbol{\beta}_\nu$ and prior mean $\boldsymbol{\beta}_0$ both have multivariate normal full conditionals, $\boldsymbol{\beta}_\nu|\mathrm{rest} \sim \mathrm{N}_m(\tilde{\boldsymbol{\beta}}_\nu, \sigma_\nu^2 \mathbf{V}_{\tilde{\beta}_\nu})$ and $\boldsymbol{\beta}_0|\mathrm{rest} \sim \mathrm{N}_m(\tilde{\boldsymbol{\beta}}_0, \boldsymbol{V}_{\tilde{\beta}_0})$, where

$$\begin{aligned} \mathbf{V}_{\tilde{\beta}_\nu} &= (\mathbf{F}_\nu^\top \mathbf{K}_\nu^{-1} \mathbf{F}_\nu + \mathbf{W}^{-1}/\tau_\nu^2)^{-1}, \\ \tilde{\boldsymbol{\beta}}_\nu &= \mathbf{V}_{\tilde{\beta}_\nu}(\mathbf{F}_\nu^\top \mathbf{K}_\nu^{-1} \mathbf{Z}_\nu + \mathbf{W}^{-1} \boldsymbol{\beta}_0/\tau_\nu^2), \\ \mathbf{V}_{\tilde{\beta}_0} &= \left(\mathbf{B}^{-1} + \mathbf{W}^{-1} \sum_{\nu=1}^R (\sigma_\nu \tau_\nu)^{-2}\right)^{-1}, \end{aligned} \tag{6}$$

$$\tilde{\boldsymbol\beta}_0 = \mathbf{V}_{\tilde\beta_0}\left(\mathbf{B}^{-1}\mu + \mathbf{W}^{-1}\sum_{\nu=1}^{R}\boldsymbol\beta_\nu(\sigma_\nu\tau_\nu)^{-2}\right).$$

The linear variance parameter τ^2 follows an inverse-gamma distribution,

$$\tau_\nu^2|\text{rest} \sim \text{IG}((\alpha_\tau + m)/2, (q_\tau + b_\nu)/2),$$

where

$$b_\nu = (\boldsymbol\beta_\nu - \boldsymbol\beta_0)^\top \mathbf{W}^{-1}(\boldsymbol\beta_\nu - \boldsymbol\beta_0)/\sigma_\nu^2.$$

The linear model covariance matrix $\mathbf{W}$ follows an inverse-Wishart distribution,

$$\mathbf{W}^{-1}|\text{rest} \sim \text{W}_m((\rho\mathbf{V} + \mathbf{V}_{\hat W})^{-1}, \rho + R),$$

where

$$\mathbf{V}_{\hat W} = \sum_{\nu=1}^{R}\frac{1}{(\sigma_\nu\tau_\nu)^2}(\boldsymbol\beta_\nu - \boldsymbol\beta_0)(\boldsymbol\beta_\nu - \boldsymbol\beta_0)^\top.$$

Analytically integrating out $\boldsymbol\beta_\nu$ and σ_ν^2 gives a marginal posterior for $\mathbf{K}_\nu$ and improves mixing of the Markov chain (Berger et al. 2001),

$$\begin{aligned} p(\mathbf{K}_\nu|\mathbf{Z}_\nu, \boldsymbol\beta_0, \mathbf{W}, \tau^2) &= \left(\frac{|\mathbf{V}_{\tilde\beta_\nu}|(2\pi)^{-n_\nu}}{|\mathbf{K}_\nu||\mathbf{W}|\tau^{2m}}\right)^{1/2} \\ &\quad\times \frac{(q_\sigma/2)^{\alpha_\sigma/2}\Gamma[(1/2)(\alpha_\sigma + n_\nu)]}{[(1/2)(q_\sigma + \psi_\nu)]^{(\alpha_\sigma + n_\nu)/2}\Gamma[\alpha_\sigma/2]}p(\mathbf{K}_\nu), \end{aligned} \quad (7)$$

where

$$\psi_\nu = \mathbf{Z}_\nu^\top\mathbf{K}_\nu^{-1}\mathbf{Z}_\nu + \boldsymbol\beta_0^\top\mathbf{W}^{-1}\boldsymbol\beta_0/\tau^2 - \tilde{\boldsymbol\beta}_\nu^\top\mathbf{V}_{\tilde\beta_\nu}^{-1}\tilde{\boldsymbol\beta}_\nu. \quad (8)$$

Equation (7) can be used to iteratively obtain draws for the parameters of $K_\nu(\cdot,\cdot)$ through MH or as part of the acceptance ratio for proposed modifications to $\mathcal{T}$ (see Sect. 4.2.2). Any hyperparameters to $K_\nu(\cdot,\cdot)$, such as parameters to priors for $\{d, g\}_\nu$ of the isotropic power family, also would require MH draws. The conditional distribution of σ_ν^2 with $\boldsymbol\beta_\nu$ integrated out allows Gibbs sampling,

$$\sigma_\nu^2|\mathbf{Z}_\nu, d_\nu, g, \boldsymbol\beta_0, \mathbf{W} \sim \text{IG}\big((\alpha_\sigma + n_\nu)/2, (q_\sigma + \psi_\nu)/2\big). \quad (9)$$

4.2.2 Tree ($\mathcal{T}$). Integrating out dependence on the tree structure ($\mathcal{T}$) is accomplished by RJ–MCMC. We augment the tree operations of Chipman et al. (1998)—grow, prune, change, swap—with a rotate operation. A *change* operation proposes moving an existing split-point $\{u, s\}$ to either the next greater or lesser value of s (s_+ or s_-) along the uth column of $\mathbf{X}$. This is done by sampling s' uniformly from the set $\{u_\nu, s_\nu\}_{\nu=1}^{\lceil R/2\rceil} \times \{+, -\}$, which causes the MH acceptance ratio for *change* to reduce to a simple likelihood ratio, because the parameters $\boldsymbol\theta_r$ in regions r below the split point $\{u, s'\}$ are held fixed.

A *swap* operation proposes changing the order in which two adjacent parent–child (internal) nodes split up the inputs. An internal parent–child node pair is picked at random from the tree, and their splitting rules are swapped. But swaps on parent–child internal nodes that split on the same variable cause problems, because a child region below both parents becomes empty after the operation. If instead a *rotate* operation from binary search trees (BSTs) is performed, then the proposal almost always will accept. Rotations are a way of adjusting the configuration and height of a BST without violating the BST property, as used by, for example, *red–black trees* (Cormen, Leiserson, and Rivest 1990).

In the context of a Bayesian MCMC tree proposal, rotations encourage better mixing of the Markov chain by providing a more dynamic set of candidate nodes for pruning, thereby helping escape local minima in the marginal posterior of $\mathcal{T}$. Figure 2 shows an example of a successful right-rotation in which a swap would produce an empty node (at the current location of $\mathcal{T}_2$). Because the partitions at the leaves remain unchanged, the likelihood ratio of a proposed rotation is always 1. The only "active" part of the MH acceptance ratio is the prior on $\mathcal{T}$, which prefers trees of minimal depth. Nonetheless, calculating the acceptance ratio for a rotation is nontrivial because

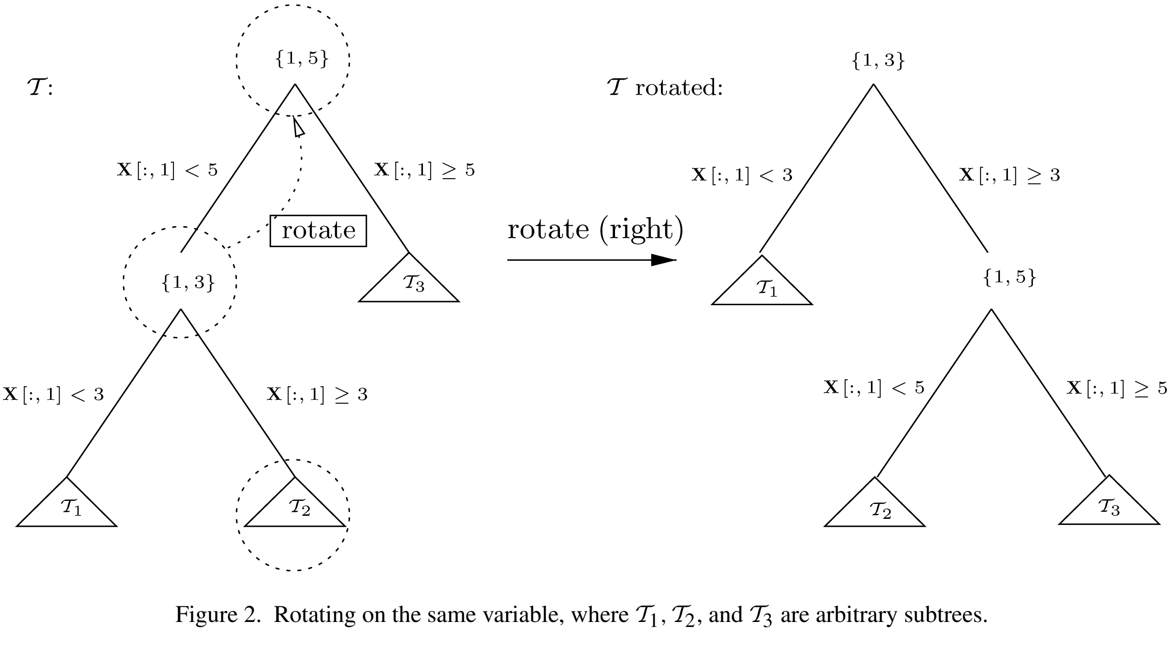


Figure 2. Rotating on the same variable, where $\mathcal{T}_1$, $\mathcal{T}_2$, and $\mathcal{T}_3$ are arbitrary subtrees.

the depth of two of its subtrees change. Figure 2 shows a right-rotation, where nodes in $\mathcal{T}_1$ decrease in depth while nodes in $\mathcal{T}_3$ increase in depth. The opposite is true for left-rotation. If $I = \{I_i, I_\ell\}$ is the set of nodes (internals and leaves) of $\mathcal{T}_1$ and $\mathcal{T}_3$, before rotation, which increase in depth after rotation, and $D = \{D_i, D_\ell\}$ are those that decrease in depth, then the MH acceptance ratio for a right-rotation is

$$\begin{aligned}\frac{p(\mathcal{T}^*)}{p(\mathcal{T})} &= \frac{p(\mathcal{T}_1^*)p(\mathcal{T}_3^*)}{p(\mathcal{T}_1)p(\mathcal{T}_3)}\\ &= \frac{\prod_{\eta\in I_i} a(2+q_\eta)^{-b}\prod_{\eta\in I_\ell}[1-a(2+q_\eta)^{-b}]}{\prod_{\eta\in I_i} a(1+q_\eta)^{-b}\prod_{\eta\in I_\ell}[1-a(1+q_\eta)^{-b}]}\\ &\quad\times\frac{\prod_{\eta\in D_i} aq_\eta^{-b}\prod_{\eta\in D_\ell}[1-aq_\eta^{-b}]}{\prod_{\eta\in D_i} a(1+q_\eta)^{-b}\prod_{\eta\in D_\ell}[1-a(1+q_\eta)^{-b}]}.\end{aligned}$$

The MH acceptance ratio for a left-rotation is analogous.

Grow and *prune* operations are complex because they add or remove partitions, changing the dimension of the parameter space. The first step in either operation is to uniformly select a leaf node (for *grow*) or the parent of a pair of leaf nodes (for *prune*). When a new region, r, is added, new parameters, $\{K(\cdot,\cdot), \tau^2\}_r$, must be proposed, and when a region is taken away, the parameters must be either absorbed by the parent region or discarded. When evaluating the MH acceptance ratio, the linear model parameters $\{\boldsymbol{\beta}, \sigma^2\}_r$ are integrated out (7). One of the newly grown children is uniformly chosen to receive the correlation function, $K(\cdot,\cdot)$, of its parent, essentially inheriting a block from its parent's correlation matrix. To ensure that the resulting Markov chain is ergodic and reversible, the other new sibling draws its $K(\cdot,\cdot)$ from the prior, thus giving a unity Jacobian term in the RJ–MCMC. Note that *grow* operations are the only ones in which priors are used as proposals; random-walk proposals are used elsewhere (see Sect. 4.4).

Symmetrically, *prune* operations randomly select parameters from $K(\cdot,\cdot)$ for the consolidated node from one of the children being absorbed. After accepting a *grow* or *prune*, σ_r^2 can be drawn from its marginal posterior, with $\boldsymbol{\beta}_r$ integrated out [eq. (9)], followed by draws for $\boldsymbol{\beta}_r$ and the rest of the parameters in the rth region.

Let $\{\mathbf{X}, \mathbf{Z}\}$ be the data at the new parent node η at depth q_η, and let $\{\mathbf{X}_1, \mathbf{Z}_1\}$ and $\{\mathbf{X}_2, \mathbf{Z}_2\}$ be the partitioned child data at depth $q_\eta + 1$ created by a new split $\{u, s\}$. Moreover, let $\mathcal{P}$ be the set of prunable nodes of $\mathcal{T}$ and let $\mathcal{G}$ be the set of growable nodes. If $\mathcal{P}'$ are the prunable nodes in $\mathcal{T}'$—after the (successful) *grow* at η—and the parent of η is prunable in $\mathcal{T}$, then $|\mathcal{P}'| = |\mathcal{P}|$; otherwise, $|\mathcal{P}'| = |\mathcal{P}| + 1$. The MH ratio for *grow* is

$$\begin{aligned}&\frac{|\mathcal{G}|}{|\mathcal{P}'|}\frac{a(1+q_\eta)^{-b}(1-a(2+q_\eta)^{-b})^2}{1-a(1+q_\eta)^{-b}}\\ &\qquad\times\frac{p(\mathbf{K}_1|\mathbf{Z}_1,\boldsymbol{\beta}_0,\tau_1^2,\mathbf{W})p(\mathbf{K}_2|\mathbf{Z}_2,\boldsymbol{\beta}_0,\tau_2^2,\mathbf{W})}{p(\mathbf{K}|\mathbf{Z},\boldsymbol{\beta}_0,\tau^2,\mathbf{W})q(\mathbf{K}_2)} \qquad (10)\end{aligned}$$

assuming that $\mathbf{K}_1$ was randomly chosen to receive the parameterization of its parent, $\mathbf{K}$, and that the new parameters for $\mathbf{K}_2$ are proposed according to q. The *prune* operation is analogous. Note that for the posteriors $p(\mathbf{K}|\mathbf{Z},\boldsymbol{\beta}_0,\tau^2,\mathbf{W})$, $p(\mathbf{K}_1|\mathbf{Z}_1,\boldsymbol{\beta}_0,\tau_1^2,\mathbf{W})$, and $p(\mathbf{K}_2|\mathbf{Z}_2,\boldsymbol{\beta}_0,\tau_2^2,\mathbf{W})$, the "constant" terms in (7) are required because they do not occur the same number of times in the numerator and denominator.

4.3 Treed GP Prediction

Prediction under the foregoing GP model is straightforward (Hjort and Omre 1994). Conditional on the covariance structure, the predicted value of $z(\mathbf{x} \in r_\nu)$ is normally distributed with mean and variance

$$\begin{aligned}\hat{z}(\mathbf{x}) &= E(\mathbf{Z}(\mathbf{x})|\text{data}, \mathbf{x} \in D_\nu)\\ &= \mathbf{f}^\top(\mathbf{x})\tilde{\boldsymbol{\beta}}_\nu + \mathbf{k}_\nu(\mathbf{x})^\top\mathbf{K}_\nu^{-1}(\mathbf{Z}_\nu - \mathbf{F}_\nu\tilde{\boldsymbol{\beta}}_\nu) \qquad (11)\end{aligned}$$

and

$$\begin{aligned}\hat{\sigma}(\mathbf{x})^2 &= \text{var}(\mathbf{z}(\mathbf{x})|\text{data}, \mathbf{x} \in D_\nu)\\ &= \sigma_\nu^2[\kappa(\mathbf{x},\mathbf{x}) - \mathbf{q}_\nu^\top(\mathbf{x})\mathbf{C}_\nu^{-1}\mathbf{q}_\nu(\mathbf{x})], \qquad (12)\end{aligned}$$

where

$$\begin{aligned}\mathbf{C}_\nu^{-1} &= (\mathbf{K}_\nu + \tau_\nu^2\mathbf{F}_\nu\mathbf{W}\mathbf{F}_\nu^\top)^{-1},\\ \mathbf{q}_\nu(\mathbf{x}) &= \mathbf{k}_\nu(\mathbf{x}) + \tau_\nu^2\mathbf{F}_\nu\mathbf{W}_\nu\mathbf{f}(\mathbf{x}), \qquad (13)\\ \kappa(\mathbf{x},\mathbf{y}) &= K_\nu(\mathbf{x},\mathbf{y}) + \tau_\nu^2\mathbf{f}^\top(\mathbf{x})\mathbf{W}\mathbf{f}(\mathbf{y}),\end{aligned}$$

with $\mathbf{f}^\top(\mathbf{x}) = (1, \mathbf{x}^\top)$, and $\mathbf{k}_\nu(\mathbf{x})$ is a n_ν vector with $\mathbf{k}_{\nu,j}(\mathbf{x}) = K_\nu(\mathbf{x}, \mathbf{x}_j)$ for all $\mathbf{x}_j \in \mathbf{X}_\nu$.

Conditional on a particular tree, $\mathcal{T}$, the posterior predictive surface described in (11)–(12) is discontinuous across the partition boundaries of $\mathcal{T}$. But in the aggregate of samples collected from the joint posterior distribution of $(\mathcal{T}, \boldsymbol{\theta})$, the mean tends to smooth out near likely partition boundaries as the tree operations *grow, prune, change*, and *swap* integrate over trees and GPs with larger posterior probability. Uncertainty in the posterior for $\mathcal{T}$ translates into higher posterior predictive uncertainty near region boundaries. When the data actually indicate a nonsmooth process, the treed GP retains the flexibility necessary to model discontinuities. When the data are consistent with a continuous process, as in the motorcycle data example in Section 4.5, the treed GP fits are almost indistinguishable from continuous.

4.4 Implementation

The treed GP model is coded in a mixture of `C` and `C++`, using `C++` for the tree structure and `C` for the GP at each leaf of $\mathcal{T}$. The `C` code either can interface with standard platform-specific `Fortran BLAS/Lapack` libraries for the necessary linear algebra routines or link to those automatically configured for fast execution on various platforms through the `ATLAS` library (Whaley and Petitet 2004). To improve usability, the routines have been wrapped up in an intuitive `R` interface and are available on CRAN (R Development Core Team 2004) at *http:// www.cran.r-project.org/web/packages/tgp/index.html* as a package called `tgp`.

It is useful to first translate and rescale the input data set, $\mathbf{X}$, so that it lies in an $\Re^{m_X}$-dimensional unit cube. This makes it easier to construct prior distributions for the width parameters to the correlation function $K(\cdot,\cdot)$. Conditioning on $\mathcal{T}$, proposals for all parameters that require MH sampling are taken from a uniform "sliding window" centered around the location of the last accepted setting; for example, a proposed new nugget parameter, g_ν, to the correlation function $K(\cdot,\cdot)$ in region r_ν would go as $g_\nu^* \sim \text{Unif}(3g_\nu/4, 4g_\nu/3)$. Calculating the forward and backward proposal probabilities for the MH acceptance ratio is straightforward.

After conditioning on $(\mathcal{T}, \boldsymbol{\theta})$, prediction can be parallelized by using a producer–consumer model. This allows the use of `PThreads` to take advantage of multiple processors and get speed-ups of at least a factor of two, which is helpful as multiprocessor machines become commonplace. Parallel sampling of the posterior of $\boldsymbol{\theta}|\mathcal{T}$ for each of the $\{\theta_\nu\}_{\nu=1}^R$ also is possible.

4.5 Illustration

In this section we illustrate the treed GP model on the motorcycle accident data set of Silverman (1985), a classic data set used in recent literature (e.g., Rasmussen and Ghahramani 2002) to demonstrate the success of nonstationary models. The data set comprises measurements of the acceleration of the head of a motorcycle rider, which we attempt to predict as a function of time in the first moments after an impact. In addition to suggesting a model with a nonstationary covariance structure, there is input-dependent noise (i.e., heteroscedasticity). To keep things simple in this illustration, the isotropic power family (2) correlation function ($p_0 = 2$) is chosen for $K^*(\cdot, \cdot|d)$ with range parameter d, combined with nugget g to form $K(\cdot, \cdot|d, g)$.

Figure 3 shows the data and the fits given by both a stationary GP and the treed GP model, along with 90% credible intervals. For the treed GP, vertical lines illustrate a typical treed partition, $\mathcal{T}$. Note that the stationary GP is unable to capture the heteroscedasticity, and that the large variability in the central region drives both ends to be more wiggly. (In particular, the transition from the flat left initial region requires an upward curve before descending.) In contrast, the treed GP clearly reflects the differing levels of uncertainty and also allows a flatter fit to the initial segment and a smoother fit to the final segment. A total of 20,000 MCMC rounds yielded an average of 3.11 partitions in $\mathcal{T}$.

These results differ from those of Rasmussen and Ghahramani (2002). In particular, the error bars that they reported for the leftmost region seem too large relative to the other regions. They used what they call an "infinite mixture of GP experts," which is a Dirichlet process mixture of GPs. They reported that the posterior distribution uses between 3 and 10 experts to fit this data set, with even 10–15 experts still having considerable posterior mass, although there are "roughly" 3 regions. Contrast this with the treed GP model, which almost always partitions into three regions (occasionally four and rarely two). On speed grounds, the treed GP also is a winner. Running the mixture of GP experts model using a total of 11,000 MCMC rounds (discarding the first 1,000 rounds) took roughly 1 hour on a 1-GHz Pentium. Allowing treed GP to use 25,000 MCMC rounds (discarding the first 5,000 rounds) took about 3 minutes on a 1.8-GHz Athlon.

We note that the mean fitted function in Figure 3(b) is essentially that of a continuous function. Figure 4 shows examples of the fits from individual MCMC iterations that are eventually averaged. Whereas the individual partition models are typically discontinuous, Figure 3 clearly shows that the mean fitted function is well behaved.

4.6 Limiting Linear Models

In some cases a GP may not be needed within a partition, and a much simpler model, such as a linear model, may suffice. In particular, because of the linear mean function in our implementation of the GP, the standard linear model can be viewed as a limiting case. The linear model is then more parsimonious, as well as much more computationally efficient. Using a model-switching prior allows practical implementation. More details have been given by Gramacy and Lee (2008). The value of such an approach can be seen from the fit shown in Figure 3(b). The leftmost partition looks quite flat, and so could be fit just as well with a line as with a GP. The center partition clearly requires a GP fit. The rightmost partition looks mostly linear and would give a posterior that is a mix of a GP and linear model. Indeed, in the examples of the individual fits shown in Figure 4, the leftmost section is nearly always flat, the rightmost section is often but not always flat, and the center section is typically curved, but even there it can be essentially piecewise linear. (The range parameter d is estimated to be large, giving a nearly linear fit.) Replacing the full GP with a linear model in a partition greatly reduces the computational resources required to update the model in that partition. Treed and nontreed Gaussian process with jumps to the limiting linear model are implemented in the `tgp` package in CRAN, and we take advantage of the full formulation in our analyses herein.

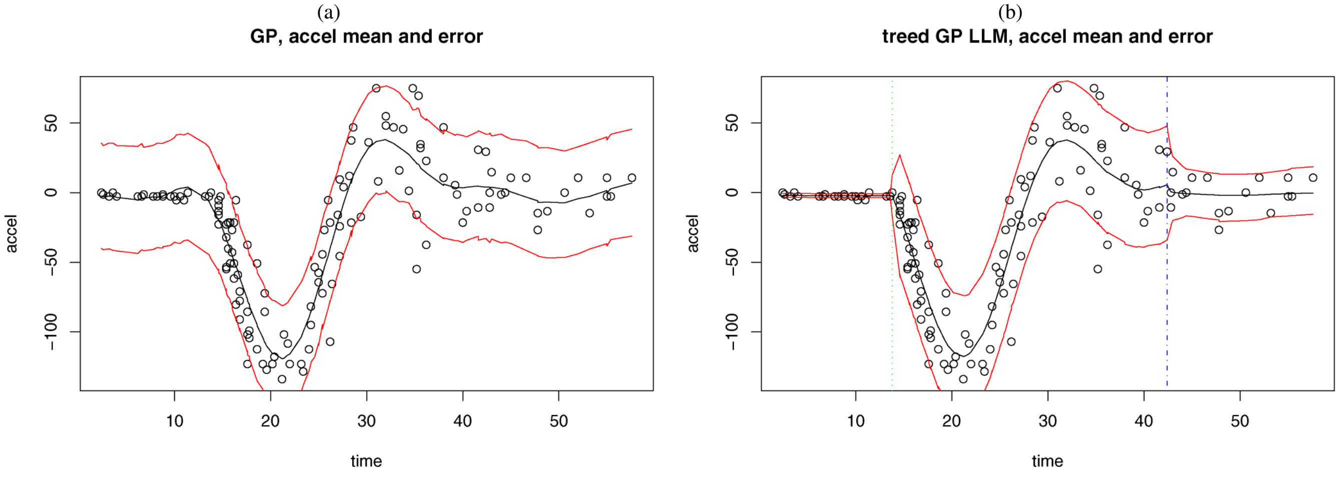


Figure 3. Motorcycle data set, fit by a stationary process (a) and by our nonstationary model (b).

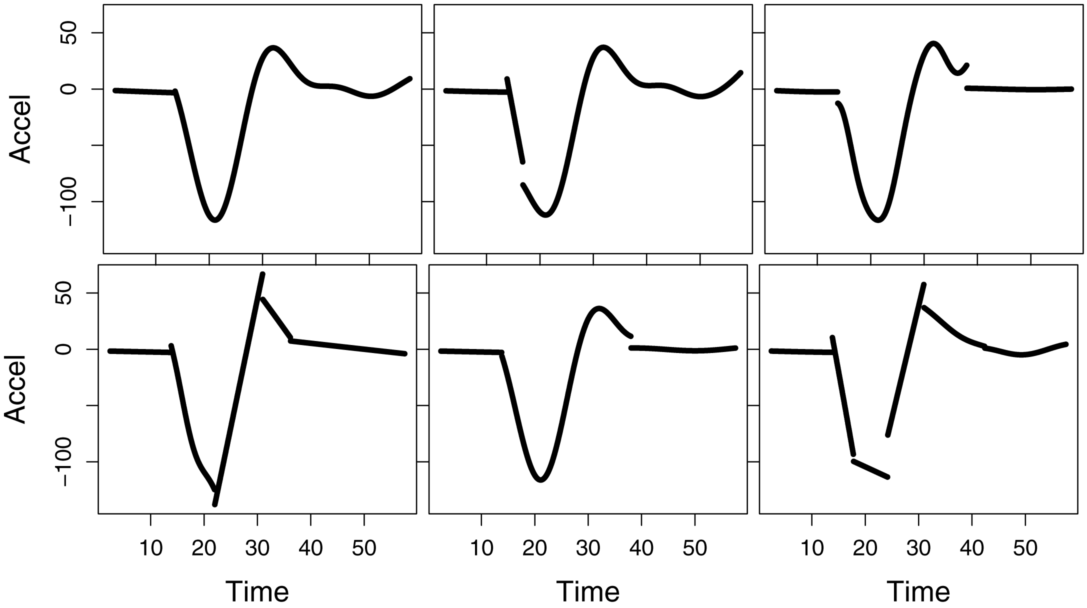


Figure 4. Fits of the motorcycle accident data from individual MCMC iterations.

5. ROCKET BOOSTER MODEL RESULTS

We fit our treed GP model to the rocket booster data using 10 independent RJ–MCMC chains with 15,000 MCMC rounds each. The first 5,000 were discarded as burn-in, and every tenth one thereafter was treated as a sample from the posterior distribution $\pi(\mathcal{T}, \boldsymbol{\theta}|Z)$. In total, 10,000 samples were saved. This took about 60 hours on a single 3.2-GHz Xeon processor. On the same machine, using the same (tuned) linear algebra libraries, inverting a single $3{,}041 \times 3{,}041$ matrix took about 17 seconds, so obtaining the same number of samples from a stationary GP would have taken a minimum of 708 hours. This is a gross underestimate, because it assumes that only one inverse is needed per MCMC round. Moreover, it does not count any of the $O(n^2)$ operations, such as determinants of $\mathbf{K}$ (assuming a factorized $\mathbf{K}$ was saved in computing $\mathbf{K}^{-1}$) or multiplications like $\mathbf{Z}\mathbf{K}^{-1}\mathbf{Z}$ in (8), nor does it factor in the time needed to sample from the posterior predictive distribution.

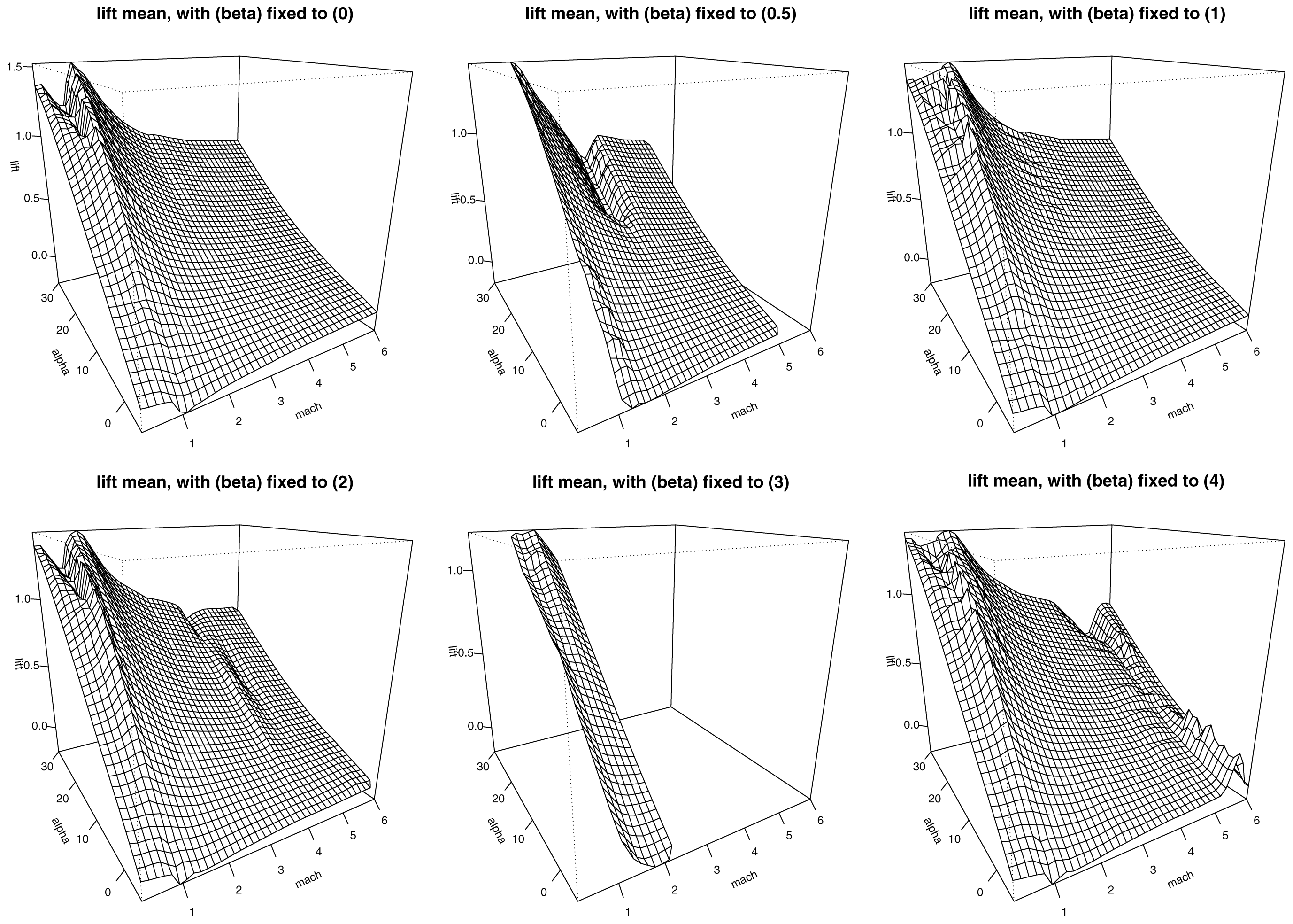
Figures 5 and 6 summarize the posterior predictive distribution for the lift response for each of the six levels of sideslip angle. Figure 5 contains plots of the fitted mean lift surface by speed and angle of attack, and Figure 6 plots a measure of the estimated predictive uncertainty given by the difference in 95% and 5% quantiles of samples from the posterior predictive distribution. The treed GP works well here. Most of the space is nicely smooth, with the sharp transition at Mach 1 also well modeled. Most of the potential false convergences have been smoothed out. But the estimated variability reflects increased variability where the function is changing rapidly (e.g., near Mach 1, particularly for higher sideslip levels), especially where there are issues of possible false numerical convergence. Note that the uncertainty is not that high near Mach 1 at sideslip level 0 because of the large number of samples taken in that region. We also note that the increased uncertainty seen in the top rows around Mach 3 and higher angles of attack is due to the noisy depression area in the data for sideslip level of .5. Figure 7 shows the MAP treed partitions, $\hat{\mathcal{T}}$, found during MCMC for the slice of sideslip level zero. Note the aggressive partitioning near Mach 1 due to the regime shift between subsonic and supersonic speeds. Extra partitioning at low speeds and large angles of attack address the singularity outlined in Figure 1 and near Mach 3 due to the numerical instabilities at sideslip level .5.

Figure 8 shows the mean fit and a 90% credible interval for a slice of predicting lift, with Mach on the x-axis and considering only an angle of attack equal to 25 and a slideslip angle equal to 0. (This is one slice from the upper left plot in Fig. 5; although this plot is from fitting the whole data set, we plot only one slice for better visualization.) The key item to note is that the fit is essentially continuous. The plot comprises only points at fitted values; no interpolation or lines have been used. In contrast, Figure 9 shows examples of the treed GP fits from individual MCMC iterations, which often have clear discontinuities from the partitioning structure. Thus, as is typical, our mean fitted values are quite smooth because they are an average, even though the individual components of the mean may not be continuous.

To measure goodness of fit, we typically rely on qualitative visual barometers. For example, we use traces to assess mixing in the Markov chain and inspect posterior predictive slices and projections, as described earlier. For a more quantitative assessment, we follow the suggestion of Gelfand (1995) and use 10-fold cross-validation. Posterior predictive quantiles are obtained for the input locations held out of each fold, and the proportion of held out responses that fall within the 90% predictive interval is recorded. For the LGBB data, we found a proportion of .96 using the treed GP LLM model. Thus our model fits well, and if anything, our predictive intervals are slightly wider than necessary, so we appear to be fully accounting for uncertainty.

6. CONCLUSION

We developed the treed Gaussian process model for the rocket booster computer experiment, but it also has a wide



lift mean, with (beta) fixed to (0)

lift mean, with (beta) fixed to (0.5)

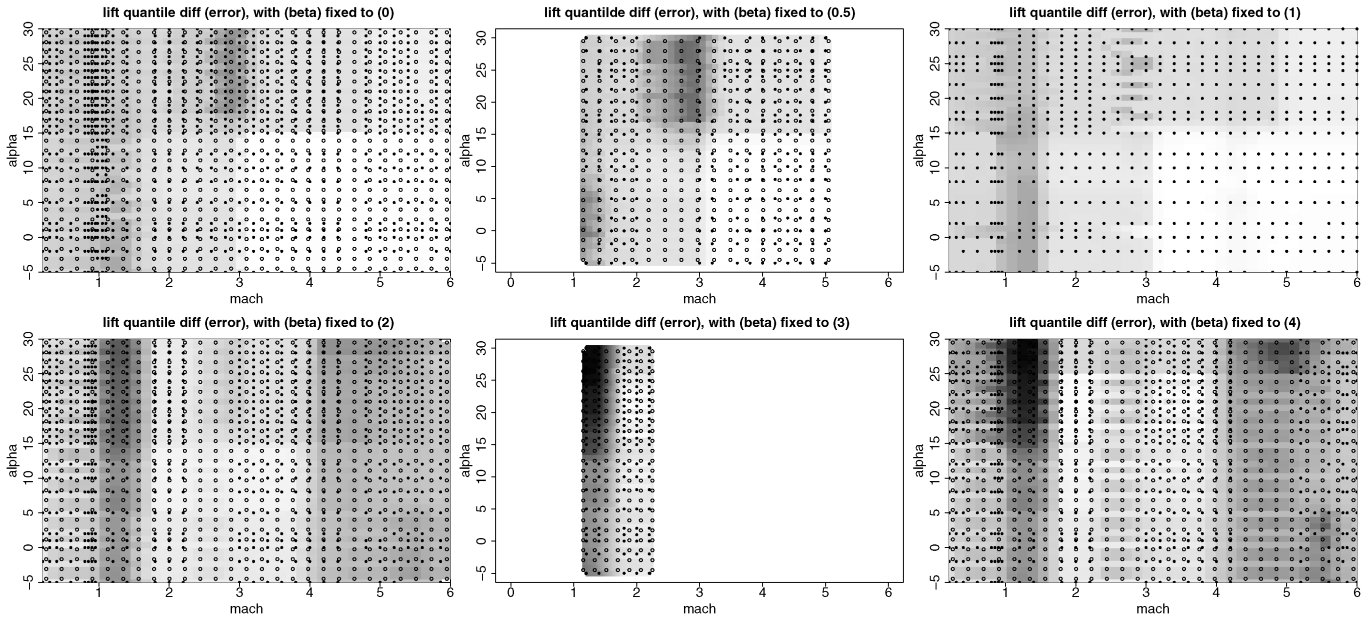
lift mean, with (beta) fixed to (1)

lift mean, with (beta) fixed to (2)

lift mean, with (beta) fixed to (3)

lift mean, with (beta) fixed to (4)

Figure 5. Posterior predictive mean surfaces of lift for all sideslip angles. Note that for levels .5 and 3 (center), Mach ranges only in (1, 5) and (1.2, 2.2).



lift quantile diff (error), with (beta) fixed to (0)

lift quantilde diff (error), with (beta) fixed to (0.5)

lift quantile diff (error), with (beta) fixed to (1)

lift quantile diff (error), with (beta) fixed to (2)

lift quantilde diff (error), with (beta) fixed to (3)

lift quantile diff (error), with (beta) fixed to (4)

Figure 6. Posterior predictive variance surfaces of lift for all six sideslip angles. Dots show the locations of experimental runs; darker shades are higher values.

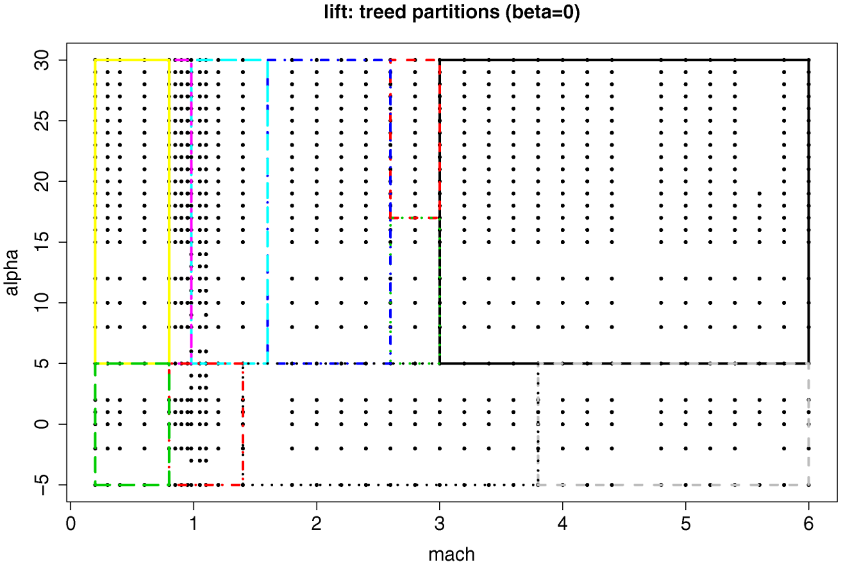


Figure 7. MAP treed partitions $\hat{\mathcal{T}}$ for the lift response at sideslip level 0.

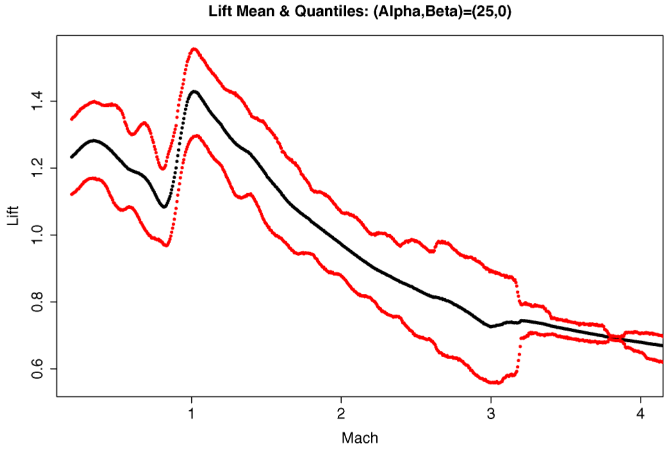


Figure 8. A slice of the mean fit with error bars as a function of Mach with alpha fixed to 25 and beta fixed to 0.

range of uses as a simple and efficient method for nonstationary modeling. In our fully Bayesian treatment of the treed GP model, the hierarchical parameterization of the correlation function $K(\cdot,\cdot)$ is treated as a modular component, easily replaced by a different family of correlations. The limiting linear model parameterization of the GP is both useful and accessible in terms of Bayesian posterior estimation and prediction, resulting in a uniquely nonstationary, semiparametric, tractable, and highly accurate model that contains the Bayesian treed linear model as a special case.

We believe that an important contribution of the treed GP will be in the domain of sequential design of computer experiments (Santner et al. 2003; Gramacy, Lee, and Macready 2004). Empirical evidence suggests that many computer experiments contain much linearity, as we have seen with large regions of the space for the rocket booster simulator. The Bayesian treed GP provides a full posterior predictive distribution (particularly

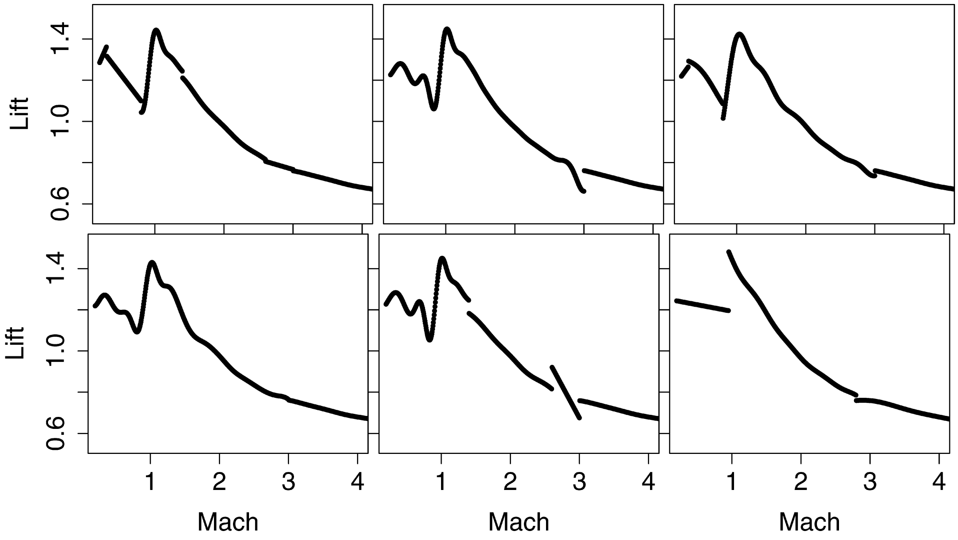


Figure 9. Slices of the posterior predictive mean from individual MCMC iterations for the LGBB data with alpha fixed to 25 and beta fixed to 0.

a nonstationary, and thus region-specific, estimate of predictive variance) that can be used in active learning in the input domain. Exploitation of these characteristics should lead to an efficient framework for the adaptive exploration of computer experiment parameter spaces.

[Received December 2006. Revised October 2007.]

REFERENCES

Abrahamsen, P. (1997), "A Review of Gaussian Random Fields and Correlation Functions," Technical Report 917, Norwegian Computing Center.

Berger, J. O., de Oliveira, V., and Sansó, B. (2001), "Objective Bayesian Analysis of Spatially Correlated Data," *Journal of the American Statistical Association*, 96, 1361–1374.

Breiman, L., Friedman, J. H., Olshen, R., and Stone, C. (1984), *Classification and Regression Trees*, Belmont, CA: Wadsworth.

Chipman, H., George, E., and McCulloch, R. (1998), "Bayesian CART Model Search" (with discussion), *Journal of the American Statistical Association*, 93, 935–960.

——— (2002), "Bayesian Treed Models," *Machine Learning*, 48, 303–324.

Cormen, T. H., Leiserson, C. E., and Rivest, R. L. (1990), *Introduction to Algorithms*, Cambridge, MA/New York: MIT Press/McGraw-Hill.

Damian, D., Sampson, P. D., and Guttorp, P. (2001), "Bayesian Estimation of Semiparametric Nonstationary Spatial Covariance Structure," *Environmetrics*, 12, 161–178.

Denison, D., Adams, N., Holmes, C., and Hand, D. (2002), "Bayesian Partition Modelling," *Computational Statistics and Data Analysis*, 38, 475–485.

Denison, D., Mallick, B., and Smith, A. (1998), "A Bayesian CART Algorithm," *Biometrika*, 85, 363–377.

Fuentes, M. (2002), "Spectral Methods for Nonstationary Spatial Processes," *Biometrika*, 89, 197–210.

Gelfand, A. (1995), "Model Determination Using Sampling-Based Methods," in *Markov Chain Monte Carlo in Practice*, eds. S. R. W. Gilks and D. Spiegelhalter, London: Chapman & Hall, pp. 145–161.

Gilks, W., Richardson, S., and Spiegelhalter, D. (1996), *Markov Chain Monte Carlo in Practice*, London: Chapman & Hall.

Gramacy, R. B. (2005), "Bayesian Treed Gaussian Process Models," unpublished doctoral thesis, University of California Santa Cruz, Dept. of Applied Mathematics and Statistics.

Gramacy, R. B., and Lee, H. K. H. (2008), "Gaussian Processes and Limiting Linear Models," *Computational Statistics & Data Analysis*, in press, available at *http://www.sciencedirect.com/science/article/B6V8V-4SW143G-1/2/5a5f2056b983c22d48a17375cbfa3bb4*.

Gramacy, R. B., Lee, H. K. H., and Macready, W. (2004), "Parameter Space Exploration With Gaussian Process Trees," in *ICML*, Omnipress & ACM Digital Library, pp. 353–360.

Higdon, D., Swall, J., and Kern, J. (1999), "Non-Stationary Spatial Modeling," in *Bayesian Statistics 6*, eds. J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, Oxford, U.K.: Oxford University Press, pp. 761–768.

Hjort, N. L., and Omre, H. (1994), "Topics in Spatial Statistics," *Scandinavian Journal of Statistics*, 21, 289–357.

Kennedy, M., and O'Hagan, A. (2001), "Bayesian Calibration of Computer Models" (with discussion), *Journal of the Royal Statistical Society*, Ser. B, 63, 425–464.

Kim, H.-M., Mallick, B. K., and Holmes, C. C. (2005), "Analyzing Nonstationary Spatial Data Using Piecewise Gaussian Processes," *Journal of the American Statistical Association*, 100, 653–668.

Matérn, B. (1986), *Spatial Variation* (2nd ed.), New York: Springer-Verlag.

Paciorek, C. (2003), "Nonstationary Gaussian Processes for Regression and Spatial Modelling," unpublished doctoral thesis, Carnegie Mellon University, Dept. of Statistics.

R Development Core Team (2004), *R: A Language and Environment for Statistical Computing*, Vienna, Austria: R Foundation for Statistical Computing.

Rasmussen, C., and Ghahramani, Z. (2002), "Infinite Mixtures of Gaussian Process Experts," in *Advances in Neural Information Processing Systems*, Vol. 14, eds. T. G. Dietterich, S. Becker, and Z. Ghahramani, Cambridge, MA: MIT Press, pp. 881–888.

Richardson, S., and Green, P. J. (1997), "On Bayesian Analysis of Mixtures With an Unknown Number of Components," *Journal of the Royal Statistical Society*, Ser. B, 59, 731–758.

Rogers, S. E., Aftosmis, M. J., Pandya, S. A., N. M. Chaderjian, E. T. T., and Ahmad, J. U. (2003), "Automated CFD Parameter Studies on Distributed Parallel Computers," AIAA Paper 2003-4229, presented at the 16th AIAA Computational Fluid Dynamics Conference.

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989), "Design and Analysis of Computer Experiments," *Statistical Science*, 4, 409–435.

Sampson, P. D., and Guttorp, P. (1992), "Nonparametric Estimation of Nonstationary Spatial Covariance Structure," *Journal of the American Statistical Association*, 87, 108–119.

Santner, T. J., Williams, B. J., and Notz, W. I. (2003), *The Design and Analysis of Computer Experiments*, New York: Springer-Verlag.

Schmidt, A. M., and O'Hagan, A. (2003), "Bayesian Inference for Nonstationary Spatial Covariance Structure via Spatial Deformations," *Journal of the Royal Statistical Society*, Ser. B, 65, 745–758.

Silverman, B. W. (1985), "Some Aspects of the Spline Smoothing Approach to Non-Parametric Curve Fitting," *Journal of the Royal Statistical Society*, Ser. B, 47, 1–52.

Stein, M. L. (1999), *Interpolation of Spatial Data*, New York: Springer.

Whaley, R. C., and Petitet, A. (2004), *ATLAS (Automatically Tuned Linear Algebra Software)*, available at *http://math-atlas.sourceforge.net/*.