UNIVERSITY OF CALIFORNIA

SANTA CRUZ

BAYESIAN TREED GAUSSIAN PROCESS MODELS

A dissertation submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

by

Robert B. Gramacy

December 2005

The Dissertation of Robert B. Gramacy is approved:

Professor Herbert K. H. Lee, Chair

Professor Bruno Sansó

Professor David P. Helmbold

Lisa C. Sloan
Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

Abstract

Bayesian Treed Gaussian Process Models

by

Robert B. Gramacy

Computer experiments often require dense sweeps over input parameters to obtain a qualitative understanding of their response. Such sweeps can be prohibitively expensive, and are unnecessary in regions where the response is easily predicted; well-chosen designs could allow a mapping of the response with far fewer simulation runs. Thus, there is a need for computationally inexpensive surrogate models and an accompanying method for selecting small designs. This dissertation explores nonparametric and semiparametric nonstationary modeling methodologies for addressing this need that couples stationary Gaussian processes and (limiting) linear models with treed partitioning. A Bayesian perspective yields an explicit measure of (nonstationary) predictive uncertainty that can be used to guide sampling. As typical experiments are high-dimensional and require large designs, a careful but thrifty implementation is essential. The methodological developments and statistical computing details which make this approach efficient are outlined in detail. In addition to several illustrations using synthetic data, classic nonstationary data analyzed in recent literature are used to validate the model, and the benefit of adaptive sampling is illustrated through a motivating example which involves the computational fluid dynamics simulation of a NASA reentry vehicle.

for Bobie

my fountain of youth

Acknowledgements

There are several people, without whom, this thesis would not have been possible. First in line, of course, is Herbie Lee. I could not have asked for a better advisor. In particular, I appreciated his balanced mix of friendliness and frankness. He was a fun person to work with, and has taught me a great deal about patience, attention to detail, writing, and (let's not forget) statistics. He is someone to emulate and someone I hope to learn a lot more from.

Next in line is Bruno Sansó. Thanks for never ceasing to give me hell! Bruno made sure I had an *authentic* graduate school experience, complete with the nerve wracking feeling that someone is "out to get you" on the advancement exam. But seriously, Bruno is a great resource and a veritable fountain of knowledge about spatial statics whom I highly respect. I am very grateful for his thoughts, comments, and criticisms on drafts of this thesis.

Dave Helmbold has played an important role in every stage of my academic development at UC Santa Cruz (UCSC), from undergraduate and graduate advising, Senior and Masters thesis advising (and reading), and now reading this dissertation. I am grateful for his enduring patience with me, and with whatever problem I happened to be in his office with at the time, be it algorithms, electrical circuits, caching, or treed Gaussian process models.

Thanks to William Macready and NASA for a very interesting problem, and ample access to the resources (and funding) necessary to get the job done.

I would like to thank the faculty of the Applied Math & Statistics department (AMS) at UCSC, especially David Draper. David's introductory course on Bayesian Statistics changed my life. I have enjoyed working with everyone in AMS—faculty and fellow graduate students. I've enjoyed sharing B & C with all of you. I would also like to shout out to Dan Merl, and thank him for reminding me that we have Seminar, for joining me at Kresge for a breakfast burrito, and for teaching me how to clean up after myself.

Graduate school would not have been nearly as much fun had I not met Leah. Thank you, Leah, for being such a dear friend and companion. Thank you for teaching me how to relax, for giving me a sense of perspective, for being brave, and for knowing when to be bold. I love you dearly, and I'm looking forward more adventures together. Next stop, Cambridge.

Finally, thanks Mom and Dad for allowing me to postpone growing up for a few more years. Thanks for always being supportive, for being proud of me, and for setting a good example. And Ryan, if you were pushing 90 years of age, I would have dedicated this to you.

Chapter 1

Introduction

Many complex phenomena are difficult to investigate directly through controlled experiments. Instead, computer simulation is becoming a commonplace alternative to provide insight into such phenomena. However, the drive towards higher fidelity simulation continues to tax the fastest computers, even in highly distributed computing environments. Computational fluid dynamics (CFD) simulations in which fluid flow phenomena are modeled are an excellent example—fluid flows over complex surfaces may be modeled accurately but only at the cost of supercomputer resources. In this thesis I explore the problem of fitting a response surface for a computer model when the experiment can be designed adaptively, i.e., online—a task to which the Bayesian approach is particularly well-suited.

Consider a simulation model which defines a mapping, perhaps non-deterministic, from parameters describing the inputs to one or more output responses. Without an analytic representation of the mapping between inputs and outputs, simulations must be run for many different input configurations in order to build up an understanding of its possible outcomes. This is called a *computer experiment.*

High fidelity computer experiments are usually run on clusters of independent computing agents, or processors (e.g. a `Beowulf` cluster). Agents can process one input configuration at a time. Multiple agents allow several input configurations to be run in parallel, starting and finishing at different, even random, times. The cluster is usually managed by master controller *(emcee)* program that gathers responses from finished simulations, and keeps free agents busy with new inputs. Even in extremely parallel computing environments, computational expense of the simulation and/or high dimensional inputs often prohibit the naïve approach of running the experiment over a dense grid of input configurations. However, computationally inexpensive surrogate models can often be found which provide accurate approximations to the simulation, especially in regions of the input space where the response is easily predicted.

For example, NASA is developing a new re-usable rocket booster called the Langley Glide-Back Booster (LGBB). Much of its development is done with computer models. In particular, NASA is interested in learning about the response in flight characteristics (lift, drag, pitch, side-force, yaw, roll) of the LGBB as a function of three inputs (side slip angle, Mach number, angle of attack). For each input configuration triplet, CFD simulations yield six response outputs. There is interest in being able to automatically and adaptively design the experiment to learn about where response is most interesting, e.g., where uncertainty is largest, and spend relatively more effort sampling in these areas. For example, consider Figure 1.1 which shows the lift response plotted as a function of speed (Mach) and angle of attack (alpha) with the side-slip angle (beta) fixed at zero. The figure illustrates how the characteristics of subsonic flows can be quite different from supersonic flows. Moreover, the transition between subsonic and supersonic is distinctly non-linear and may possibly even be non-differentiable or non-continuous. The CFD simulations in this experiment involve the integration of the inviscid Euler equations over a mesh of 1.4 million cells. Each run of the Euler solver for a given set of

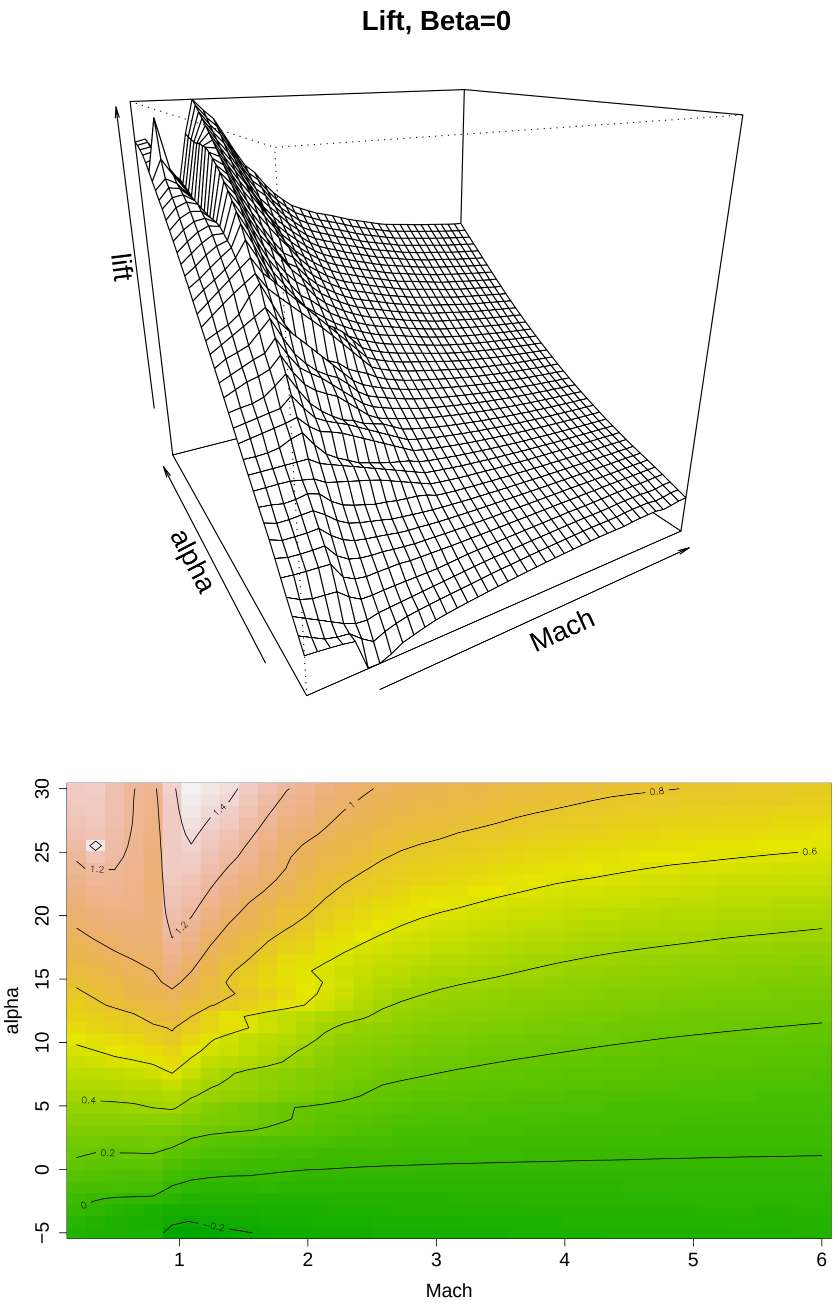


Figure 1.1: Lift plotted as a function of Mach (speed) and alpha (angle of attack) with beta (side-slip angle) fixed to zero. The *top* is a perspective plot, and the *bottom* is an image plot.

parameters takes on the order of 5-20 hours on a high end workstation. Clever sampling could drastically reduce of size the final experimental design, and possibly save thousands of hours of computing time.

The traditional surrogate model used to approximate outputs to computer experiments is the Gaussian process (GP). The GP is conceptually straightforward, easily accommodates prior knowledge in the form of covariance functions, and returns estimates of predictive confidence. In spite of its simplicity, there are three important disadvantages to the standard GP in this setting. Firstly, inference on the GP scales poorly with the number of data points, typically requiring computing time that grows with the cube of the sample size. Secondly, GP models are usually stationary in that the same covariance structure is used throughout the entire input space. In the application of high-velocity computational fluid dynamics, where subsonic flow is quite different than supersonic flow, this limitation is unacceptable. Thirdly, the error (standard deviation) associated with a predicted response under a GP model does not locally depend on any of the previously observed output responses.

All of these shortcomings may be addressed by partitioning the input space into regions, and fitting separate GP models within each region. Partitioning allows for the modeling of nonstationary behavior, and can ameliorate some of the computational demands by fitting models to less data. Finally, a fully Bayesian approach yields uncertainty measures for predictive inference which can help direct future sampling. However, the MCMC required to estimate the parameters of a Bayesian model can be computationally intensive. Careful but thrifty implementation is required to ensure the development of a cost-effective aid in the sequential design of computer experiments.

1.1 What is in this thesis?

This thesis is in three parts and combines work from four research areas in Statistics and Machine Learning. In their own right, each part represents a significant contribution. The common theme and ultimate goal is to describe an efficient model for the sequential design of computer experiments.

The foundation of this work is set in Bayesian hierarchical modeling, model averaging, and Markov chain Monte Carlo (MCMC). Chapter 2 combines stationary Gaussian processes (GPs) and treed partitioning to create treed GPs, implementing a tractable nonstationary model for nonparametric regression. The methodology is illustrated and validated on synthetic data, as well as on a number of classic nonstationary data sets. Chapter 3 exploits a particular Gaussian process parameterization which implements a semiparametric model that treats some or all of the input dimensions as linear, decoupling them from GP correlation function. This approach is dubbed the GP with jumps to the limiting linear model (LLM), or GP LLM for short. The utility of the GP LLM will be made apparent in its own right, however the greatest “bang for your buck” is obtained when combining it with treed partitioning. The result is a uniquely efficient nonstationary semiparametric regression tool.

Finally, Chapter 4 shows how the treed GP LLM can be used as a surrogate model for computer experiments like the NASA LGBB. Techniques from the active learning branch of the Machine Learning community, and the design of experiments branch of the Statistics community, which have been previously applied to stationary GPs, are applied here to treed GPs (and GP LLMs). The key contribution of the treed GP model in this setting is region-specific estimates of model uncertainty which can be used to guide sampling. Together with an asynchronous interface to simulation codes, e.g., CFD solvers which evaluate cases on a supercomputer, the result is a unique methodology and framework for the sequential design of computer experiments, which I call *adaptive sampling.*

Though the chapters naturally build on one another, each has been authored in such a way as to be relatively self-contained, with its own introduction, development, results, and conclusions. Chapter 5 offers a full re-cap, and collects some thoughts about avenues for further research.

1.2 Related work: four key ingredients

The base of this recipe for nonparametric, nonstationary modeling and design of experiments is Bayesian hierarchical modeling and model averaging—a theme that resonates with almost every idea in the following chapters. The three additional ingredients are stationary GPs, treed partitioning, and adaptive sampling or (sequential) design of experiments. All four concepts are briefly outlined below. In some cases, further references and in-depth analysis is left to later chapters. Readers familiar with the above concepts are encouraged to skip ahead to Chapter 2.

1.2.1 Bayesian Model Averaging

The statistical modeling approach in this thesis is distinctly Bayesian. That is, parameters $\boldsymbol{\theta}$ to models $\mathcal{M}$ are given *prior* distributions $p(\boldsymbol{\theta})$, and inference, given data $\mathbf{Y}$, proceeds by combining the prior with the likelihood $p(\mathbf{Y}|\boldsymbol{\theta})$ in Bayes theorem, yielding a *posterior* distribution $p(\boldsymbol{\theta}|\mathbf{Y})$:

$$p(\boldsymbol{\theta}|\mathbf{Y}) = \frac{p(\mathbf{Y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\mathbf{Y})}. \tag{1.1}$$

Prior distributions encode the scientific prior knowledge (or ignorance) of the modeling scientist(s). They can be based on past experimentation, and/or even defined hierarchically to depend on parameters that have their own, separate, hyper-prior distribution. The full specification is typically referred to as a *Bayesian hierarchical model.* Details of the Bayesian approach to statistics, including discussions of merits and criticisms, are available from many sources (Robert, 2001; Cogdon, 2001; Carlin & Louis, 2000; Gelman et al., 1995; Bernardo & Smith, 1994; Press, 1989; Hartigan, 1964; Jeffreys, 1961).

Families of priors which, when combined a likelihood, produce posterior distributions in the same family are called *conjugate.* Conjugate priors are particularly convenient because they necessarily lead to analytically tractable posteriors. If no known conjugate prior exists for the set of parameters $\boldsymbol{\theta}$, then it may be possible to find a *conditionally conjugate* prior distribution for some of the parameters $\tilde{\boldsymbol{\theta}} \subset \boldsymbol{\theta}$, conditional on the others $\boldsymbol{\theta}^- = \boldsymbol{\theta} \setminus \tilde{\boldsymbol{\theta}}$, so that $p(\tilde{\boldsymbol{\theta}}|\mathbf{Y}, \boldsymbol{\theta}^-)$ is the same family as $p(\tilde{\boldsymbol{\theta}}|\boldsymbol{\theta}^-)$.

A key benefit of Bayesian statistical modeling is a full accounting of uncertainty. The posterior distribution implicitly contains a full summary of the estimated model, rather than just point estimates of its parameters. Another nice feature of the Bayesian paradigm is that that which applies to parameters $\boldsymbol{\theta}$ also applies to the joint distribution of models and their parameters $\{\mathcal{M}, \boldsymbol{\theta}_\mathcal{M}\}$. For example, priors on models $p(\mathcal{M})$, implying a probability on its parameters $\boldsymbol{\theta}_\mathcal{M}$, give way to posteriors, again via Bayes theorem:

$$p(\mathcal{M}|\mathbf{Y}) = \frac{p(\mathbf{Y}|\mathcal{M})p(\mathcal{M})}{p(\mathbf{Y})}. \tag{1.2}$$

Model selection can be carried out by finding the maximum *a' posteriori* (MAP) model $\mathcal{M}$—i.e., by finding the mode of the posterior distribution—or through the use of *Bayes factors* (Kass & Raftery, 1995). Model averaging can be carried out by integrating over the space of models $\mathcal{M}$ (Hoeting et al., 1999). Bayes factors are not used in this thesis, but model averaging is used extensively.

Markov chain Monte Carlo

When fully conjugate priors cannot be found to adequately encode prior beliefs about models and parameters, posterior inference usually proceeds by simulation. Markov chain Monte Carlo (MCMC) is the standard choice (Gamerman, 1997; Robert & Castella, 2000; Gilks et al., 1996; Gelman et al., 1995) for posterior inference by simulation, and is the ubiquitous tool of inference in this thesis. Andreiu et al. (2003) provide nice descriptions of MCMC, and other simulation based methods of inference, using a vernacular more familiar to a Machine Learning audience.

The main idea of MCMC is to establish a Markov chain whose stationary distribution is the posterior distribution of interest, and then collect samples from that chain. The transition probabilities from state $\boldsymbol{\theta}_n$ to $\boldsymbol{\theta}_{n+1}$ of the Markov chain, representing samples from the posterior of $\boldsymbol{\theta}$, can be set up in two ways: using the Metropolis-Hastings (MH) (Metropolis et al., 1953; Hastings, 1970), or Gibbs (Geman & Geman, 1984) algorithms.

The MH algorithm proceeds by proposing a new $\boldsymbol{\theta}^*$ from a proposal distribution $q(\boldsymbol{\theta}^*|\boldsymbol{\theta_n})$. The next ($n + 1^{\text{st}}$) sample from the posterior for $\boldsymbol{\theta}$ is chosen based on a ratio of posterior and proposal distributions.

$$\alpha = \min \left\{ 1,\ \frac{p(\mathbf{Y}|\boldsymbol{\theta}^*)p(\boldsymbol{\theta}^*)q(\boldsymbol{\theta}_n|\boldsymbol{\theta}^*)}{p(\mathbf{Y}|\boldsymbol{\theta}_n)p(\boldsymbol{\theta}_n)q(\boldsymbol{\theta}^*|\boldsymbol{\theta}_n)} \right\} \tag{1.3}$$

Equation (1.3) is referred to as the MH acceptance ratio, or simply α. Since a ratio of posteriors is what is of interest here, calculating $p(\mathbf{Y})$ of (1.1) is not required. This is the main benefit of MH sampling, as calculating $p(\mathbf{Y})$ usually requires computing an intractable integral. The randomly proposed $\boldsymbol{\theta}^*$ is accepted or rejected based on α:

$$\boldsymbol{\theta}_{n+1} = \begin{cases} \boldsymbol{\theta}^* & \text{with prob.} \quad \alpha \\ \boldsymbol{\theta}_n & \text{with prob.} \quad 1-\alpha. \end{cases} \tag{1.4}$$

Gibbs sampling is a special case of MH where $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}_n) = p(\boldsymbol{\theta}^*|\mathbf{Y})$, resulting in an acceptance ratio of $\alpha = 1$, so all proposals are accepted. Parameters with conditionally conjugate priors can usually be sampled with Gibbs steps. Those without conditionally conjugate priors generally require MH steps. Mixing of MH and Gibbs samples is allowed in order to obtain samples from the full joint posterior. Enough samples from the posterior distribution of $\boldsymbol{\theta}$ are taken in order to summarize the statistics of inferential interest, e.g., means & variances, medians, predictive means & variances, etc.

When using MCMC for model selection or model averaging, an augmentation of the MH acceptance ratio (1.3) is needed in order to account for possible changes in the dimension of the parameter space in a proposed $\{\mathcal{M}^*, \boldsymbol{\theta}_{\mathcal{M}^*}\}$ compared to the previous model and parameters $\{\mathcal{M}_n, \boldsymbol{\theta}_{\mathcal{M}n}\}$. This is handled by so called reversible jump Markov chain Monte Carlo (RJ-MCMC) (Richardson & Green, 1997). RJ-MCMC augments Eq. (1.3) to include a Jacobian term which accounts for a stretching or shrinking of the volume of the parameter space in moving from $\boldsymbol{\theta}_{\mathcal{M}n}$ to $\boldsymbol{\theta}^*_{\mathcal{M}}$. However, if the proposals are taken from the prior:

$$q(\boldsymbol{\theta}^*_{\mathcal{M}}|\boldsymbol{\theta}_{\mathcal{M}n}) = p(\boldsymbol{\theta}^*_{\mathcal{M}}),$$

then the Jacobian term can be shown to reduce to one, and the MH ratio for RJ-MCMC is the analog of (1.3) for model averaging:

$$\alpha = \min\left\{1,\ \frac{p(\mathbf{Y}|\mathcal{M}^*)p(\mathcal{M}^*)q(\mathcal{M}^*|\mathcal{M}_n)}{p(\mathbf{Y}|\mathcal{M}_n)p(\mathcal{M}_n)q(\mathcal{M}_n|\mathcal{M}^*)}\right\}. \tag{1.5}$$

Similarly, the analog of Eq. (1.4) is used for sampling, replacing $\mathcal{M}$ for $\boldsymbol{\theta}$. Such is the extent to which RJ-MCMC is used in this thesis.

1.2.2 Stationary Gaussian Processes

In a computer experiment, the (possibly multi-dimensional) simulation output $\mathbf{z}(\mathbf{x})$, is typically modeled as (Sacks et al., 1989)

$$\mathbf{z}(\mathbf{x}) = \boldsymbol{\beta}^\top \mathbf{x} + \mathbf{w}(\mathbf{x}) \tag{1.6}$$

for a particular (multivariate) input value $\mathbf{x}$, where $\boldsymbol{\beta}$ are linear trend coefficients, $\mathbf{w}(\mathbf{x})$ is a zero mean random process with covariance $C(\mathbf{x}, \mathbf{x}') = \sigma^2 K(\mathbf{x}, \mathbf{x}')$, and $\mathbf{K}$ is a correlation matrix. Low-order polynomials are sometimes used instead of the simple linear mean $\boldsymbol{\beta}^\top \mathbf{x}$, or the mean process is specified generically, often as $m(\mathbf{x}, \boldsymbol{\beta})$ or $m(\mathbf{x})$ (Stein, 1999). The stationary Gaussian process is a popular example of a model that fits this description, and consequently is the canonical surrogate model used in designing computer experiments (Sacks et al., 1989; Santner et al., 2003).

Gaussian processes (GPs) are a popular method for nonparametric regression and classification. Though the method can be traced back to Kriging (Matheron, 1963), it is only recently that GPs have been broadly applied in Machine Learning. Consider a training set $D = \{\mathbf{x}_i, \mathbf{z}_i\}_{i=1}^N$ of m_X-dimensional input parameters and m_Z-dimensional simulation outputs. The collection of inputs is indicated as the $N \times m_X$ matrix $\mathbf{X}$ whose i^{th} row is $\mathbf{x}_i^\top$. Formally (Stein, 1999), a Gaussian process is a collection of random variables $\mathbf{Z}(\mathbf{x})$ indexed by $\mathbf{x}$ having a jointly Gaussian distribution for any subset of indices. It is specified by a mean $\boldsymbol{\mu}(\mathbf{x}) = E\big(\mathbf{Z}(\mathbf{x})\big)$ and correlation function $K(\mathbf{x}, \mathbf{x}') = \frac{1}{\sigma^2} E\big([\mathbf{Z}(\mathbf{x}) - \boldsymbol{\mu}(\mathbf{x})][\mathbf{Z}(\mathbf{x}') - \boldsymbol{\mu}(\mathbf{x}')]^\top\big)$. Given a set of observations D, the resulting density over outputs at a new point $\mathbf{x}$ has a Normal distribution with

$$\begin{aligned} &\text{mean} && \hat{z}(\mathbf{x}) = \mathbf{k}^\top(\mathbf{x})\mathbf{K}^{-1}\mathbf{Z}, \quad \text{and} \\ &\text{variance} && \hat{\sigma}^2_{\hat{z}}(\mathbf{x}) = \sigma^2[K(\mathbf{x}, \mathbf{x}) - \mathbf{k}^\top(\mathbf{x})\mathbf{K}_N^{-1}\mathbf{k}(\mathbf{x})] \end{aligned} \tag{1.7}$$

where $\mathbf{k}^\top(\mathbf{x})$ is the N-vector whose ith component is $K(\mathbf{x}, \mathbf{x}_i)$, $\mathbf{K}$ is the $N \times N$ matrix with i, j element $K(\mathbf{x}_i, \mathbf{x}_j)$, and $\mathbf{Z}$ is the N-vector of observations with i^{th} component z_i. For simplicity, it is assumed that the output is scalar (i.e., multiple output response are modeled independently, and so effectively $m_Z = 1$) so that the image of the covariance function is a scalar. There are many ways of dealing with multiple responses jointly, such as cokriging (Wackernagel, 2003; Ver Hoef & Barry, 1998) or co-regionalization (Schmidt & Gelfand, 2003). Also, the linear trend term in (1.6) is zero. Later, both will be treated with more generality, though neither cokriging or co-regionalization will be directly addressed in this thesis. It is important to note that the uncertainty, $\sigma^2_{\hat{z}}(\mathbf{x})$, associated with the prediction has no direct dependence on the nearby observed simulation outputs $\mathbf{Z}$.

The correlation matrix $\mathbf{K}$, which is the heart of the GP, is determined by one of a family of parametric correlation functions $K(\cdot, \cdot)$. Examples include the isotropic or separable power or Matérn families, each outlined below. In all cases, the correlation functions used in this thesis have the form

$$K(\mathbf{x}_j, \mathbf{x}_k | g) = K^*(\mathbf{x}_j, \mathbf{x}_k) + g\delta_{j,k}. \tag{1.8}$$

where $\delta_{\cdot,\cdot}$ is the Kronecker delta function, and K^* is a *true* correlation function. The only properties required of correlation functions $K^*(\cdot,\cdot)$, and the correlation matrices $\mathbf{K}^*$ the functions produce, is symmetry ($\mathbf{K}^* = (\mathbf{K}^*)^\top$) and positive semi-definiteness ($\mathbf{a}^\top \mathbf{K}^* \mathbf{a} \geq 0$, for any column-vector $\mathbf{a}$). Valid correlation functions are usually generated as a member of a parametric family. The following subsections highlight elements of (1.8) which are of particular interest in this thesis. A nice general reference for families of correlation functions K^* is provided by Abrahamsen (1997).

The nugget

The g term in the correlation function $K(\cdot,\cdot)$ in Eq. (1.8) is referred to as the *nugget* in the geostatistics literature (Matheron, 1963; Cressie, 1991) and sometimes as *jitter* in Machine Learning literature (Neal, 1997). It must always be positive ($g > 0$), and serves two purposes. Primarily, it provides a mechanism for introducing measurement error into the stochastic process. It arises when considering a model of the form:

$$Z(\mathbf{X}) = m(\mathbf{X}, \boldsymbol{\beta}) + \varepsilon(\mathbf{X}) + \eta(\mathbf{X}), \tag{1.9}$$

where $m(\cdot,\cdot)$ is underlying (usually linear) mean process, $\varepsilon(\cdot)$ is a process covariance whose underlying correlation is governed by K^*, and $\eta(\cdot)$ is simply Gaussian noise. Secondarily, though perhaps of equal practical importance, the nugget (or jitter) prevents $\mathbf{K}$ 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 $K(\mathbf{x}_i, \mathbf{x}_i) > 1$. Appendix B outlines an isomorphic model specification wherein $\mathbf{K}$ depicts honest correlations. Under both specifications K^* does indeed define a valid correlation matrix $\mathbf{K}^*$. For further details please refer to Appendix B.

Power family

A common family of correlation functions is the *isotropic power* family. Correlation functions in this family are *stationary* which means that correlations are measured identically throughout the input domain, and *isotropic* in that correlations $K^*(\mathbf{x}_j, \mathbf{x}_k)$ depend only on a function of the Euclidean distance between $\mathbf{x}_j$ and $\mathbf{x}_k$: $||\mathbf{x}_j - \mathbf{x}_k||$. A common parameterization is

$$K^*_\nu(\mathbf{x}_j, \mathbf{x}_k|d_\nu) = \exp\left\{-\frac{||\mathbf{x}_j - \mathbf{x}_k||^{p_0}}{d}\right\}, \tag{1.10}$$

where $d > 0$ is referred to as the *width* or *range* parameter. The power $0 < p_0 \leq 2$ determines the smoothness of the underlying process, which can either be fixed in advance or estimated. Every process with $0 < p_0 \leq 2$ is continuous at the origin, i.e., when $||\mathbf{x}_j - \mathbf{x}_k|| = 0$, and none, except the Gaussian $p_0 = 2$, is differentiable at the origin. A white noise process, with constant global correlation, is obtained when $p_0 = 0$. When modeling computer experiments, a typical default choice is the Gaussian $p_0 = 2$.

For more on the smoothness properties of the power family of correlation functions, and others, see Alder (1997), Abrahamsen (1997), or Stein (1999)—some relevant highlights of which are motivated and quoted below. Chapter 3 contains a detailed exploration of how the range (d) and nugget (g) parameters interact in order to describe varying degrees of smoothness in the posterior predictive surface.

A straightforward enhancement to the isotropic power family is to employ a unique range parameter d_i in each dimension ($i = 1, \ldots, m_X$). The resulting correlation function is still stationary, but no longer isotropic. A common parameterization is:

$$K^*(\mathbf{x}_j, \mathbf{x}_k|\mathbf{d}) = \exp\left\{-\sum_{i=1}^{m_X} \frac{|x_{ij} - x_{ik}|^{p_0}}{d_i}\right\} \tag{1.11}$$

in which the (non-separable) isotropic exponential family is a special case (when $d_i = d$, for $i = 1, \ldots, m_X$). With the separable power family, one can model correlations in some input variables as stronger than others. However, with added flexibility comes added costs in the form of more parameters to estimate. When the true underlying correlation structure is isotropic, the extra parameters of the separable model represent a sort of overkill, and in terms of efficiency of implementation, a hindrance.

Matérn Family

In a recently published monograph on Spatial statistics (Stein, 1999), Michael Stein strongly suggests using the Matérn family of correlation functions. Correlations in this family are isotropic, and have the form:

$$K_\nu(\mathbf{x}_j, \mathbf{x}_k | \rho, \phi, \alpha) = \frac{\pi^{1/2}\phi}{2^{\rho-1}\Gamma(\rho+1/2)\alpha^{2\rho}}(\alpha||\mathbf{x}_j - \mathbf{x}_k||)^\rho \mathcal{K}_\rho(\alpha||\mathbf{x}_j - \mathbf{x}_k||) \tag{1.12}$$

where $\mathcal{K}_\rho$ is a modified Bessel function of the second kind (Abramowitz & Stegun, 1964). This family of correlation functions are obtained from spectral densities of the form $f(\omega) = \phi(\alpha^2 + \omega^2)^{-\rho-1/2}$. Since the resulting process can shown to be $\lceil \rho \rceil - 1$ times differentiable, ρ can be thought thought of as a smoothness parameter. The ability to specify smoothness is a significant feature of the Matérn family, especially in comparison to the power exponential family which is either nowhere differentiable ($0 < p_0 < 2$) or infinitely differentiable ($p_0 = 2$). Other properties of the Matérn family compared to those of other families are discussed by Paciorek (2003) and Stien (1999). Proper specification or estimation of ρ may shrink the role of a special nugget parameter, or white noise process, in estimating a trade-of between a smoothing or interpolating process. It may also make decomposing $\mathbf{K}$ more numerically stable. Separable versions of the Matérn family also exist.

Estimation

Parameter settings to the correlation function(s) are determined either by maximizing the likelihood, integrating over them, or by taking a Bayesian approach. The usual priors (Gelman et al., 1995) can be placed on the linear ($\boldsymbol{\beta}$) part of the model, including a conjugate inverse-gamma prior for σ^2. Gibbs samples can be obtained for these parameters. Priors also need to be placed on the hyperparameters to the correlation structure $\mathbf{K}$. If little is known in advance about the process, then Objective Bayes priors—vague, reference or Jeffreys—can be used (Berger et al., 2001). They can be sampled using the Metropolis-Hastings algorithm.

It is known that in certain cases the parameters to the Matérn family cannot be estimated consistently. Nonetheless there is a quantity, arguably of greater interest to spatial interpolation than the parameters themselves, that can be estimated consistently (Zhang, 2004). Though conjectured, to my knowledge a similar result has not been shown for the power exponential family. In general I find that when inputs $\mathbf{X}$ are translated and scaled, e.g., to the unit cube—a common practice—the marginal Markov chains for the range and nugget parameters, while correlated, tend to mix well. Further details on (Bayesian) inference for the separable and isotropic power family are are left to Chapter 3.

Alternative GP specifications

An alternative *process-convolution* specification of GPs (Higdon, 2002) has become popular for modeling lower dimensional (e.g. 2-d) space-time models. A duality can be shown (Thiébaux & Pedder, 1987; Thiébaux, 1997; Ver Hoef & Barry, 1998) between a stationary GP with spatial inputs $\mathbf{s}$ and responses $Z(\mathbf{s})$, for $\mathbf{s} \in \Re^m$, and the convolution of a Gaussian white noise process $X(\mathbf{s})$ as

$$Z(\mathbf{s}) = \int_{\Re^m} K(\mathbf{s} - \mathbf{u}) X(\mathbf{u})\, d\mathbf{u}.$$

One of the main advantages of this approach is that inverting an $N \times N$ covariance matrix (as in Eq. (1.7)) is not required. However, the implementation requires that a lattice of kernels $K(\cdot)$ be placed, somewhat densely, throughout the input space. Any savings that comes from not having to invert a covariance matrix is quickly diminished when the dimension (m) of the input space gets large, because the number of kernels needed to adequately fill out the space grows exponentially in m. This prohibits its use in higher dimensions. Initial implementations of the GP model in this thesis actually used this formulation. But, since computer experiments can vary greatly in input dimension, the standard (Kriging) approach, in the end, seemed more appropriate.

1.2.3 Treed Partitioning for Nonstationary Modeling

As motivated above, designing computer experiments can require more flexibility in a surrogate model than is offered by a stationary GP. A nonstationary model seems more appropriate. One way to achieve non-stationarity is to use a partition model—a model which somehow divides up the input space and fits different models to data independently in the regions depicted by the partitions. Treed partitioning is one possible approach. Discussion of other approaches to nonstationary modeling is deferred to the end of this subsection.

Binary treed partition models divide up the input space by making binary splits on the value of a single variable (e.g., speed > 0.8) so that partition boundaries are parallel to coordinate axes. Partitioning is recursive, so each new partition is a sub-partition of a previous one. 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 will then divide only the space below (or above) the midpoint of the first variable, so that there are now three partitions (not four). Since variables may be revisited, there is no loss of generality by using binary splits, as multiple splits on the same variable will be equivalent to a non-binary split.

These sorts of models are often referred to as Classification and Regression Trees (CART) (Breiman et al., 1984). CART has become popular because of its ease of use, clear interpretation, and ability to provide a good fit in many cases.

For example, a tree $\mathcal{T}$ partitions the input space into R non-overlapping 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. Each split in the tree is based on a (randomly) selected dimension $u_j \in \{1, \dots, m_X\}$ and an associated split criterion s_j, so that one of the resulting sub-partitions consists of those observations in $\{\mathbf{X}_\nu, \mathbf{Z}_\nu\}$ with the u_j^{th} parameter less than s_j, and the other contains those observations greater than or equal to s_j. Thus, the structure of the tree is determined by a hierarchy of splitting criteria $\{u_j, s_j\}$, $j = 1, \dots, \lceil R/2 \rceil$.

Figure 1.2 shows an example tree. In this example, D_1 contains $\mathbf{x}$'s whose u_1 coordinate is less than s_1 and whose u_2 coordinate is less than s_2. Like D_1, D_2 has $\mathbf{x}$'s whose coordinate u_1 is less than s_1, but differs from D_1 in that the u_2 coordinate must be bigger than or equal to s_2. Finally, D_3 contains the rest of the $\mathbf{x}$'s differing from those in D_1 and D_2 because the u_1 coordinate of its x's is greater than or equal to s_1. The corresponding response values z accompany the $\mathbf{x}$'s of each region.

The Bayesian approach is straightforward to apply to tree models (Chipman et al., 1998; Denison et al., 1998), provided that one can specify a meaningful prior for the size of the tree. I follow Chipman et al. (1998, 2002) who specify the prior through a tree-generating process. Starting with a null tree (all data in a single partition), a leaf in the tree ($\mathcal{T}$) is split recursively with each node η representing a region of the input space, being split with probability $p_{\text{SPLIT}}(\eta, \mathcal{T}) = a(1 + q_\eta)^{-b}$, where q_η is the depth of η in $\mathcal{T}$ and a and b are parameters chosen to give an appropriate size and spread to the distribution of trees. As part

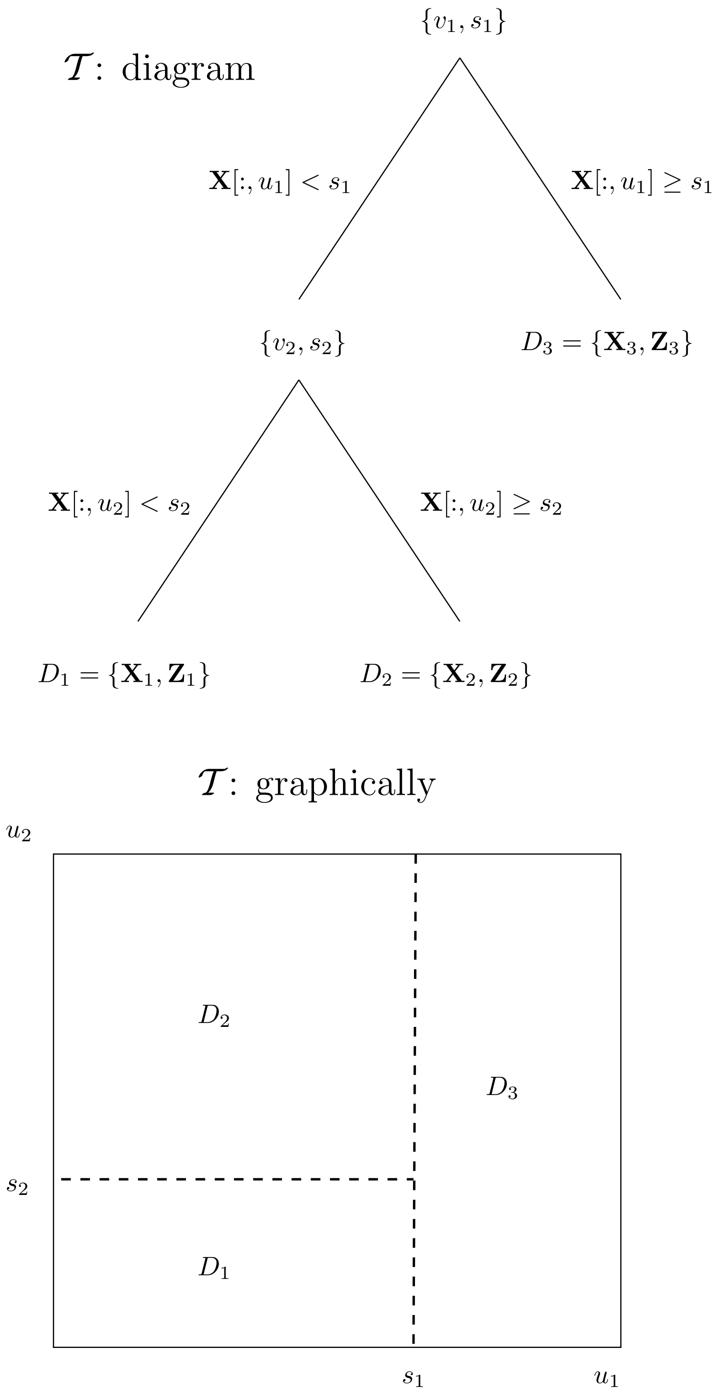

$\mathcal{T}$: diagram
$\{v_1, s_1\}$
$\mathbf{X}[:, u_1] < s_1$
$\mathbf{X}[:, u_1] \geq s_1$
$\{v_2, s_2\}$
$D_3 = \{\mathbf{X}_3, \mathbf{Z}_3\}$
$\mathbf{X}[:, u_2] < s_2$
$\mathbf{X}[:, u_2] \geq s_2$
$D_1 = \{\mathbf{X}_1, \mathbf{Z}_1\}$
$D_2 = \{\mathbf{X}_2, \mathbf{Z}_2\}$
$\mathcal{T}$: graphically
u_2
D_2
D_3
s_2
D_1
s_1
u_1

Figure 1.2: An example tree $\mathcal{T}$ with two splits, resulting in $R = 3$ partitions, shown in a diagram (*top*) and pictorially (*bottom*).

of the process prior, one might further require that each new region have at least a minimal number of data points, since the parameters of the model within the partitions may not be effectively estimated if there are too few points.

Chipman et al. call the prior process for generating splitting locations p_{RULE}. The default choice for p_{RULE} is to have the splitting dimension u and location s chosen randomly from a subset of the locations $\mathbf{X}$ in the u^{th} dimension:

$$\begin{array}{ll} u \in \{1, \dots, m\} & \text{chooses the splitting dimension,} \\ s_j \in \mathbf{X}[:, u] & \text{column } u \text{ of } \mathbf{X} \text{ from the parent region.} \end{array}$$

Integrating out dependence on the tree structure $\mathcal{T}$ can be accomplished via MCMC using the Metropolis Hastings algorithm, with the help of tree-modification proposals called *grow, prune, change*, and *swap* developed by Chipman et al. (1998, 2002). Parameters to the constant (1998) or linear (2002) models used at the leaves of the tree can be integrated out, avoiding the need to use Reversible-Jump MCMC (RJ-MCMC) (Richardson & Green, 1997) usually required in such settings.

Extending the work of Chipman et. al (2002), Chapter 2 describes a model wherein stationary GPs with linear trend are fit independently within each of R regions, $\{r_\nu\}_{\nu=1}^{R}$, depicted by the tree $\mathcal{T}$. Since the parameters to the GP correlation function $K(\cdot, \cdot)$ cannot usually be analytically integrated out, a RJ-MCMC approach will be needed as *grow* and *prune* tree proposals cause the dimension of the parameter space to change. This approach bears some similarity to the models of Kim et al. (2002), who fit separate GPs in each element of a Voronoi tessellation. The treed GP approach is better geared toward problems with a smaller number of distinct partitions, leading to a simpler overall model. Using a Voronoi tessellation allows an intricate partitioning of the space, but has the trade-off of added complexity and can produce a final model that is difficult to interpret. A nice review of Bayesian partition modeling is provided by Denison et al. (2002).

Other approaches to nonstationary modeling

Other approaches to nonstationary modeling include those which use spatial deformations and process convolutions. The idea behind the spatial deformation approach is to map nonstationary inputs in the original, geographical space, into another dispersion space wherein the process is stationary. The approach taken by Sampson & Guttorp (1992) uses thin-plate spline models and multidimensional scaling (MDS) to construct the mapping. Damian et al. (2001) explore a similar methodology from a Bayesian perspective. Schmidt & O'Hagan (2003) also take the Bayesian approach, but put a Gaussian process prior on the mapping.

Rather than mapping inputs into another space, the process convolution approach (Higdon et al., 1999; Fuentes & Smith, 2001; Paciorek, 2003) proceeds by allowing the convolution kernels $K_{\mathbf{s}}(\cdot)$ to vary in parameterization as a function of their location $\mathbf{s} \in \Re^d$. $K_{\mathbf{s}}$ is treated as an unknown, smooth function of $\mathbf{s}$. It is given a prior specification, and estimated along with other parameters of the model, in a fully Bayesian fashion.

A common theme among such nonstationary models is the introduction of meta-structure which ratchets up the flexibility of the model, ratcheting up the computational demands as well. In particular, the nonstationary versions tend to require significantly more computation compared to the base, stationary, version of the same model. This is in stark contrast to the treed approach which introduces a structural mechanism, the tree $\mathcal{T}$, that can actually reduce the computational burden relative to the base model.

1.2.4 Adaptive Sampling

In the world of Machine learning, adaptive sampling would fall under the blanket of a research focus called *active learning*. In the literature (Fine, 1999; Angluin, 1987; Fine et al., 2000; Atlas et al., 1990), active learning, or equivalently *query learning* or *selective sampling*, refers to the situation where a learning algorithm has some, perhaps limited, control over the inputs it trains on. Active learning techniques have been proposed in areas such as computational drug design/discovery to aid in the search for compounds that are active against a biological target (Warmuth et al., 2001; Warmuth et al., 2003). However, I am not aware of any other active learning algorithms that use nonstationary modeling to help select small designs.

In the statistics community, the traditional approach to sequential data solicitation is called *(Sequential) Design of Experiments* (Sacks et al., 1989; Santner et al., 2003; Currin et al., 1988; Welch et al., 1992). Depending on whether the goal of the experiment is inference or prediction, as described by a choice of utility, different algorithms for obtaining optimal designs can be derived. For example, one can choose the Kullback-Leibler distance between the posterior and prior distributions (with parameters $\boldsymbol{\theta}$) as a utility. For Gaussian process models with correlation matrix $\mathbf{K}$, this is equivalent to maximizing $\det(\mathbf{K})$. Subsequently chosen input configurations are called $D-$optimal designs. Choosing quadratic loss leads to what are called $A-$optimal designs. An excellent review of Bayesian approaches to the design of experiments is provided by Chaloner & Verdinelli (1995).

Finding optimal designs can be computationally intensive, especially when the algorithm involves calculating repeated decompositions, inverses, or determinants of large covariance matrices. Often D-optimal designs are chosen from a subset of candidate locations. Maxima in determinant-space are sought via stochastic search, simulated annealing (Andrieu et al., 2003), tabu-search (Glover & Laguna, 1997), genetic algorithms (Hamada et al., 2001), etc. (Welch et al., 1992; Currin et al., 1988; Mitchell, 1974). Determinant-space can have many local maxima. Each algorithm is a variation on a theme: one of proposing to remove a candidate from the design in favor of another, computing the resulting change in determinant, and accepting or rejecting the swap based on the magnitude and direction of the change, preferring those that yield an increase in the value of the determinant. Candidate designs/configurations are usually subsampled from a dense grid. Alternatively, the search for a D-optimal design could be restricted to a subset of a Latin Hypercube (LH) design (Santner et al., 2003) [see below].

An alternative approach to optimal design is to formulate the design problem as a (Bayesian) decision problem (Müeller, 1999). Cast in a decision-theoretic framework, the choice of a design $\mathbf{X}$ is associated with some utility $U(D)$, perhaps encoding A or D-optimality. For a model with parameters $\boldsymbol{\theta}$ and responses (outcomes or future data) $\mathbf{z}_{\mathbf{X}}$, which depend on the design ($\mathbf{X}$), the rational decision-maker seeks to maximize $U(\mathbf{X}) = \int u(\boldsymbol{d}, \boldsymbol{\theta}, \mathbf{z})\ dp_{\mathbf{X}}(\boldsymbol{\theta}, \mathbf{z})$. Finding optimal designs can be non-trivial on at least two fronts. First, the integral above may be analytically intractable. Simulation-based approaches to integration are often the only recourse, especially when the parameter space ($\boldsymbol{\theta}$) is high-dimensional. When it is possible to generate a Monte Carlo sample $(\boldsymbol{\theta}_j, \mathbf{z}_j) \sim p(\boldsymbol{\theta}_j)p_{\mathbf{d}}(\mathbf{z}_j|\boldsymbol{\theta}_j)$ for $j = 1, \ldots, M$, a common technique is to approximate $U(\mathbf{X})$ with $\hat{U}(\mathbf{X}) = \frac{1}{M}\sum_{j=1}^{M} u(\mathbf{X}, \boldsymbol{\theta}_j, \mathbf{y}_j)$.

Second, maximization over the design space can also be quite difficult. Simulation-based approaches to maximization exists as well. One approach is simulated annealing (Glover & Laguna, 1997). Simulated annealing is basically inhomogeneous Markov chain simulation where the sequence of stationary distributions are increasingly focused around the maximum. However, in the case of a joint parameter and design space, maxima obtained via simulated annealing will be for the joint "density" $f(\boldsymbol{\theta}, \mathbf{X})$, rather than the marginal $U(\mathbf{X})$. A cleaner approach blends MCMC integration with simulated annealing to simultaneously addresses maximization and integration (Müller et al., 2004). The result is an algorithm for finding maxima in the marginal "density" of designs $U(\mathbf{X})$ for the most probable of parameterizations $p_{\mathbf{X}}(\boldsymbol{\theta}, \mathbf{Z})$. Another approach would be to obtain samples from the joint parameter and design space, and use a surrogate (curve-fitting) model to approximate the expected utility space $U(\mathbf{X})$ and then use calculus to find an optimal design deterministically (Müeller & Parmigiani, 1995).

Since D-optimal designs involve covariance matrices, a model of covariance is needed. Usually a parametric family is assumed in advance, or a preliminary analysis is used to find maximum likelihood (ML) estimates. In a sequential design, parameters estimated from previous designs can be used. The Bayesian design theoretic approach "chooses" a parameterization and optimal design jointly.

Some other approaches used by the statistics community do not require a model of covariance. These include space-filling designs: e.g. max-min distance and LH designs (Box et al., 1978; Santner et al., 2003). Computing max-min distance designs can also be computationally intensive, whereas LH sampling are easy to compute and results in well-spaced designs relative to random sampling. The `FIELDS` package (Fields Development Team, 2004) available from the Comprehensive `R` Archive Network (R Development Core Team, 2004) implements code for space-filling designs in addition to Kriging and Thin Plate Spline models for spatial interpolation.

To create a LH (McKay et al., 1979) design with n samples in a m_X-dimensional space, one starts with an n^{m_X} gridding of the search space. For each row in the first dimension of the grid, a row in every other dimension is chosen randomly without replacement, so that exactly one sample point appears in each row for each dimension. Within the chosen grid cells, the actual sample point is typically sampled randomly. In one dimension a LH design is equivalent to a grid, but as the number of dimensions grows, the number of points in the LH design stays constant.

Though LH designs usually give nice spacing to the design, there are some degenerate cases (Santner et al., 2003), like diagonal LH designs. LH designs can also be less advantageous in a sequential sampling environment. While LH sampled locations are usually well-dispersed with respect to one another, they might not be well-spaced relative to previously sampled (fixed) locations. Whereas most optimal design methods like D-optimal, A-optimal, and max-min, are more computationally intense, they are easily converted into sequential design methods by simply fixing the locations of samples whose response has already been obtained, and then optimizing only over over new sample locations.

An active learning approach sequential experimental design

There are essentially two active learning approaches to the design of experiments using Gaussian Processes as a surrogate model. The first approach tries to maximize the information gained about model parameters by selecting from a pool of candidates $\tilde{\mathbf{X}}$, the location $\tilde{\mathbf{x}} \in \tilde{\mathbf{X}}$ which has the greatest standard deviation in predicted output. This approach, called ALM for Active Learning–Mackay, has been shown to approximate maximum expected information designs (MacKay, 1992).

An alternative algorithm, called ALC for Active Learning–Cohn, is to select $\tilde{\mathbf{x}}$ minimizing the expected squared error averaged over the input space (Cohn, 1996). The global reduction in variance, given that the location $\tilde{\mathbf{x}}$ is added into the data, is obtained by averaging over the reduction in predictive variance at other locations $\mathbf{y}$:

$$\begin{aligned}\Delta\hat{\sigma}^2(\tilde{\mathbf{x}}) &= \int_{\mathbf{y}} \Delta\hat{\sigma}^2_{\mathbf{y}}(\tilde{\mathbf{x}}) \\ &= \int_{\mathbf{y}} \hat{\sigma}^2_{\mathbf{y}} - \hat{\sigma}^2_{\mathbf{y}}(\tilde{\mathbf{x}}) \\ &= \int_{\mathbf{y}} \frac{\sigma^2 \left[\mathbf{k}^\top(\mathbf{y})\mathbf{K}_N^{-1}\mathbf{k}(\tilde{\mathbf{x}}) - K(\tilde{\mathbf{x}},\mathbf{y})\right]^2}{K(\tilde{\mathbf{x}},\tilde{\mathbf{x}}) - \mathbf{k}(\tilde{\mathbf{x}})^\top \mathbf{K}_N^{-1}\mathbf{k}(\tilde{\mathbf{x}})}.\end{aligned}$$

In practice the integral in the above equation is really a sum over a grid of locations $\tilde{\mathbf{Y}}$, as are the candidates $\tilde{\mathbf{x}} \in \tilde{\mathbf{X}}$, usually with $\tilde{\mathbf{Y}} = \tilde{\mathbf{X}}$, and the parameterization to the model, i.e. $K(\cdot,\cdot)$ and σ^2, is known in advance. A comparison between ALC and ALM using standard GPs appears in (Seo et al., 2000).

Computer Experiments

There are some peculiarities in the tradition of using Gaussian processes and standard design of experiment techniques in the literature of (sequential) design and analysis of computer experiments (SDACE). For example, the separable power family (1.11) of correlation functions are the weapon of choice (Santner et al., 2003). This is a generally sensible approach, especially since the isotropic family is a special case of the separable. Chapter 5 shows how the NASA LGBB data is clearly separable in the sense that correlation in the speed (Mach) input is clearly different than correlation in angle of attack (alpha), while correlation for varying side-slip angle (beta) is nearly perfect. Using a separable correlation function for this data is reasonable. The mostly application-oriented SDACE community often finds that the nice theoretical implications of the Matérn family of correlation functions (1.12) do not outweigh their additional computational requirements. Smoothness and noise considerations are apparently of less concern when the data are output from computer code, rather than, say, observations from nature or from a physical experiment.

So in addition, most literature on the DACE (Santner et al., 2003; Sacks et al., 1989; Chaloner & Verdinelli, 1995) deliberately omit the nugget parameter on grounds that computer experiments are deterministic (never noisy). Thus they consider only models of the form:

$$Z(\mathbf{X}) = m(\mathbf{X}, \boldsymbol{\beta}) + \varepsilon(\mathbf{X})$$

omitting the explicit noise component $\eta(\mathbf{x})$ from (1.9). However, there are many reasons why one may wish to study a computer experiment, though technically deterministic, with a model that includes an explicit independent noise component. In particular, the experiment may, in fact, be non-deterministic. Researchers at NASA remark that their CFD solvers are often started with random initial conditions, involve forced random restarts when diagnostics indicate that convergence is poor, and that input configurations arbitrarily close to one another often fail to achieve the same estimated convergence, even after satisfying the same stopping criterion. Thus a conventional GP model without a small-distance noise process, e.g. a nugget, can be a mismatch to such inherently non-smooth data.

Numerical stability in decomposing covariance matrices has been cited (Neal, 1997) as sufficient justification for including a nugget (or *jitter*) parameter. Illustrations and further comments are deferred to Chapter 3.

Other approaches to designing computer experiments

Some traditional Bayesian & non-Bayesian approaches to surrogate modeling and design for computer experiments can be found in (Sebastiani & Wynn, 2000; Welch et al., 1992; Currin et al., 1988; Currin et al., 1991; Mitchell & Morris, 1992; Sacks et al., 1989; Bates et al., 1996). References for the Bayesian approach usually include the landmark papers by

Kennedy & O'Hagan et al. (1999, 2000, 2001). More recently, an approach using stationary GPs, which bears some similarity to the approach taken in this thesis, has proposed using a so called *spatial aggregate language* to aid in an *active data mining* of the input space of the experiment (Ramakrishnan et al., 2005). However, as will become evident in the following chapters, the methods developed in this thesis are in contrast more than they are similar to those in the standard, even recent, approaches in the literature, especially in terms of design. Motivations for fresh approach to SDACE range from the inadequate nature of standard surrogate models (both in terms of speed and modeling capacity) to the challenges inherent in adapting standard optimal design techniques to modern computer experiments which tend to run on highly parallel and distributed, and certainly no less expensive, supercomputers.

Chapter 2

Treed Gaussian Process Models

In this chapter the treed Gaussian Process model (treed GP for short) is described in detail. The chapter concludes with experiments to validate the model as a sensible and efficient approach to nonstationary and nonparametric regression. Extending the work of Chipman et al. (1998, 2002), stationary GP models with linear trend are fit 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 reversible-jump Markov chain Monte Carlo (RJ-MCMC) (Richardson & Green, 1997). As in Chipman et al. (1998, 2002) the prior for $\mathcal{T}$ is specified through a tree-generating process. Starting with a null tree (all data in a single partition), the tree $\mathcal{T}$ is probabilistically split recursively, with each partition η being split with probability $p_{\mathrm{SPLIT}}(\eta, \mathcal{T}) = a(1 + q_\eta)^{-b}$ where q_η is the depth of η in $\mathcal{T}$ and a and b are parameters chosen to give an appropriate size and spread to the distribution of trees. The split location p_{RULE} is chosen uniformly from the data locations $\mathbf{X}$ as possible dividing points between the two new regions. Prediction is conditioned on the tree structure, and is averaged over in the posterior to get a full accounting of uncertainty.

2.1 Hierarchical Model

A tree $\mathcal{T}$ recursively partitions the input space into into R non-overlapping 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. A hierarchical generative model for R stationary GPs with linear trend is specified on data D_ν in each region $\{r_\nu\}_{\nu=1}^R$. For a particular region ν, the hierarchical generative model is

$$\begin{aligned}
\mathbf{Z}_\nu|\boldsymbol{\beta}_\nu, \sigma_\nu^2, \mathbf{K}_\nu &\sim N_{n_\nu}(\mathbf{F}_\nu\boldsymbol{\beta}_\nu, \sigma_\nu^2\mathbf{K}_\nu), \\
\boldsymbol{\beta}_\nu|\sigma_\nu^2, \tau_\nu^2, \mathbf{W}, \boldsymbol{\beta}_0 &\sim N_{m_X}(\boldsymbol{\beta}_0, \sigma_\nu^2\tau_\nu^2\mathbf{W}) \\
\boldsymbol{\beta}_0 &\sim N_{m_X}(\boldsymbol{\mu}, \mathbf{B}), \\
\sigma_\nu^2 &\sim IG(\alpha_\sigma/2, q_\sigma/2), \\
\tau_\nu^2 &\sim IG(\alpha_\tau/2, q_\tau/2), \\
\mathbf{W}^{-1} &\sim W((\rho\mathbf{V})^{-1}, \rho),
\end{aligned} \tag{2.1}$$

with $\mathbf{F}_\nu = (\mathbf{1}, \mathbf{X}_\nu)$, and $\mathbf{W}$ is a $(m_X+1)\times(m_X+1)$ matrix. N, IG, and W are the (Multivariate) Normal, Inverse-Gamma, and Wishart distributions, respectively. Constants $\boldsymbol{\mu}, \mathbf{B}, \mathbf{V}, \rho, \alpha_\sigma, q_\sigma, \alpha_\tau, q_\tau$. are treated as known.

The hierarchical model (2.1) specifies a multivariate normal likelihood with linear trend coefficients $\boldsymbol{\beta}_\nu$, variance σ_ν^2 and $N \times N$ 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 $\sigma_\nu^2\tau_\nu^2$.

The GP correlation structure $\mathbf{K}_\nu$ for each partition r_ν is chosen either from the isotropic power family (1.10), or separable power family (1.11), with a fixed power p_0, but unknown (random) range and nugget parameters. Most of the discussion in this chapter is presented without reference to the mechanism used to construct $\mathbf{K}_\nu$. However, the tacit assumption is that the correlation function takes the form $K_\nu(\mathbf{x}_j, \mathbf{x}_k) = K_\nu^*(\mathbf{x}_j, \mathbf{x}_k) + g_\nu\delta_{j,k}$, where $\delta_{\cdot,\cdot}$ is the Kronecker delta function, and K^*_ν is a *true* correlation representative from a parametric family. Priors which encode a belief that the global covariance structure is nonstationary are chosen for parameters to K^*_ν and g_ν. Further discussion of these priors, particularly pertaining to the power family, is deferred until the next chapter.

There is no explicit mechanism in the model (2.1) to ensure that the process near the boundary of two adjacent regions is continuous across the partitions depicted by $\mathcal{T}$. In fact, conditional on a single tree ($\mathcal{T}$), the transition between the posterior predictive distributions across partition boundaries is strictly discontinuous [further discussion deferred to Section 2.3].

I chose to include the nugget (g) in the correlation model for completeness, but also in light of the discussion at the end of Section 1.2.4 about the dubiousness of treating computer experiments as deterministic. More practically, the nugget, or jitter (Neal, 1997) component is helpful for insuring against the numerical instability of inverting and decomposing $\mathbf{K}$.

2.2 Estimation

The data $D_\nu = \{\mathbf{X}, \mathbf{Z}\}_\nu$ are used to estimate the GP parameters $\boldsymbol{\theta}_\nu \equiv \{\boldsymbol{\beta}, \sigma^2, \mathbf{K}\}_\nu$, for $\nu = 1, \ldots, R$. Parameters to the hierarchical priors ($\boldsymbol{\theta}_0 = \{\mathbf{W}, \beta_0, \boldsymbol{\gamma}\}$) depend only on $\{\boldsymbol{\theta}_\nu\}_{\nu=1}^R$. Conditional on the tree $\mathcal{T}$, the full set of parameters is denoted as $\boldsymbol{\theta} = \boldsymbol{\theta}_{\mathbf{0}} \cup \bigcup_{\nu=1}^R \boldsymbol{\theta}_\nu$.

Samples from the posterior distribution of $\boldsymbol{\theta}$ are gathered using Markov chain Monte Carlo (MCMC) (Gelman et al., 1995) by first conditioning on the hierarchical priors $\boldsymbol{\theta}_0$ and drawing $\boldsymbol{\theta}_\nu | \boldsymbol{\theta}_0$ for $\nu_1, \ldots, \nu_r$, and then $\boldsymbol{\theta}_0$ is drawn as $\boldsymbol{\theta}_0 | \bigcup_{\nu=1}^R \boldsymbol{\theta}_\nu$. Section 2.2.1 gives the details. In short, all parameters can be sampled with Gibbs steps, except those which parameterize the covariance function $K(\cdot, \cdot)$. Parameters which describe $\mathbf{K}$ require Metropolis-Hastings (MH) draws.

Section 2.2.2 shows how RJ-MCMC (Richardson & Green, 1997) is used to gather samples from the joint posterior of $(\boldsymbol{\theta}, \mathcal{T})$ by alternately drawing $\boldsymbol{\theta}|\mathcal{T}$ and then $\mathcal{T}|\boldsymbol{\theta}$ using a superset of the tree operations from Chipman et al.

2.2.1 GPs given tree ($\mathcal{T}$)

Finding full conditionals is a good first step towards efficient sampling. Full conditionals for the parameters associated with the linear trend are listed first. Since they have conditionally conjugate priors, these can be sampled using Gibbs steps. Some parameters $(\{\mathbf{K}, \sigma^2\}_\nu)$ are sampled more efficiently if their full conditionals can be marginalized by analytically integrating out dependence on other parameters. The full derivations are included in Appendix A.1.

The linear regression parameters $\boldsymbol{\beta}_\nu$ have a conditionally conjugate multivariate normal posterior distribution:

$$\boxed{\boldsymbol{\beta}_\nu|\text{rest} \sim N(\tilde{\boldsymbol{\beta}}_\nu, \sigma_\nu^2 \mathbf{V}_{\tilde{\beta}_\nu})} \tag{2.2}$$

where

$$\mathbf{V}_{\tilde{\beta}_\nu} = (\mathbf{F}_\nu^\top \mathbf{K}_\nu^{-1} \mathbf{F}_\nu + \mathbf{W}^{-1}/\tau_\nu^2)^{-1}, \tag{2.3}$$

$$\tilde{\boldsymbol{\beta}}_\nu = \mathbf{V}_{\tilde{\beta}_\nu} (\mathbf{F}_\nu^\top \mathbf{K}_\nu^{-1} \mathbf{Z}_\nu + \mathbf{W}^{-1} \boldsymbol{\beta}_0/\tau_\nu^2). \tag{2.4}$$

Similarly for the hierarchical mean regression parameters $\boldsymbol{\beta}_0$:

$$\boxed{\boldsymbol{\beta}_0|\text{rest} \sim N(\tilde{\boldsymbol{\beta}}_0, \boldsymbol{V}_{\tilde{\beta}_0})} \tag{2.5}$$

where

$$\mathbf{V}_{\tilde{\beta}_0} = \left(\mathbf{B}^{-1} + \mathbf{W}^{-1} \sum_{i=0}^{r} (\sigma_\nu \tau_\nu)^{-2} \right)^{-1} \tag{2.6}$$

$$\tilde{\boldsymbol{\beta}}_0 = \mathbf{V}_{\tilde{\beta}_0} \left(\mathbf{B}^{-1}\mu + \mathbf{W}^{-1} \sum_{i=1}^{r} \boldsymbol{\beta}_\nu (\sigma_\nu \tau_\nu)^{-2} \right). \tag{2.7}$$

The linear variance parameter τ^2 has a conditionally conjugate inverse-gamma posterior:

$$\boxed{\tau_\nu^2|\text{rest} \sim IG((\alpha_\tau + m)/2, (q_\tau + b_\nu)/2)} \tag{2.8}$$

where

$$b_\nu = (\boldsymbol{\beta}_\nu - \boldsymbol{\beta}_0)^\top \mathbf{W}^{-1} (\boldsymbol{\beta}_\nu - \boldsymbol{\beta}_0)/\sigma_\nu^2. \tag{2.9}$$

The linear model covariance matrix $\mathbf{W}$ has a conditionally conjugate inverse-Wishart posterior:

$$\boxed{\mathbf{W}^{-1}|\text{rest} \sim W\left(\rho\mathbf{V}{+}\mathbf{V}_{\hat{T}}, \rho + r\right)} \tag{2.10}$$

where

$$\mathbf{V}_{\hat{T}} = \sum_{i=1}^{r} \frac{1}{(\sigma_\nu \tau_\nu)^2} (\boldsymbol{\beta}_\nu - \boldsymbol{\beta}_0)(\boldsymbol{\beta}_\nu - \boldsymbol{\beta}_0)^\top. \tag{2.11}$$

Analytically integrating out $\boldsymbol{\beta}$ and σ^2 gives a marginal posterior for $\mathbf{K}_\nu$ which is the result of $K(\cdot,\cdot)$ applied to all pairs of input locations from $\mathbf{X}_\nu$, and improves mixing of the Markov chain (Berger et al., 2001). As before, I shall simply quote the results here, and leave the details to Appendix A.2.

$$p(\mathbf{K}_\nu|\mathbf{t}_\nu, \boldsymbol{\beta}_0, \mathbf{W}, \tau^2) =$$
$$\left(\frac{|\mathbf{V}_{\tilde{\beta}_\nu}|}{(2\pi)^{n_\nu}|\mathbf{K}_\nu||\mathbf{W}|\tau^{2m}}\right)^{\frac{1}{2}} \frac{(q_\sigma/2)^{\alpha_\sigma/2}}{\left[(q_\sigma+\psi_\nu)/2\right]^{(\alpha_\sigma+n_\nu)/2}} \frac{\Gamma\left[(\alpha_\sigma+n_\nu)/2\right]}{\Gamma\left[\alpha_\sigma/2\right]} p(\mathbf{K}_\nu), \tag{2.12}$$

where

$$\psi_\nu = \mathbf{Z}_\nu^\top \mathbf{K}_\nu^{-1} \mathbf{Z}_\nu + \boldsymbol{\beta}_0^\top \mathbf{W}^{-1} \boldsymbol{\beta}_0/\tau^2 - \tilde{\boldsymbol{\beta}}_\nu^\top \mathbf{V}_{\tilde{\beta}_\nu}^{-1} \tilde{\boldsymbol{\beta}}_\nu. \tag{2.13}$$

Eq. (2.12) can be used to iteratively obtain draws for the parameters of $K(\cdot,\cdot)$ in region ν via Metropolis-Hastings (MH), or as part of the acceptance ratio for proposed modifications to $\mathcal{T}$ [see Section 2.2.2]. Many terms in (2.12) cancel when examining the MH acceptance ratio for $\mathbf{K}_\nu$ in isolation. Dropping constants that would be common in the numerator and denominator of the MH acceptance ratio for a proposed $\mathbf{K}_\nu$ results in the simplified posterior

$$p(\mathbf{K}_\nu|\mathbf{Z}_\nu, \boldsymbol{\beta}_0, \tau_\nu^2, \mathbf{W}) \propto p(d_\nu, g_\nu) \times \left(\frac{|\mathbf{V}_{\tilde{\beta}_\nu}|}{|\mathbf{K}_\nu|}\right)^{\frac{1}{2}} \times \left(\frac{q_\sigma+\psi_\nu}{2}\right)^{-\frac{\alpha_\sigma+n_\nu}{2}}. \tag{2.14}$$

Any hyperparameters to $K(\cdot,\cdot)$ would also require MH draws. Dropping the prior $p(d_\nu, g_\nu)$ gives an integrated likelihood (Berger et al., 2001).

The conditional distribution of σ_ν^2 with $\boldsymbol{\beta}_\nu$ integrated out is

$$\sigma_\nu^2|d_\nu, g, \boldsymbol{\beta}_0, \mathbf{W} \sim IG((\alpha_\sigma+n_\nu)/2, (q_\sigma+\psi_\nu)/2) \tag{2.15}$$

which allows Gibbs sampling. The full derivation of (2.15) is also included in Appendix A.2.

2.2.2 Tree ($\mathcal{T}$)

Integrating out dependence on the tree structure ($\mathcal{T}$) is accomplished by reversible-jump MCMC (RJ-MCMC) (Richardson & Green, 1997). The tree operations used—*grow, prune, change*, and *swap*—are similar to those in Chipman et al. (1998). Tree proposals can change the size of the parameter space ($\boldsymbol{\theta}$). To keep things simple, proposals for new parameters—via an increase in the number of partitions R—are drawn from their priors, thus eliminating the Jacobian term usually present in RJ-MCMC. New splits are chosen uniformly from the set of marginalized input locations ($\mathbf{X}$).

Swap and *change* tree operations are straightforward because the number of partitions, and thus parameters, stays the same. 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 u^{th} column of $\mathbf{X}$. This is accomplished by sampling s' uniformly from the set $\{u_\nu, s_\nu\}_{\nu=1}^{\lceil R/2 \rceil} \times \{+, -\}$. Parameters $\boldsymbol{\theta}_r$ in regions below the split-point $\{u, s'\}$ are held fixed. Uniform proposals and priors on split-points cause the MH acceptance ratio for *change* to reduce to a simple likelihood ratio.

A *swap* operation proposes changing the order in which two adjacent parent–child (internal) nodes split up the inputs. Basically, an internal parent–child node pair is picked at random from the tree and their splitting rules are swapped. When both child splitting rules are the same, Chipman et al. (1998) propose jointly swapping the parent with both of its children. I have found that this situation is rare in practice, especially for continuously defined inputs (in $\Re^d$) with GP regression models at the leaves. So instead, I have modified *swap* for the following, more common, situation. That is, swaps which are proposed on parent-child internal nodes which split on the same variable are always rejected because a child region below both parents becomes empty after the operation. Figure 2.1 gives an illustration. However, if instead a *rotate* operation from Binary Search Trees (BSTs) is performed, the proposal will

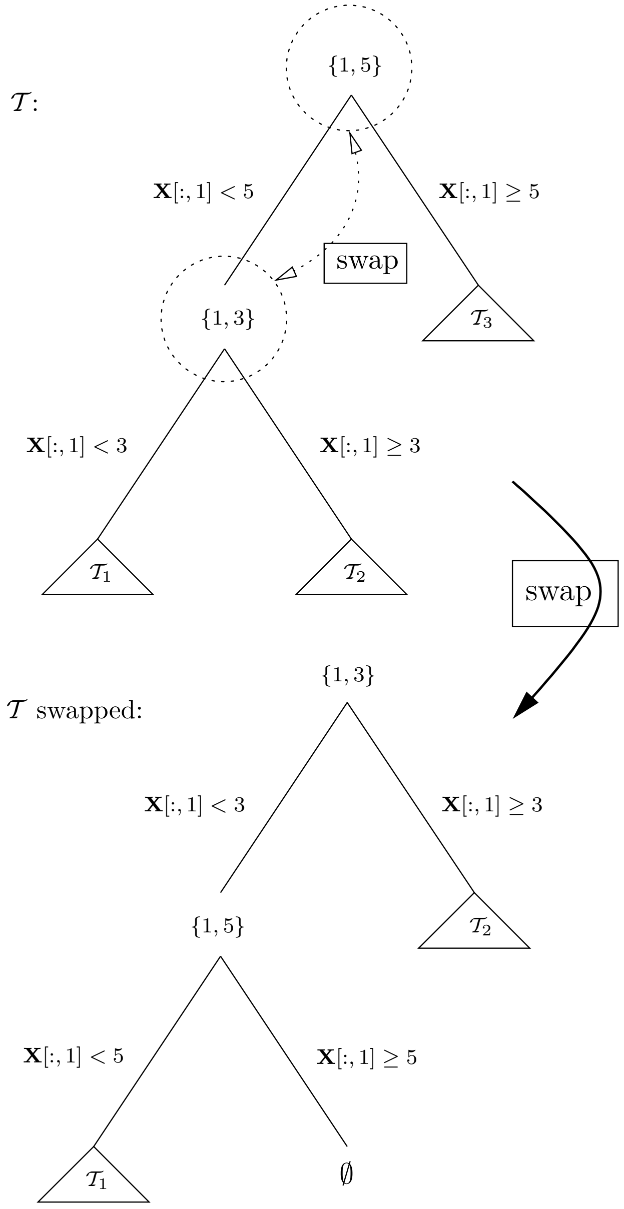


Figure 2.1: Swapping on the same variable is always rejected because one of the leaves corresponds to an empty region. $\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3$ are arbitrary sub-trees (could be leaves).

almost always accept. Rotations are a way of adjusting the configuration and height of a BST without violating the BST property. *Red-Black Trees* make extensive use of *rotate* operations (Cormen et al., 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 it escape local minima in the marginal posterior of $\mathcal{T}$. Figure 2.2 shows an example of a successful right-rotation where the swap of Figure 2.1 fails. Since the partitions at the leaves remain unchanged, the likelihood ratio of a proposed rotate is always 1. The only "active" part of the MH acceptance ratio is the prior on $\mathcal{T}$, preferring trees of minimal depth. Still, calculating the acceptance ratio for a *rotate* is non-trivial because the depth of *two* of its sub-tress change. Sub-trees $\mathcal{T}_1$ and $\mathcal{T}_3$ of Figure 2.2 change depth, either increasing or decreasing respectively, depending on the direction of the rotation. In a right-rotate, nodes in $\mathcal{T}_1$ decrease in depth, while those in $\mathcal{T}_3$ increase. 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 which decrease in depth, then the MH acceptance ratio for a rotate is

$$\begin{aligned}\frac{p(\mathcal{T}^*)}{p(\mathcal{T})} &= \frac{p(\mathcal{T}_1^*)p(\mathcal{T}_3^*)}{p(\mathcal{T}_1)p(\mathcal{T}_3)} \\ &= \frac{\prod_{\eta\in I_i} a(2+q_\eta)^{-b} \prod_{\eta\in I_\ell}[1-a(2+q_\eta)^{-b}]}{\prod_{\eta\in I_i} a(1+q_\eta)^{-b} \prod_{\eta\in I_\ell}[1-a(1+q_\eta)^{-b}]} \times \\ &\times \frac{\prod_{\eta\in D_i} aq_\eta^{-b} \prod_{\eta\in D_\ell}[1-aq_\eta^{-b}]}{\prod_{\eta\in D_i} a(1+q_\eta)^{-b} \prod_{\eta\in D_\ell}[1-a(1+q_\eta)^{-b}]}.\end{aligned} \tag{2.16}$$

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

Grow and *prune* operations are more complex because they add or remove partitions, causing a change in the dimension of the parameter space. The first step for either operation

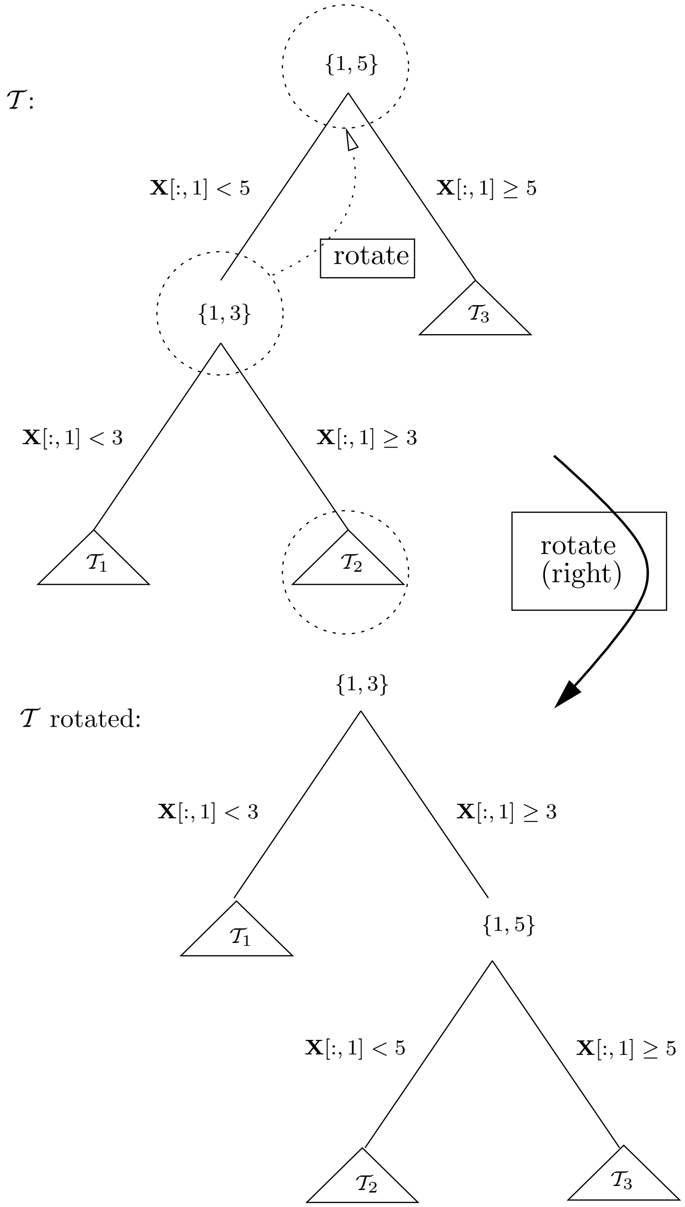


Figure 2.2: Rotating on the same variable is almost always accepted. $\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3$ are arbitrary sub-trees (could be leaves).

is to select a leaf node (for *grow*), or the parent of a pair of leaf nodes (for *prune*). Leaves are chosen uniformly from the set of valid candidates. 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 absorbed by the parent region, or discarded. When evaluating the MH acceptance ratio for either operation, the linear model parameters $\{\boldsymbol{\beta},\sigma^2\}_r$ are integrated out as in (2.12). 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 correlation function from the prior. Symmetrically, *prune* operations randomly select parameters from $K(\cdot,\cdot)$ for the consolidated node from one of the children being absorbed. If the *grow* or *prune* operation is accepted, σ_r^2 can next be drawn from its marginal posterior, with $\boldsymbol{\beta}_r$ integrated out, after which draws for $\boldsymbol{\beta}_r$ and the other parameters for the r^{th} region can then proceed as usual.

Let $\{\mathbf{X},\mathbf{Z}\}$ be the data at the new parent node η at depth q_η, and $\{\mathbf{X}_1,\mathbf{Z}_1\}$ and $\{\mathbf{X}_2,\mathbf{Z}_2\}$ be the new child data at depth $q_\eta+1$ created by the new split $\{u,s\}$. Also, let $\mathcal{P}$ be the set of pruneable nodes of $\mathcal{T}$, and $\mathcal{G}$ the number of growable nodes respectively. The Metropolis-Hastings acceptance ratio for *grow* is:

$$\frac{|\mathcal{P}|+1}{|\mathcal{G}|} \times \frac{a(1+q_\eta)^{-b}(1-a(2+q_\eta)^{-b})^2}{1-a(1+q_\eta)^{-b}} \times \frac{p(\mathbf{K}_1,|\mathbf{Z}_1,\boldsymbol{\beta}_0,\tau_1^2,\mathbf{W})p(\mathbf{K}_2|\mathbf{Z}_2,\boldsymbol{\beta}_0,\tau_2^2,\mathbf{W})}{p(\mathbf{K}|\mathbf{Z},\boldsymbol{\beta}_0,\tau^2,\mathbf{W})}.$$

The *prune* operation is analogous:

$$\frac{|\mathcal{G}|+1}{|\mathcal{P}|} \times \frac{p(\mathbf{K}|\mathbf{Z},\boldsymbol{\beta}_0,\tau^2,\mathbf{W})}{p(\mathbf{K}_1|\mathbf{Z}_1,\boldsymbol{\beta}_0,\tau_1^2,\mathbf{W})p(\mathbf{K}_2|\mathbf{Z}_2,\boldsymbol{\beta}_0,\tau_2^2,\mathbf{W})} \times \frac{1-a(1+d_\eta)^{-b}}{(1-a(2+d_\eta)^{-b})^2 a(1+d_\eta)^{-b}}.$$

Note that in the above two acceptance ratios for *grow* and *prune* operations, 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 $p(\mathbf{K}_2|\mathbf{Z}_2, \boldsymbol{\beta}_0, \tau_2^2, \mathbf{W})$ must be evaluated using the formula in (2.12), *not* the simplified one in (2.14). This is because the terms canceled from (2.12) do not occur the same number of times in the numerator and denominator. Using (2.14) would cause the ratio to be off by a constant factor.

2.3 Treed GP Prediction (Kriging)

Prediction under the above GP model, called Kriging (Matheron, 1963) in the geostatistics community, is straightforward (Hjort & Omre, 1994). The predicted value of $z(\mathbf{x} \in r_\nu)$ is normally distributed with

$$\begin{aligned} \text{mean} \qquad \hat{z}(\mathbf{x}) &= E(\mathbf{Z}(\mathbf{x})|\text{ data}, \mathbf{x} \in D_\nu) \\ &= \mathbf{f}^\top(\mathbf{x})\tilde{\boldsymbol{\beta}}_\nu + \mathbf{k}_\nu(\mathbf{x})^\top \mathbf{K}_\nu^{-1}(\mathbf{Z}_\nu - \mathbf{F}_\nu \tilde{\boldsymbol{\beta}}_\nu), \end{aligned} \tag{2.17}$$

$$\begin{aligned} \text{and variance} \qquad \hat{\sigma}(\mathbf{x})^2 &= \mathrm{Var}(\mathbf{z}(\mathbf{x})|\text{ data}, \mathbf{x} \in D_\nu) \\ &= \sigma_\nu^2[\kappa(\mathbf{x}, \mathbf{x}) - \mathbf{q}_\nu^\top(\mathbf{x})\mathbf{C}_\nu^{-1}\mathbf{q}_\nu(\mathbf{x})], \end{aligned} \tag{2.18}$$

$$\begin{aligned} \text{where} \qquad \mathbf{C}_\nu^{-1} &= (\mathbf{K}_\nu + \tau_\nu^2 \mathbf{F}_\nu \mathbf{W} \mathbf{F}_\nu^\top)^{-1} \\ \mathbf{q}_\nu(\mathbf{x}) &= \mathbf{k}_\nu(\mathbf{x}) + \tau_\nu^2 \mathbf{F}_\nu \mathbf{W}_\nu \mathbf{f}(\mathbf{x}) \\ \kappa(\mathbf{x}, \mathbf{y}) &= K_\nu(\mathbf{x}, \mathbf{y}) + \tau_\nu^2 \mathbf{f}^\top(\mathbf{x}) \mathbf{W} \mathbf{f}(\mathbf{y}) \end{aligned} \tag{2.19}$$

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

Notice that the predictive mean equations use $\tilde{\boldsymbol{\beta}}_\nu$, the posterior mean estimate of

$\tilde{\boldsymbol{\beta}}_\nu$, not $\boldsymbol{\beta}_\nu$ itself. To use $\boldsymbol{\beta}_\nu$ instead one must employ the predictive variance relation in (1.7). Also, one must be careful when using the above Kriging equations with the definition of the correlation matrix $\mathbf{K}$ as given in (1.8). In particular, the nugget term only applies when computing the correlation between a data location and itself. It does not apply for duplicate locations with the same coordinates. For more details see Appendix B.1.

As alluded to briefly in Section 2.1, the posterior predictive surface described in Eqs. (2.17–2.18), conditional on a particular tree ($\mathcal{T}$), is discontinuous across the partition boundaries of $\mathcal{T}$. However, in the aggregate of samples collected from the joint posterior distribution of $\{\mathcal{T}, \boldsymbol{\theta}\}$, samples gathered from the posterior predictive distribution tend 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. Even though each realization of the Kriging equations for $\boldsymbol{\theta}_\nu|\mathcal{T}$ necessarily produces a discontinuous predictive surface, the aggregated mean tends to approximate continuous transitions between regions quite well, and uncertainty in the posterior for $\mathcal{T}$ translates into higher posterior predictive uncertainty near region boundaries. The results in Section 2.5 provide illustration.

For cases where the data possibly indicates a non-smooth process, as in the transition between subsonic and supersonic speeds in the NASA LGBB data [Chapter 1, and Section 4.4.3], the treed GP retains the flexibility necessary to model discontinuities, in the posterior predictive surface.

2.4 Implementation

The treed GP model is coded in a mixture of `C` and `C++`: `C++` for the tree data structure ($\mathcal{T}$) and `C` for the GP at each leaf of $\mathcal{T}$. The `C` code can interface with either standard platform-specific `Fortran BLAS/Lapack` libraries for the linear algebra necessary to estimate the parameters of the GP, or link to those automatically configured for fast execution on a variety of platforms via the `ATLAS` library (Whaley & Petitet, 2004). In most cases, the `ATLAS` implementation is significantly faster than standard `BLAS/Lapack`. The code has been tested on Unix (`Solaris, Linux, FreeBSD, OSX`) and Windows (2000, XP) platforms.

It is useful to first translate and re-scale the input data ($\mathbf{X}$) so that it lies in an $\Re^{m_X}$ dimensional unit cube. Doing this makes it easier to construct prior distributions for the width parameters to the correlation function $K(\cdot,\cdot)$ in particular. Many implementation details regarding the tree $\mathcal{T}$ have already been outlined in Section 2.2.2. Conditioning on $\mathcal{T}$, proposals for all parameters which require MH sampling are taken from a uniform "sliding window" centered around the location of the last accepted setting. For example, a proposed a new nugget parameter g_ν to the correlation function $K(\cdot,\cdot)$ in region r_ν would go as

$$g_\nu^* \sim \text{Unif}\left(\frac{3}{4}g_\nu, \frac{4}{3}g_\nu\right).$$

Calculating the forward and backwards proposal probabilities for the MH acceptance ratio is straightforward.

After conditioning on the tree and parameters ($\{\mathcal{T},\boldsymbol{\theta}\}$), prediction can be parallelized by using a producer/consumer model. This allowed the use of `PThreads` in order to take advantage of multiple processors, and get speed-ups of at least a factor of two. This is particularly relevant since dual processor workstations and multi-processor servers are becoming commonplace in modern research labs. Parallel sampling of the posterior of $\boldsymbol{\theta}|\mathcal{T}$ for each of the $\{\theta_\nu\}_{\nu=1}^R$ is also possible. However, the speed-up in this second case is less impressive. To ice the cake, the whole thing is wrapped up in an intuitive `R` interface (R Development Core Team, 2004). Compared to existing methods, this approach lead to an extremely fast implementation of nonstationary GPs.

2.5 Illustration & Experimentation

In this section the treed GP model is illustrated on two synthetic data sets, and one set of real world data. Further experimentation is deferred until the next chapter, in Section 3.3, after a more mature semiparametric nonstationary regression model is developed. To keep things simple, for now, the isotropic power family (1.10) correlation function ($p_0 = 2$) is chosen for $K^*(\cdot, \cdot|d)$ in the following experiments, with range parameter d, combined with nugget g to form $K(\cdot, \cdot|d, g)$. (More about correlation functions in the next chapter.)

2.5.1 1-d Synthetic Sinusoidal data

Consider 1-dimensional simulated data on the input space $[0, 20]$. The true response comes partly from Higdon et al. (2002), augmented to include a linear region. Eq. (2.20) gives a formula describing the data, and a picture is shown in Figure 2.3. As is obvious from the figure, this dataset typifies the type of nonstationary response surface that the treed GP model was designed to exploit. Zero mean Gaussian noise with sd = 0.1 is added to the response to keep things interesting.

$$z(x) = \begin{cases} \sin\left(\frac{\pi x}{5}\right) + \frac{1}{5}\cos\left(\frac{4\pi x}{5}\right) & x < 10 \\ x/10 - 1 & \text{otherwise} \end{cases} \tag{2.20}$$

Figure 2.4 shows the posterior predictive surfaces of three regression models for comparison based on samples obtained at $N = 200$ evenly-spaced input locations—mean in solid black, and 95% intervals in dashed-red. The *top* panel is from a Bayesian Linear CART model (Chipman et al., 2002), which does well in the linear region, but comes up short in the sinusoidal region. The *middle* panel is from a stationary GP model which is heavily influenced by the sinusoidal region, and consequently fits it well, but is unable to model the more smooth

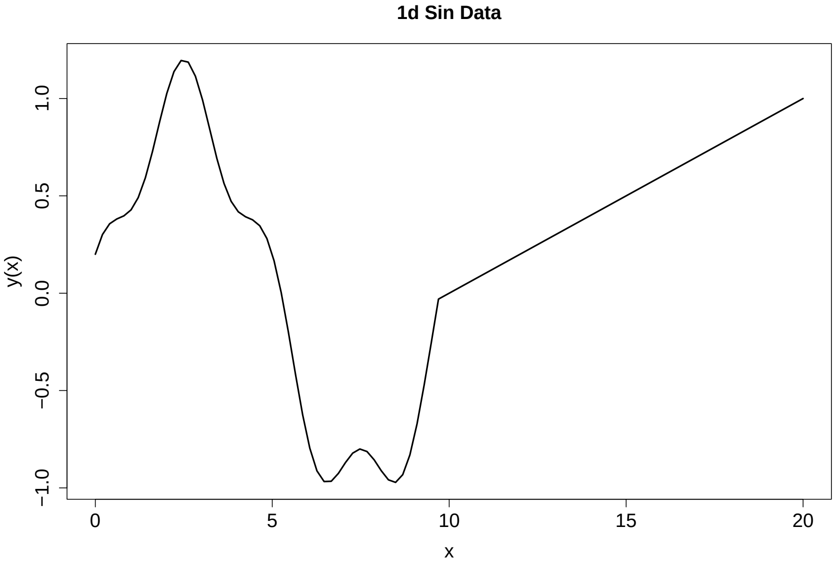


Figure 2.3: Sinusoidal data

linear process. This is because nonstationarity in the data cannot be captured by a stationary (or homogeneous) correlation structure. The *bottom* panel shows the best of both worlds: a treed GP, which fits a sinusoidal, lower correlation, GP in the sinusoidal region, and smooth, higher correlation, GP in the linear region.

2.5.2 2-d Synthetic Exponential data

Next, results are shown for a two-dimensional input space in $[-2, 6] \times [-2, 6]$. The true response is given by

$$z(\mathbf{x}) = x_1 \exp(-x_1^2 - x_2^2). \tag{2.21}$$

A small amount of Gaussian noise (with sd = 0.001) is added. Besides its dimensionality, a key difference between this data set and the last one is that it is not defined using step functions; this smooth function does not have any artificial breaks between regions.

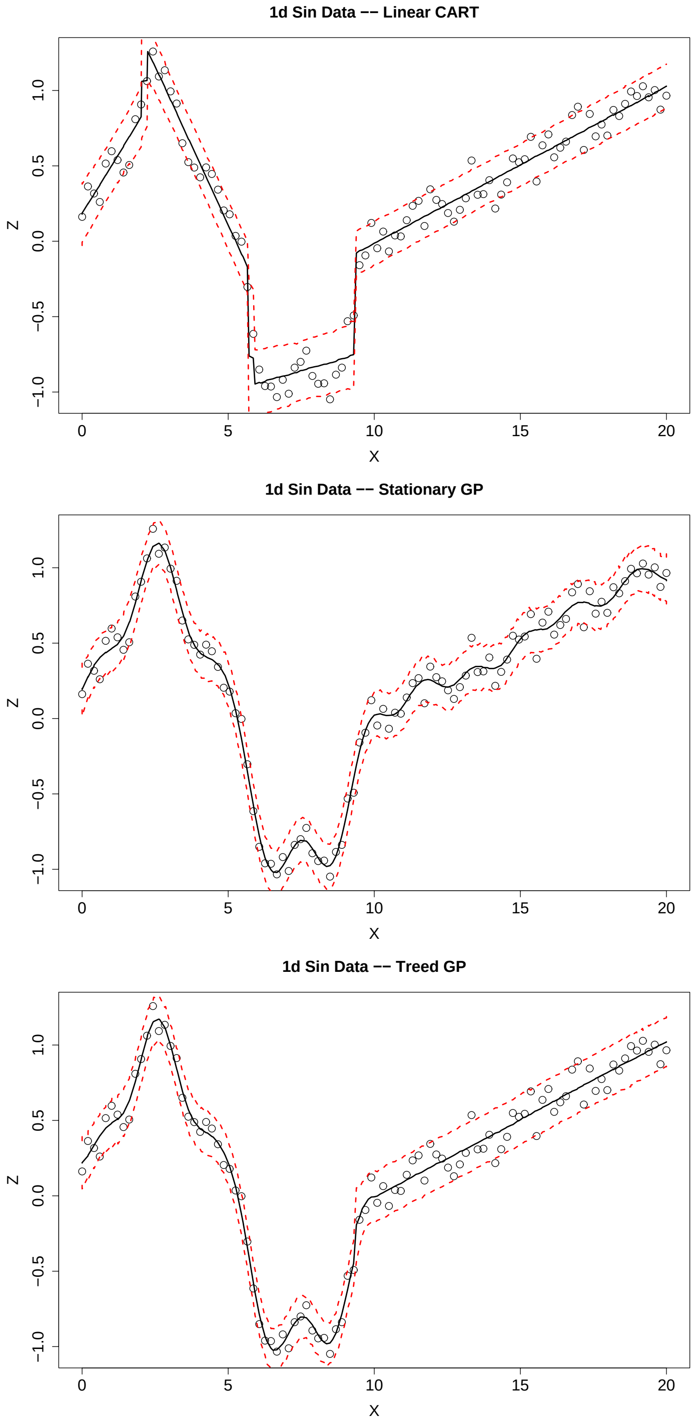
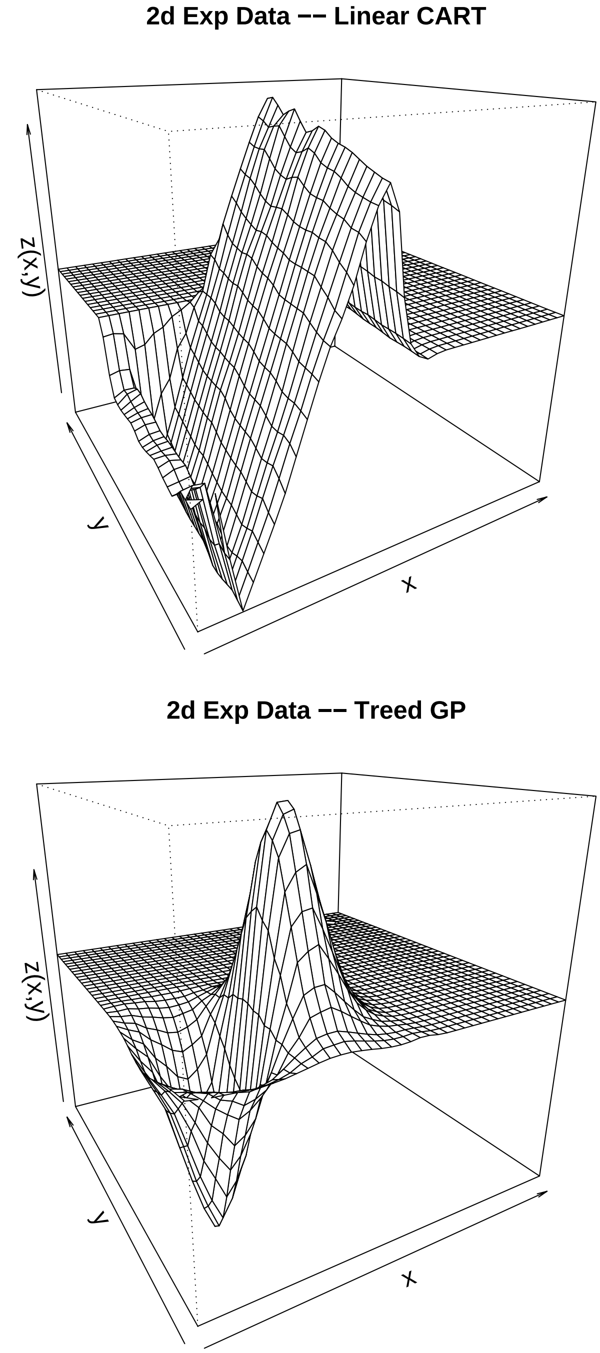


Figure 2.4: Comparison between Bayesian linear CART (*top*), stationary GP (*middle*) and the treed GP model (*bottom*), for the 1-d Sine data.



2d Exp Data −− Linear CART

2d Exp Data −− Treed GP

Figure 2.5: Comparison between Bayesian linear CART (*top*), and the treed GP model *bottom*, for the 2-d Exponential data.

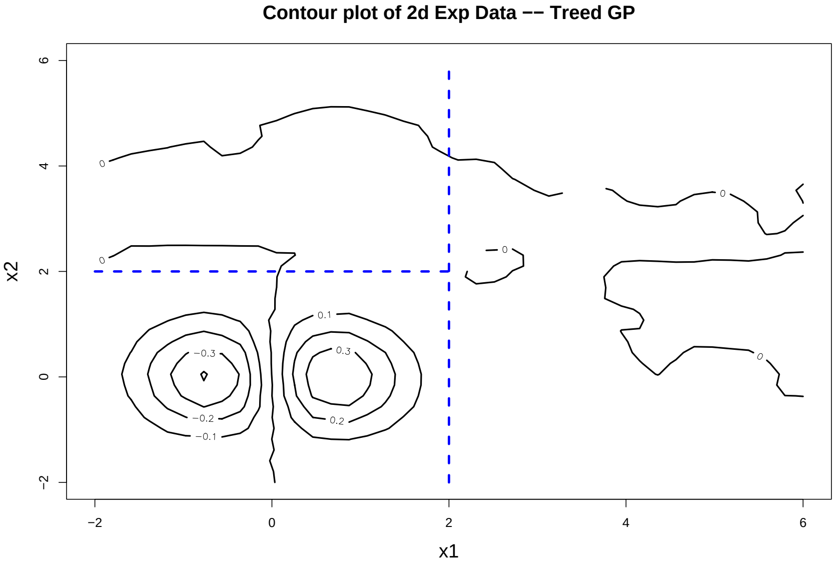


Figure 2.6: Contour plot showing the mean surface and representative partitions found using the treed GP model.

Figure 2.5 shows plots comparing fits of Bayesian Linear CART (*top*) and the treed GP (*bottom*). It is clear from the figure that the treed GP is better. The fit for a stationary GP is not shown because it looks very similar to that of the treed GP. The data are indeed stationary. Still, the tree GP finds an average of three partitions, as shown in Figure 2.6. Much of the advantage of the treed GP in this situation, over a single stationary GP, is in speed of computation. Inverting three matrices, one of half and two of one quarter of the original size (N), is considerably faster than inverting a single $N \times N$ matrix.

2.5.3 Motorcycle data

The Motorcycle Accident Dataset (Silverman, 1985) is a classic nonstationary data set used in recent literature (Rasmussen & Ghahramani, 2002) to demonstrate the success of nonstationary models. The data set consists of measurements of acceleration of the head of a motorcycle rider as a function of time in the first moments after an impact. In addition to being nonstationary, the data has input-dependent noise, which makes it useful for illustrating how the treed GP model handles this nuance. There are at least two, and perhaps three regions where the response exhibits different behavior both in terms of the correlation structure and noise level.

Figure 2.7 shows the data, and the fit given by the treed GP model. The *top* panel shows the estimate of the surface with 90%-quantile error bars; the *bottom* panel shows the difference in quantiles. Vertical lines on both panels illustrate a typical treed partition $\mathcal{T}$. The error bars, and estimated error spread, can give insight into the uncertainty in the posterior distribution for $\mathcal{T}$. Notice the sharp rise in estimated variance from the leftmost region to the center region. Contrast this with the more gradual, stepwise, descent in variance from the center region to the rightmost region. There was far more certainty in the posterior for the left split than the right one. The average number of partitions in the posterior for $\mathcal{T}$ over $20,000$ rounds ($5,000$ burn in) was 3.111. The occasional extra partition usually "popped up" to help smooth the boundary between the center and rightmost region. Rather than a single partition near $x \approx 40$, the two partitions arise near $x \approx 36$ and $x \approx 42$. Less often the tree would prune back splits in the right-hand part of the domain, leaving only two partitions: a leftmost one, and a single right-hand region.

These results are quite different from those reported by Rasmussen & Ghahramani (2002). In particular, the error-bars they report for the leftmost region seem too large relative to the center and rightmost regions. They use a what they call an "infinite" mixture of GP "experts" which is really a Dirichet process mixture of GPs. They report that the posterior distribution uses between 3 and 10 experts to fit this data, which they admit has "roughly" three regions. In fact, in their histogram of the number of GP experts used throughout the

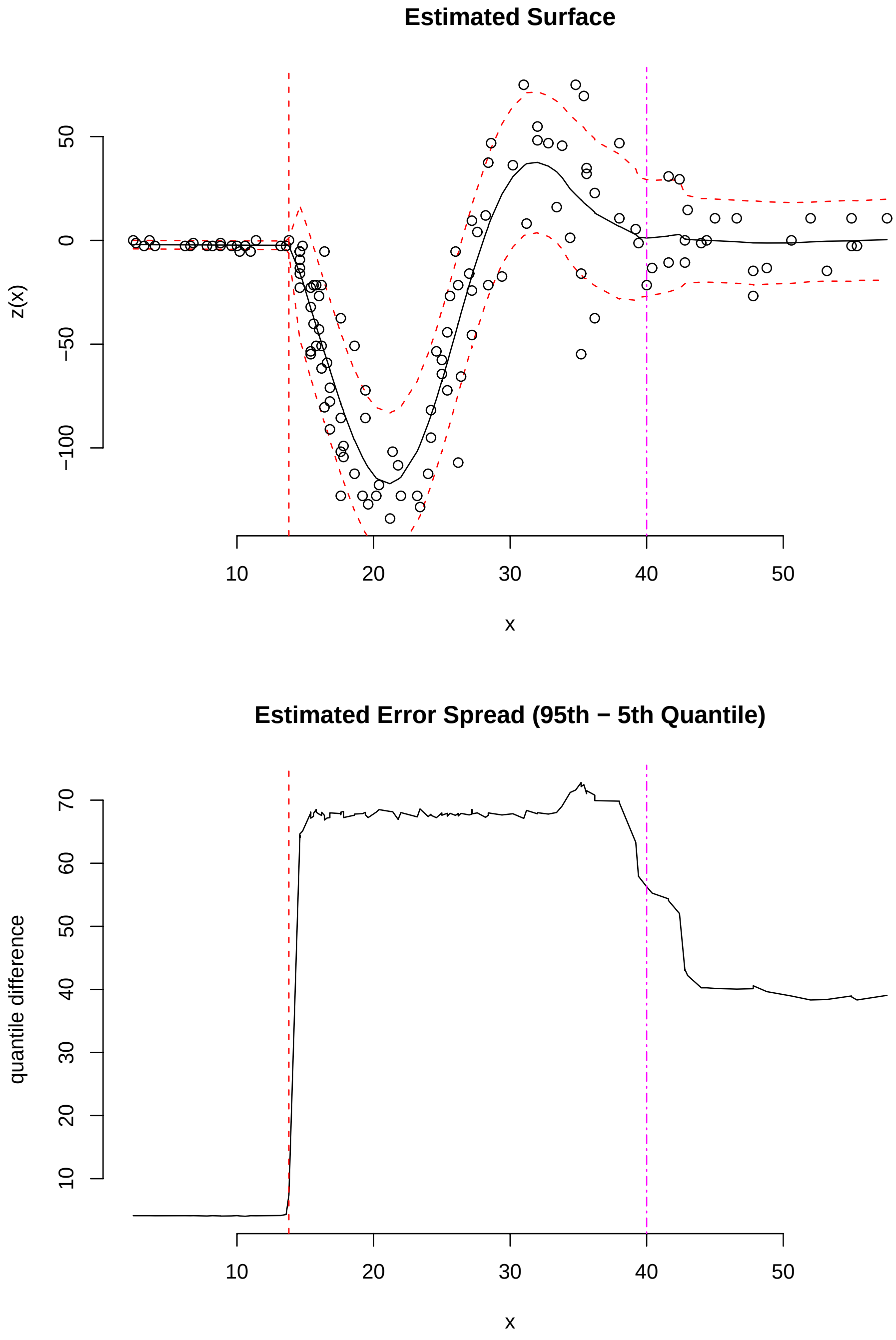


Figure 2.7: 1-d Motorcycle Dataset, fit by our nonstationary model.

MCMC rounds, they show that between 3 and 10 experts are equally likely, and even 10-15 experts still have considerable posterior mass. Contrast this with the treed GP model which almost always partitions into three regions, occasionally four, rarely two.

On speed grounds, the treed GP is also a winner. Rasmussen & Ghahramani (2002) report that they ran mixture of GP experts model using a total of 11,000 MCMC rounds, discarding the first 1,000 and keeping every 100$^{\text{th}}$ after that. This took roughly one hour on a 1 GHz Pentium. Allowing treed GP to use 25,000 MCMC rounds, discarding the first 5,000 and keeping every sample thereafter takes less than ~ 3 minutes on on a 1.8 GHz Athalon.

2.6 Conclusion

In this chapter the treed Gaussian Process model was introduced as a nonparametric extension of Bayesian Linear CART model, and validated as a nonstationary regression tool on synthetic and real data. A fully Bayesian treatment of the treed GP model was laid out, treating the hierarchical parameterization of a correlation function $K(\cdot,\cdot)$ as a black box. The next chapter is dedicated to the study of the prior specification for parameters to the correlation function $K(\cdot,\cdot)$ for the separable and isotropic power families, motivating and developing the GP LLM model. Chapter 4 takes advantage of the nonstationary nature of the measures of predictive error provided by the treed GP (or GP LLM) model in order to design experiments.

Chapter 3

Gaussian Processes and Limiting Linear Models

Gaussian processes (GPs) retain the linear model (LM) either as a special case, or in the limit. This chapter shows how the limiting parameterization can be exploited when the data are at least partially linear. However, from the prospective of the Bayesian posterior, the GPs which encode the LM either have probability of nearly zero or are otherwise unattainable without the explicit construction of a prior with the limiting linear model (LLM) in mind.

In this chapter, a sensitivity analysis on the prior specification for the parameters to the correlation function $K(\cdot, \cdot)$ is carried out. An appropriate prior is developed, yielding practical benefits which extend well beyond the computational and conceptual simplicity of the LM. For example, linearity can be extracted on a per-dimension basis, or can be combined with treed partition models to yield a highly efficient nonstationary model. The resulting (treed) GP LLM model is demonstrated and validated on synthetic and real datasets of varying linearity and dimensionality. Comparisons are made to other approaches in the literature.

Correlation function and notation disclaimer

The correlation function and its parameters are the focus of this Chapter, so a parameterization needs to be chosen. To remain consistent with the Design and Analysis of Computer Experiments (DACE) literature I choose to work with the power family, with power $p_0 = 2$ (see Section 1.2.2), and nugget g (1.8): $K(\mathbf{x}_j, \mathbf{x}_k|g) = K^*(\mathbf{x}_j, \mathbf{x}_k) + g\delta_{j,k}$. Recall that the isotropic correlation function (1.10) is parameterized with a single range parameter, d: $K^*(\mathbf{x}_j, \mathbf{x}_k|d) = \exp\{-||\mathbf{x}_j - \mathbf{x}_k||^2/d\}$ and that the separable function (1.11) has m_X range parameters $\mathbf{d} = \{d_1, \ldots, d_{m_X}\}$: $K^*(\mathbf{x}_j, \mathbf{x}_k|\mathbf{d}) = \exp\{-\sum_{i=1}^{m_X} |x_{ij} - x_{ik}|^2/d_i\}$. The following discussion is generic enough to easily extend to other families of correlation functions.

When the discussion applies to both separable and isotropic versions, I shall use d and $\mathbf{d}$ interchangeably, noting that the isotropic version is a special case of the separable one. Notice the absence of region-specific subscripts (ν) in the above equations, as the discussion applies generally to any GP. However, when coupled with treed partitioning, it may be possible to treat formerly non-linear data as piecewise linear and gain a great advantage. In fact this was the motivation for the limiting linear model of the GP, and will be exploited later.

A possible first approach to extending the work in this Chapter would be to treat the power p_0, which governs the smoothness of the underlying process, as random.

3.1 Limiting Linear Models

A special limiting case of the Gaussian process model is the standard linear model. Replacing the top (likelihood) line in the hierarchical model given in Eq. (2.1)

$$\mathbf{Z}|\boldsymbol{\beta}, \sigma^2, \mathbf{K} \sim N_N(\mathbf{F}\boldsymbol{\beta}, \sigma^2\mathbf{K}) \qquad \text{with} \qquad \mathbf{Z}|\boldsymbol{\beta}, \sigma^2 \sim N_N(\mathbf{F}\boldsymbol{\beta}, \sigma^2\mathbf{I}),$$

where $\mathbf{I}$ is the $N \times N$ identity matrix, gives a parameterization of a linear model. From a phenomenological perspective, GP regression is more flexible than standard linear regression in that it can capture nonlinearities in the interaction between covariates ($\mathbf{x}$) and responses (z). From a modeling perspective, the GP can be more than just overkill for linear data. Parsimony and over-fitting considerations are the tip of the iceberg. It is also unnecessarily computationally expensive, as well as numerically unstable. Specifically, it requires the inversion of a large covariance matrix—an operation whose computing cost grows with the cube of the sample size. Moreover, large finite $\mathbf{d}$ parameters can be problematic from a numerical perspective. Unless g is also large, the resulting covariance matrix can be numerically singular when the off-diagonal elements of $\mathbf{K}$ are nearly one.

It is common practice to scale the inputs ($\mathbf{X}$) either to lie in the unit cube, or to have a mean of zero and a range of one. As will be shown in Section 3.1.1, scaled data and mostly linear predictive surfaces can result in almost singular covariance matrices even when the range parameter is relatively small ($2 < d \ll \infty$). So for some parameterizations, the GP is operationally equivalent to the limiting linear model (LLM), but comes with none of its benefits, e.g., speed and stability. This chapter will show how exploiting and/or manipulating such equivalence can be of great practical benefit. As Bayesians, this means constructing a prior distribution on $\mathbf{K}$ that makes it clear in which situations each model is preferred; i.e., when should $\mathbf{K} \to c\mathbf{I}$? The key idea is to specify a prior on a "jumping" criterion between the GP and its LLM, thus setting up a Bayesian model selection/averaging framework.

Theoretically, there are only two parameterizations to a GP correlation structure $K(\cdot, \cdot)$ which encode the LLM. Though they are well-known, without intervention they are quite unhelpful from the perspective of *practical* estimation and inference. The first one is when the range parameter d is set to zero. In this case $\mathbf{K} = (1 + g)\mathbf{I}$, and the result is clearly a linear model. The other parameterization may be less obvious.

Cressie (1991) [in Section 3.2.1] analyzes the “effect of variogram parameters on kriging” paying special attention to the nugget (g) and its interaction with the range parameter (d). He remarks that the larger the nugget the more the kriging interpolator smoothes and in the limit predicts with the linear mean. However, perhaps more relevant to the forthcoming discussion is his later remarks on the interplay between the range and nugget parameter in determining the kriging neighborhood. Specifically, a large nugget coupled with a large range drives the interpolator towards the linear mean. This is refreshing since constructing a prior for the LLM by exploiting the former GP parameterization (range $d \rightarrow 0$) is difficult, and for the latter (nugget $g \rightarrow \infty$) near impossible. Cressie hints that an (essentially) linear model may be attainable with nonzero d and finite g.

3.1.1 Exploratory analysis

Before constructing a prior, it makes sense to study the kriging neighborhood and look for a platform from which to “jump” to the LLM. The following exploratory analysis focuses on studying likelihoods and posteriors for GPs fit to data generated from the linear model

$$z_i = 1 + 2x_i + \epsilon_, \quad \text{where} \quad \epsilon_i \overset{\text{iid}}{\sim} N(0,1) \tag{3.1}$$

using $n = 10$ evenly spaced x-values in the range $[0,1]$.

GP likelihoods on linear data

Figure 3.1 shows two interesting samples from (3.1). Also plotted is the generating line (dot-dashed), the maximum likelihood (ML) linear model ($\hat{\boldsymbol{\beta}}$) line (dashed), the mean predictive mean surface of the ML GP, maximized over d and g and $[\sigma^2|d,g]$ (solid), and its

95% errorbars (dotted). The ML values of d and g are also indicated in each plot. The GP likelihoods were evaluated for ML estimates of the regression coefficients $\hat{\boldsymbol{\beta}}$. Conditioning on g and d, the ML variance was computed by solving

$$0 \equiv \frac{d}{d\sigma^2} \log N(\mathbf{Z}|\mathbf{F}\hat{\boldsymbol{\beta}}, \sigma^2\mathbf{K}) = -\frac{n}{\sigma^2} + \frac{(\mathbf{Z}-\mathbf{F}\hat{\boldsymbol{\beta}})^\top \mathbf{K}^{-1}(\mathbf{Z}-\mathbf{F}\hat{\boldsymbol{\beta}})}{(\sigma^2)^2}.$$

This gave an MLE with the form $\hat{\sigma}^2 = (\mathbf{Y} - \mathbf{F}\hat{\boldsymbol{\beta}})^\top \mathbf{K}^{-1}(\mathbf{Z}-\mathbf{F}\hat{\boldsymbol{\beta}})/n$. For the linear model the likelihood was evaluated as $P(\mathbf{Y}) = N_{10}(\mathbf{F}\hat{\boldsymbol{\beta}}, \hat{\sigma}^2\mathbf{I})$, and for the GP as

$$P(\mathbf{Z}|d,g) = N_{10}\left[\mathbf{F}\hat{\boldsymbol{\beta}}, \hat{\sigma}^2\mathbf{K}_{\{d,g\}}\right],$$

where $\mathbf{F} = (\mathbf{1}, \mathbf{X})$ and $\mathbf{K}_{\{d,g\}}$ is the covariance matrix generated using $K(\cdot,\cdot) = K^*(\cdot,\cdot|d) + g$ for $K^*(\cdot,\cdot|d)$ from the isotropic power family with range parameter d.

Both samples and fits plotted in Figure 3.1 have linear looking predictive surfaces, but only for the one in the *top* row does the linear model have the maximum likelihood. Though the predictive surface in the *bottom-left* panel could be mistaken as “linear”, it was indeed generated from a GP with large range parameter ($d = 2$) and modest nugget setting (g) as this parameterization had higher likelihood than the linear model. The *right* column of Figure 3.1 shows likelihood surfaces corresponding to the samples in the *left* column. Also shown is likelihood value of the MLE $\hat{\boldsymbol{\beta}}$ of the linear model (solid horizontal line). The likelihood surfaces for each sample look drastically different. In the top sample the LLM ($d = 0$) uniformly dominates all other GP parameterizations. Contrast this with the likelihood of the second sample. There, the resulting predictive surface looks linear, but the likelihood of the LLM is comparatively low.

Illustrating the other limiting linear model parameterization ($g \to \infty$), Figure 3.2

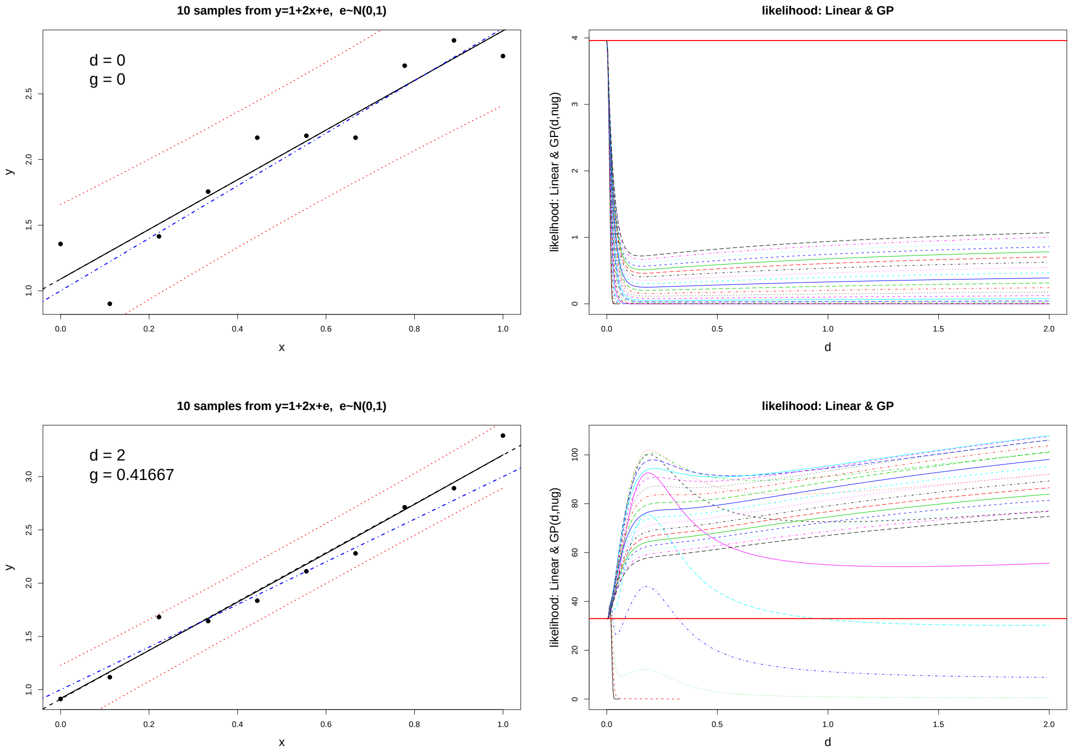


Figure 3.1: Two simulations (*rows*) from $y_i = 1 + 2x_i + \epsilon_i,\ \epsilon_i \sim N(0,1)$. *Left* column shows GP fit (solid) with 95% errorbars (dotted), maximum likelihood $\hat{\boldsymbol{\beta}}$ (dashed), and generating linear model ($\boldsymbol{\beta} = (1,2)$) (dot-dashed). *Right* column shows GP(d,g) likelihood surfaces. The (maximum) likelihood ($\hat{\boldsymbol{\beta}}$) of the linear model is indicated by the solid horizontal line.

shows how as the nugget g increases, likelihood of the GP approaches that of the linear model. The range parameter was set at $d = 1$. The x-axis of nugget values is plotted on a log scale. The nugget must be quite large relative to the actual variability in the data be before the likelihoods of the GP and LLM become comparable. A sample of size $n = 100$ from (3.1) was used.

Most simulations from (3.1) gave predictive surfaces like the *upper left*-hand side of Figure 3.1 and corresponding likelihoods like the *upper-right*. But this is not always the case. Occasionally a simulation would give high likelihood to GP parameterizations if the sample was

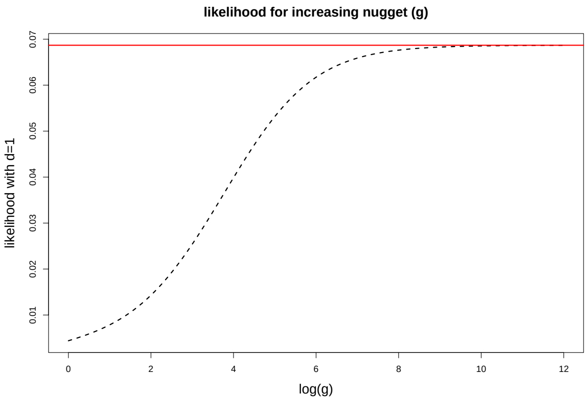


Figure 3.2: Likelihoods as the nugget gets large for an $n = 100$ sample from Eq. (3.1). The x-axis is $(\log g)$, the range is fixed at $d = 1$; the likelihood of the LLM $(d = 0)$ is shown for comparison.

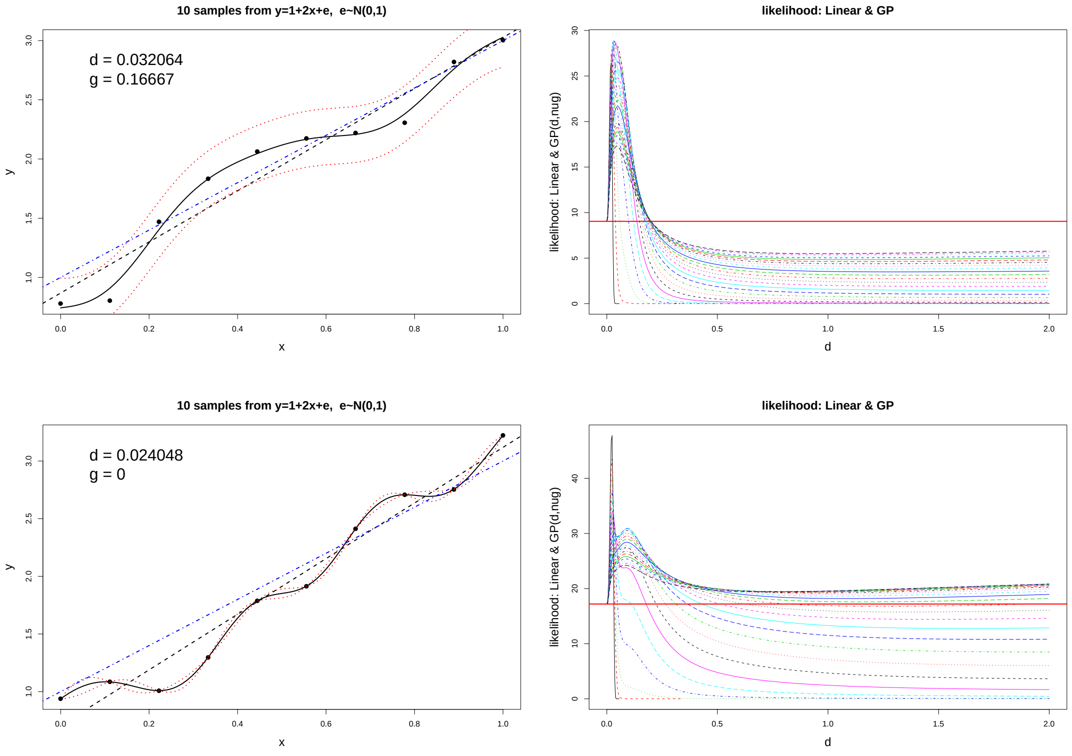
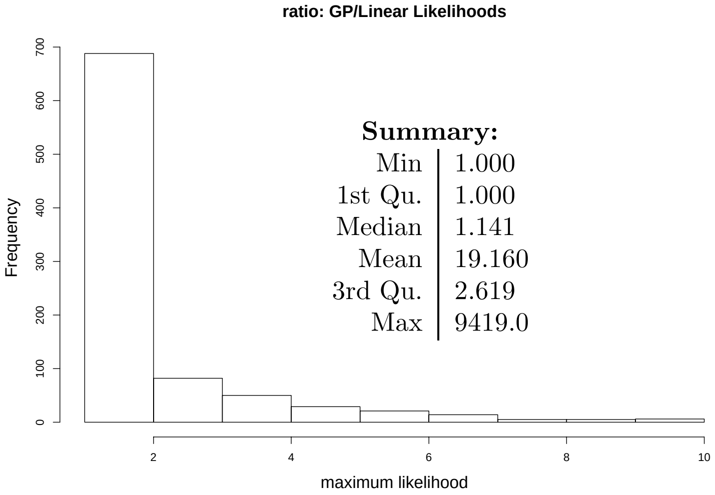


Figure 3.3: GP(d, g) fits (*left*) and likelihood surfaces (*right*) for two of samples for the linear model (3.1).



Summary:

Min	1.000
1st Qu.	1.000
Median	1.141
Mean	19.160
3rd Qu.	2.619
Max	9419.0

Figure 3.4: Histograms of the ratio of the maximum likelihood GP parameterization over the likelihood of the limiting linear model. Only the smaller 90% of the ratios are included in the histogram. Full summary statistics for the ratio are also shown.

smoothly waving. This is not uncommon for small sample sizes such as $n = 10$—for example, consider those shown in Figure 3.3. Waviness becomes less likely as the sample size n grows.

Figure 3.4 summarizes the ratio of the ML GP parameterization over the ML linear model based on 1000 simulations of ten evenly spaced random draws from (3.1). A likelihood ratio of one means that the LLM was best for a particular sample. The 90%-quantile histogram and summary statistics in Figure 3.4 show that the GP is seldom much better than the linear model. For some samples the ratio can be really large (> 9000) in favor of the GP, but more than two-thirds of the ratios are close to one—approximately 1/3 (362) were exactly one but 2/3 (616) had ratios less than 1.5. What this means is that posterior inference for borderline linear data is likely to depend heavily the prior specification of $K(\cdot, \cdot)$.

For some of the smaller nugget values, in particular $g = 0$, and larger range settings d, some of the likelihoods for the GP could not be computed because the imputed covariance matrices were numerically singular, and could not be inverted. This illustrates a phenomenon noted by Neal (1997) who advocates that a non-zero nugget (or *jitter*) should be included in the model, if for no other reason, than to increase numerical stability. Numerical instabilities may also be avoided by allowing $p_0 < 2$, or by using the Matérn family of correlation functions [see Section 1.2.2]. This phenomenon reappears when examining the posterior of the GP and LLM.

GP posteriors on linear data

Suppose that rather than examining the multivariate-normal likelihoods of the linear and GP model, using the ML mean $\hat{\boldsymbol{\beta}}$ and variance $\hat{\sigma}^2$ values, the marginalized posterior $p(\mathbf{K}|\mathbf{Z}, \boldsymbol{\beta}_0, \tau^2, \mathbf{W})$ of Eq. (2.14) was used, which integrates out $\boldsymbol{\beta}$ and σ^2. Using (2.14) requires specification of the prior $p(\mathbf{K})$, which for the power family means specifying $p(d, g)$. Alternatively, one could consider dropping the $p(d, g)$ term from (2.14) and look solely at the marginalized likelihood. However, in light of the arguments above, there is reason to believe that the prior specification might carry significant weight.

If it is suspected that the data might be linear this bias should be encoded in the prior somehow. This is a non-trivial task given the nature of the GP parameterizations which encode the LLM. Pushing d towards zero is problematic because small non-zero d causes the predictive surface to be quite wiggly—certainly far from linear. Deciding how small the range parameter (d) should be before treating it as zero—as in Stochastic Search Variable Selection (SSVS) (George & McCulloch, 1994), or Chapter 12 in (Gilks et al., 1996)—while still allowing a GP to fit truly non-linear data is no simple task. The large nugget approach is also out of the question because putting increasing prior density on a parameter as it gets large is impossible. Rescaling the responses might work, but constructing the prior would be nontrivial, and moreover, such an approach would preclude its use in many applications, particularly for adaptive sampling or sequential design of experiments when one hopes to learn about the range of responses, and/or search for extrema.

However, for a continuum of large d values (say $d > 0.5$ on the unit interval) the predictive surface is practically linear. Consider a mixture of gammas prior for d:

$$\begin{aligned} p(d,g) &= p(d) \times p(g) \\ &= p(g) \times \frac{1}{2}[G(d|\alpha = 1, \beta = 20) + G(d|\alpha = 10, \beta = 10)]. \end{aligned} \tag{3.2}$$

It gives roughly equal mass to small d representing a population of GP parameterizations for wavy surfaces, and a separate population for those which are quite smooth or approximately linear. Figure 3.5 depicts $p(d)$ via histogram, ignoring $p(g)$ which is usually taken to be a simple exponential distribution. Alternatively, one could encode the prior as $p(d,g) = p(d|g)p(g)$ and then use a reference prior (Berger et al., 2001) for $p(d|g)$. I chose the more deliberate, independent, specification in order to encode my prior belief that there are essentially two kinds of processes: wavy and smooth.

Evaluation of the marginalized posterior (2.14) requires settings for the prior mean coefficients $\boldsymbol{\beta}_0$, covariance $\tau^2\mathbf{W}$, and hierarchical specifications $(\alpha_\sigma, \gamma_\sigma)$ for σ^2. For now, these parameter settings are fixed to those which were known to generate the data.

Figure 3.6 shows three samples from the linear model (3.1) along with likelihood and posterior surfaces. Some of the likelihood and posterior lines suddenly stop due to a numerically unstable parameterization (Neal, 1997). The GP fits shown in the first column of the figure are based on the maximum *a posteriori* (MAP) estimates of d and g. The posteriors in the third column clearly show the influence of the prior. Nevertheless, the posterior density for large d-

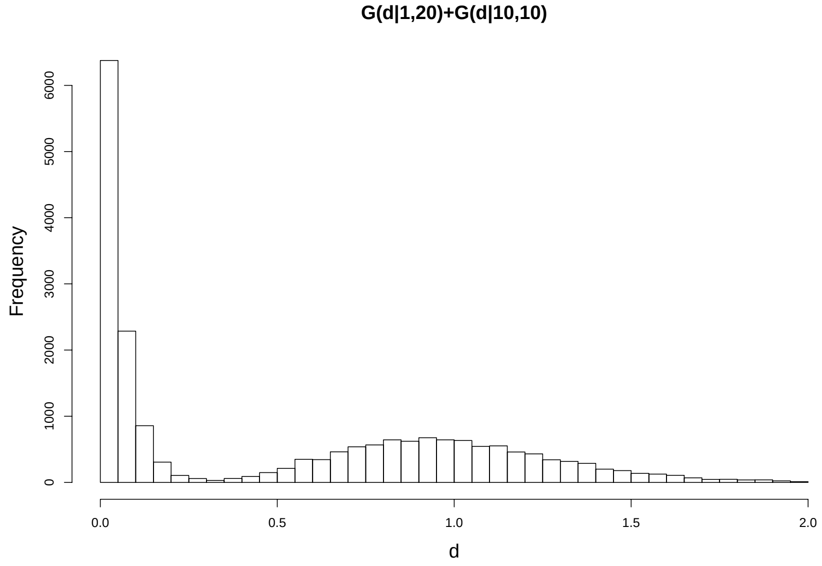


Figure 3.5: Histogram of the mixture of gammas prior $p(d)$ as given in Eq. (3.2).

values is dis-proportionately high relative to the prior. For all but the sample in the first column, large d-values represent at least 90% of the cumulative posterior distribution. Samples from these posteriors would yield mostly linear predictive surfaces. The last sample is particularly interesting as well as being the most representative across all samples. Here, the LLM ($d = 0$) is the MAP GP parameterization, and uniformly dominates all other parameterizations in posterior density. Still, the cumulative posterior density favors large d-values thus favoring linear "looking" predictive surfaces over the actual (limiting) linear parameterization.

Figure 3.7 (*top*) shows a representative MAP GP fit for a sample of size $n = 100$ from (3.1). Since larger samples have a lower probability of coming out wavy, the likelihood of the LLM is much higher than other GP parameterizations. However, the likelihood around $d = 0$, shown in the *middle* panel, is severely peaked. Small, nonzero, d-values have extremely low likelihood. The posterior in the *bottom* panel has high posterior density on large d values.

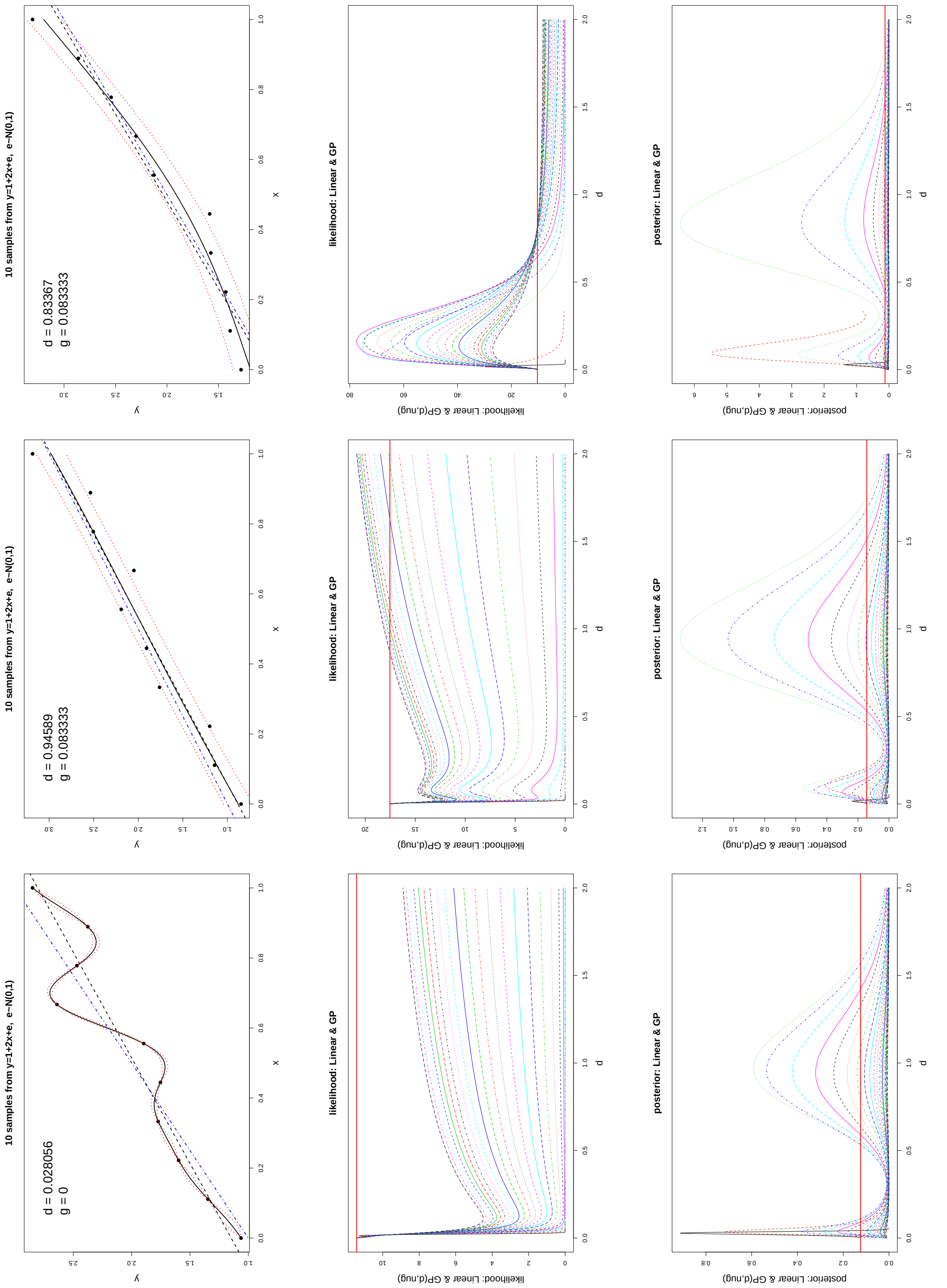


Figure 3.6: *Top row* (when rotated) shows the GP(d, g) fits; *Middle row* shows likelihoods and *bottom row* shows the integrated posterior distribution for range (d, x-axis) and nugget (g, lines) settings for three samples, one per each column.

All other GP parameterizations have low posterior probability relative to that of the LLM (horizontal solid line). The MAP predictive surface (*top* panel) has a very small, but noticeable, amount of curvature.

Ideally, linear looking predictive surfaces should not have to bear the computational burden implied by full-fledged GPs. But since the LLM ($d = 0$) is a point-mass (which is the only parameterization that actually gives an identity covariance matrix), it has zero probability under the posterior. It would never be sampled in an MCMC, even when it is the MAP estimate. Section 3.2 develops a prior on the range parameter (d) so that there is high posterior probability of "jumping" to the LLM whenever d is large. The goal is to do this without actually proposing $d = 0$.

GP posteriors and likelihoods on non-linear data

For completeness, Figures 3.8 and 3.9 show fits, likelihoods, and posteriors on non-linear data. The first column of Figure 3.8 (when rotated) corresponds to a linear sample, and each successive column corresponds to a sample which is increasingly less linear, ranging from low degree polynomials to mixtures of exponentials and mixtures of trigonometric functions in the last column of Figure 3.9. Each sample is of size $n = 50$. The shape of the prior looses its influence as the data becomes more non-linear. As the samples become less linear the d-axis (x-axis) shrinks in order to focus in on the mode. Though in all six cases the MLEs do not correspond to the MAP estimates, the corresponding ML and MAP predictive surfaces look remarkably similar (not shown). This is probably due to the fact that the posterior integrates out $\boldsymbol{\beta}$ and σ^2, whereas the likelihoods were computed with point estimates of these parameters.

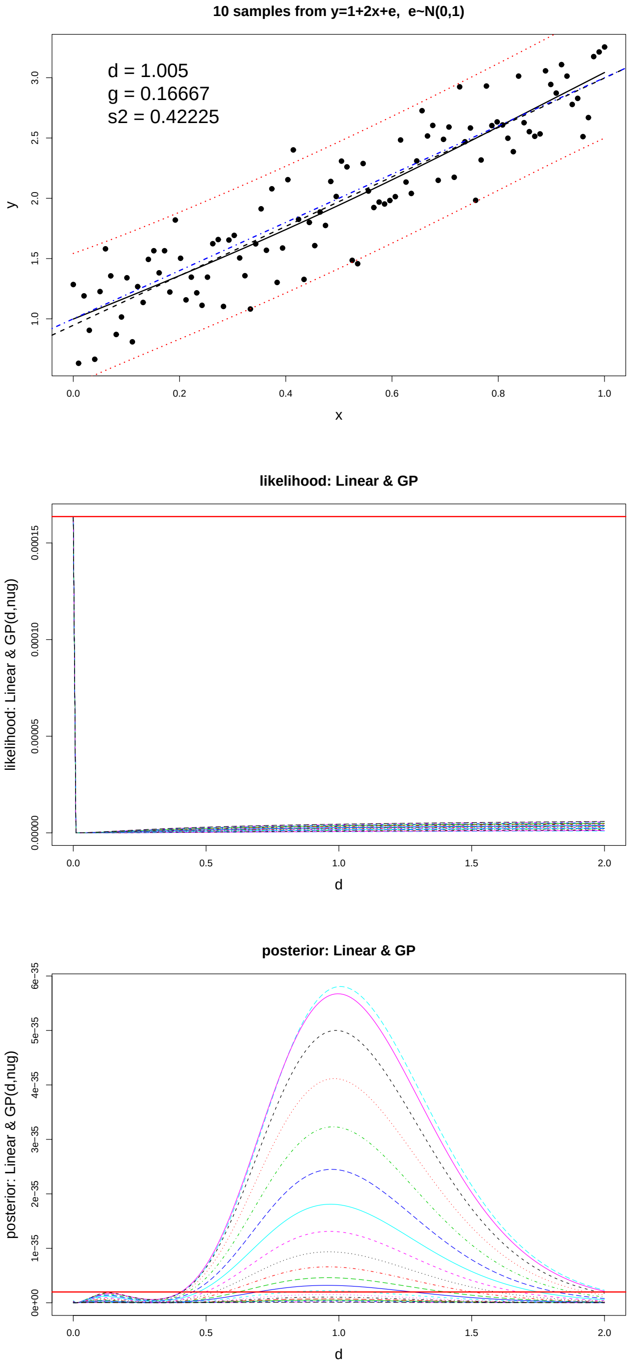


Figure 3.7: *Top* shows the GP(d, g) fit with a sample of size $n = 100$; *middle* shows the likelihood and *bottom* shows the integrated posterior distribution for range (d, x-axis) and nugget (g, lines) settings.

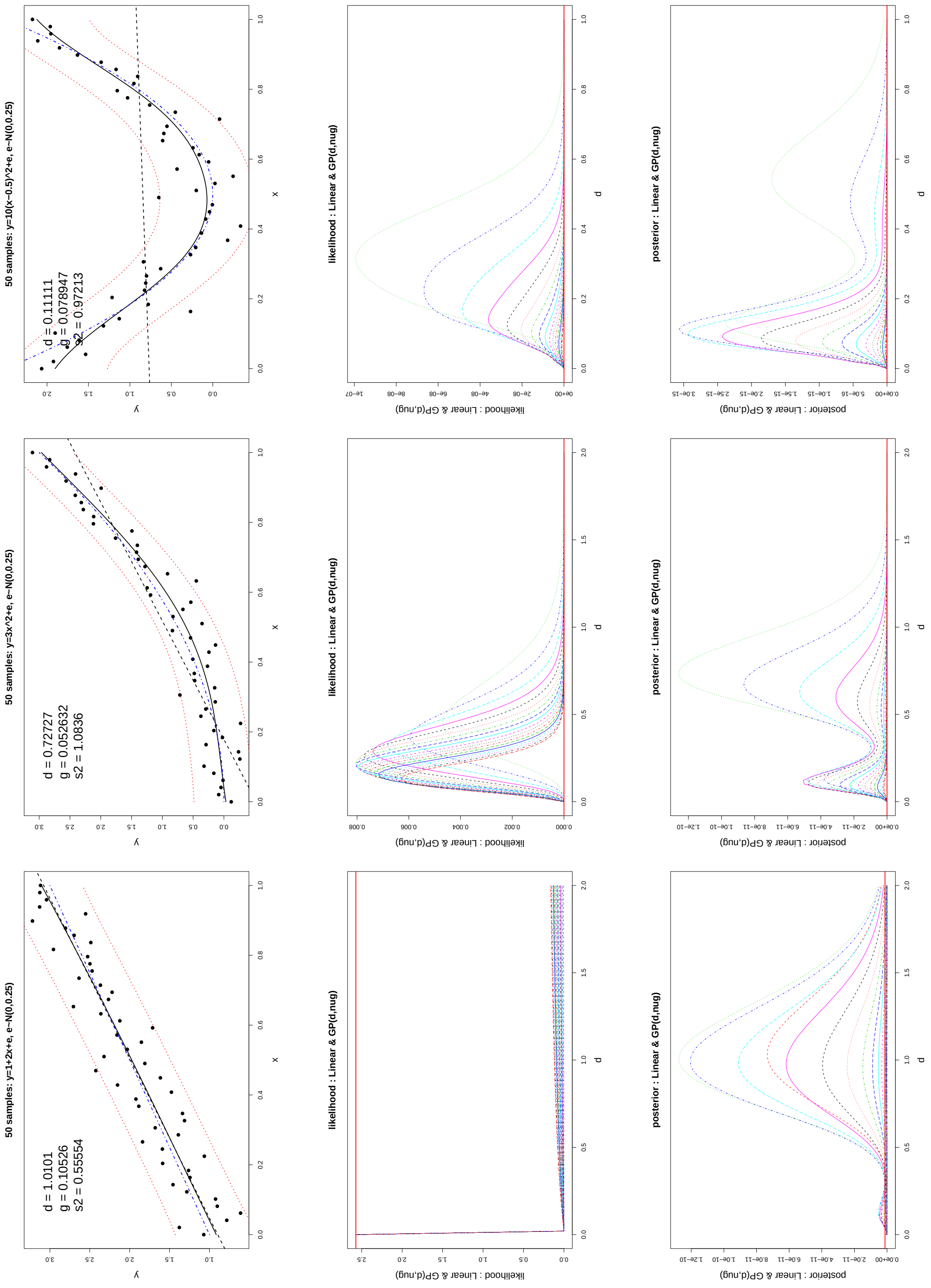


Figure 3.8: *Top row* (when rotated) shows the GP(d, g) fits; *Middle row* shows likelihoods and *bottom row* shows the integrated posterior distribution for range (d, x-axis) and nugget (g, lines) settings for four samples, one per each column.

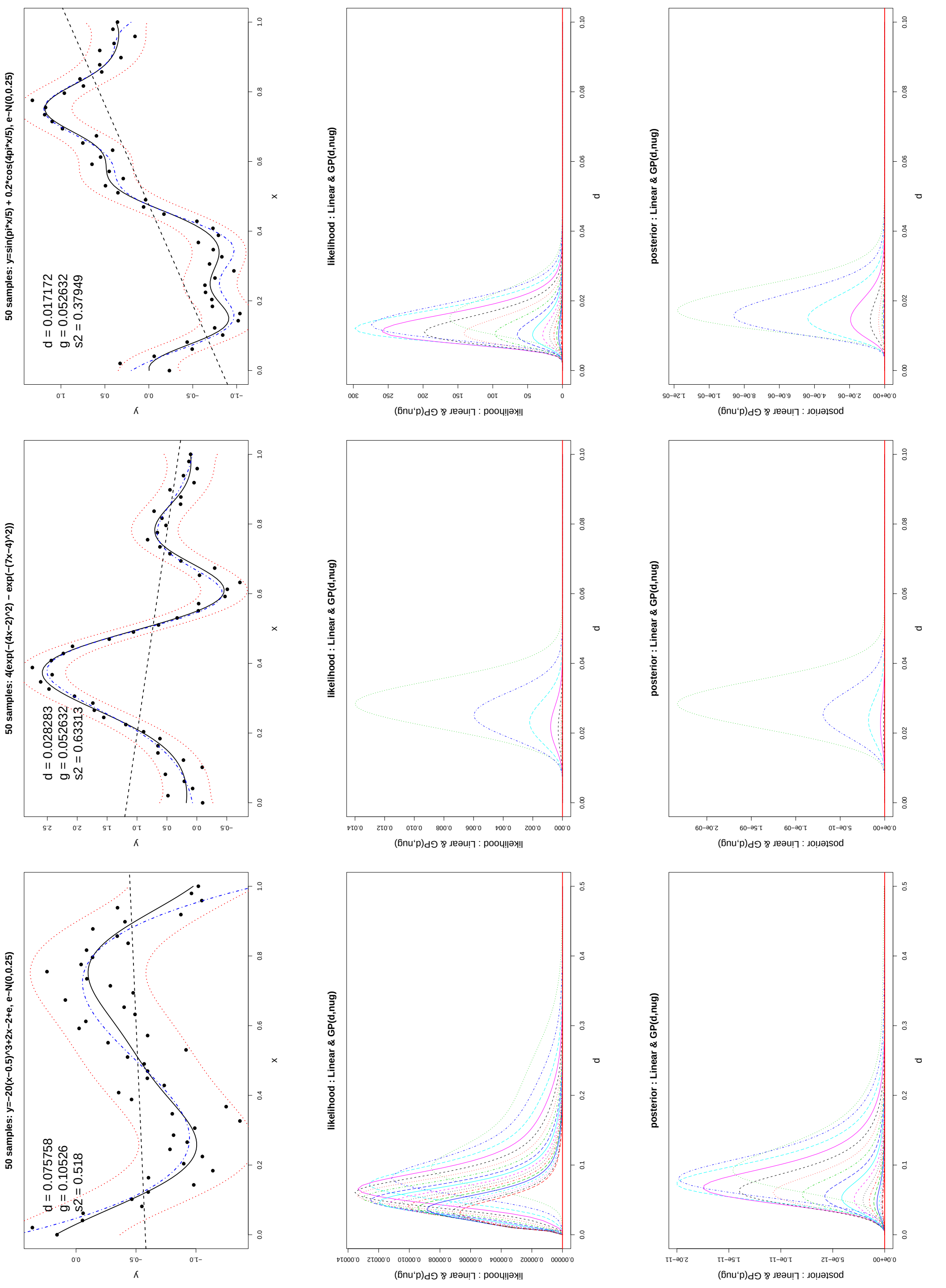


Figure 3.9: Continued from Figure 3.8.

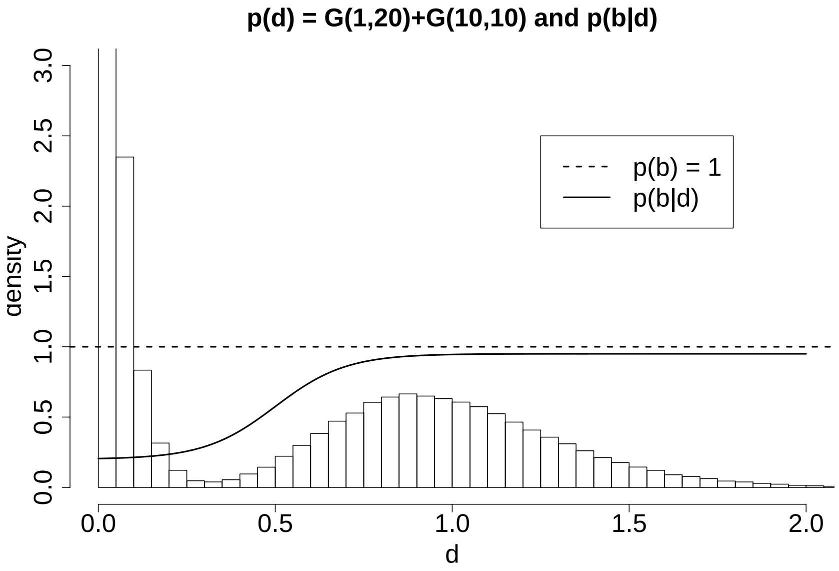


Figure 3.10: Prior distribution for the boolean (b) superimposed on $p(d)$.

3.2 Model Selection Priors

Motivated by the discussion above, this section sets out to construct a prior for the "mixture" of the GP with its LLM. The key idea is an augmentation of the parameter space by m_X indicators $\mathbf{b} = \{b\}_{i=1}^{m_X} \in \{0,1\}^{m_X}$. The boolean b_i is intended to select either the GP ($b_i = 1$) or its LLM for the i^{th} dimension. The actual range parameter used by the correlation function is multiplied by $\mathbf{b}$: e.g., $K^*(\cdot, \cdot|\mathbf{bd})$.[1] To encode the preference that GPs with larger range parameters be more likely to "jump" to the LLM, the prior on b_i is specified as a function of the range parameter d_i: $p(b_i, d_i) = p(b_i|d_i)p(d_i)$.

Probability mass functions which increase as a function of d_i, e.g.,

$$p_{\gamma,\theta_1,\theta_2}(b_i = 0|d_i) = \theta_1 + \frac{\theta_2 - \theta_1}{1 + \exp\{-\gamma(d_i - 0.5)\}} \tag{3.3}$$

[1] i.e. component-wise multiplication—like the "$\mathbf{b}.{*}\mathbf{d}$" operation in `Matlab`

with $0 < \gamma$ and $0 \leq \theta_1 \leq \theta_2 < 1$, can encode such a preference by calling for the exclusion of dimensions i with with large d_i when constructing $\mathbf{K}$. Thus b_i determines whether the GP or the LLM is in charge of the marginal process in the i^{th} dimension. Accordingly, θ_1 and θ_2 represent minimum and maximum probabilities of jumping to the LLM, while γ governs the rate at which $p(b_i = 0|d_i)$ grows to θ_2 as d_i increases. Figure 3.10 plots $p(b|d)$ with $(\gamma, \theta_1, \theta_2) = (10, 0.2, 0.95)$ superimposed on the mixture of Gamma prior $p(d_i)$ from (3.2). The θ_2 parameter is taken to be strictly less than one so as not to preclude a GP which models a genuinely nonlinear surface using an uncommonly large range setting.

The implied prior probability of the full m_X-dimensional LLM is

$$p(\text{linear model}) = \prod_{i=1}^{m_X} p(b_i = 0|d_i) = \prod_{i=1}^{m_X} \left[\theta_1 + \frac{\theta_2 - \theta_1}{1 + \exp\{-\gamma(d_i - 0.5)\}}\right]. \tag{3.4}$$

The resulting process is still a GP if any of the booleans b_i are one. The primary computational advantage associated with the LLM is foregone unless all of the b_i's are zero. However, the intermediate result offers an improvement in numerical stability in addition to describing a unique transitionary model lying somewhere between the GP and the LLM. Specifically, it allows for the implementation of semiparametric stochastic processes like $Z(\mathbf{x}) = \boldsymbol{\beta} f(\mathbf{x}) + \varepsilon(\tilde{\mathbf{x}})$ representing a piecemeal spatial extension of a simple linear model. The first part $(\boldsymbol{\beta} f(\mathbf{x}))$ of the process is linear in some known function of the the full set of covariates $\mathbf{x} = \{x_i\}_{i=1}^{m_X}$, and $\varepsilon(\cdot)$ is a spatial random process, e.g., a GP, which acts on a subset of the covariates $\tilde{\mathbf{x}}$. Such models are commonplace in the statistics community (Dey et al., 1998). Traditionally, $\tilde{\mathbf{x}}$ is determined and fixed *a priori*. The separable boolean prior in (3.3) implements an adaptively semiparametric process where the subset $\tilde{\mathbf{x}} = \{x_i : b_i = 1, i = 1, \ldots, m_X\}$ is given a prior distribution, instead of being fixed.

Note that since the isotropic family has only one range parameter, only one boolean

b is needed, and the product can be dropped from (3.4).

3.2.1 Prediction

Prediction under the limiting GP model is a simplification of Eqs. (2.17) and (2.18) since it is known that $\mathbf{K} = (1+g)\mathbf{I}$. A characteristic of the standard linear model is that all input configurations ($\mathbf{x}$) are treated as independent conditional on knowing $\boldsymbol{\beta}$. This additionally implies that in (2.17) and (2.18) the terms $k(\mathbf{x})$ and $K(\mathbf{x}, \mathbf{x})$ are zero for all $\mathbf{x}$. Thus, the predicted value of z at $\mathbf{x}$ is normally distributed with mean $\hat{z}(\mathbf{x}) = \mathbf{f}^\top(\mathbf{x})\tilde{\boldsymbol{\beta}}$, and variance

$$\begin{aligned}\hat{\sigma}(\mathbf{x})^2 &= \sigma^2[1 + \tau^2\mathbf{f}^\top(\mathbf{x})\mathbf{W}\mathbf{f}(\mathbf{x}) \\ &\quad - \tau^2\mathbf{f}^\top(\mathbf{x})\mathbf{W}\mathbf{F}^\top((1+g)\mathbf{I} + \tau^2\mathbf{F}\mathbf{W}\mathbf{F}^\top)^{-1}\mathbf{F}\mathbf{W}\mathbf{f}(\mathbf{x})\tau^2].\end{aligned} \tag{3.5}$$

It is helpful to re-write the above expression for the variance as

$$\begin{aligned}\hat{\sigma}(\mathbf{x})^2 &= \sigma^2[1 + \tau^2\mathbf{f}^\top(\mathbf{x})\mathbf{W}\mathbf{f}(\mathbf{x})] \\ &\quad - \sigma^2\left[\frac{\tau^2}{1+g}\mathbf{f}^\top(\mathbf{x})\mathbf{W}\mathbf{F}^\top\left(\mathbf{I} + \frac{\tau^2}{1+g}\mathbf{F}\mathbf{W}\mathbf{F}^\top\right)^{-1}\mathbf{F}\mathbf{W}\mathbf{f}(\mathbf{x})\tau^2\right].\end{aligned} \tag{3.6}$$

A matrix inversion lemma called the Woodbury formula (Golub & Van Loan, 1996) [pp. 51] or the Sherman-Morrison-Woodbury formula (Bernstein, 2005) [pp. 67; best to see `Mathworld` for easy access to both formulas]: states that for $(\mathbf{I} + \mathbf{V}^\top\mathbf{A}\mathbf{V})$ non-singular

$$(\mathbf{A}^{-1} + \mathbf{V}\mathbf{V}^\top)^{-1} = \mathbf{A} - (\mathbf{A}\mathbf{V})(\mathbf{I} + \mathbf{V}^\top\mathbf{A}\mathbf{V})^{-1}\mathbf{V}^\top\mathbf{A}.$$

Taking $\mathbf{V} \equiv \mathbf{F}^\top (1+g)^{-\frac{1}{2}}$ and $\mathbf{A} \equiv \tau^2 \mathbf{W}$ in (3.6) gives

$$\hat{\sigma}(\mathbf{x})^2 = \sigma^2 \left[1 + \mathbf{f}^\top(\mathbf{x}) \left(\frac{\mathbf{W}^{-1}}{\tau^2} + \frac{\mathbf{F}^\top \mathbf{F}}{1+g} \right)^{-1} \mathbf{f}(\mathbf{x}) \right]. \tag{3.7}$$

Not only is (3.7) a simplification of the predictive variance given in (3.5), but it should be familiar. Recall the expression for the posterior variance of the regression coefficients $\mathbf{V}_{\tilde{\beta}}$ given in (2.4). Writing $\mathbf{V}_{\tilde{\beta}}$ with $\mathbf{K}^{-1} = \mathbf{I}/(1+g)$ gives

$$\mathbf{V}_{\tilde{\beta}} = \left(\frac{\mathbf{W}^{-1}}{\tau^2} + \frac{\mathbf{F}^\top \mathbf{F}}{1+g} \right)^{-1}.$$

What this means is that the predictive variance for the LLM is actually

$$\hat{\sigma}(\mathbf{x})^2 = \sigma^2 \left[1 + \mathbf{f}^\top(\mathbf{x}) \mathbf{V}_{\tilde{\beta}} \mathbf{f}(\mathbf{x}) \right]. \tag{3.8}$$

But this is just the usual result for the predictive variance at $\mathbf{x}$ under the standard linear model. What serendipity! Therefore, the posterior predictive distribution under the LLM is simply

$$y(\mathbf{x}) = N[\mathbf{f}^\top(\mathbf{x})\tilde{\boldsymbol{\beta}}, \sigma^2(1 + \mathbf{f}^\top(\mathbf{x})\mathbf{V}_{\tilde{\beta}}\mathbf{f}(\mathbf{x}))]. \tag{3.9}$$

This means that there is a choice when it comes to obtaining samples from the posterior predictive distribution under the LLM. Eq. (3.8) is preferred over (3.5) because the latter involves inverting the $N \times N$ matrix, $\mathbf{I} + \tau^2 \mathbf{F}\mathbf{W}\mathbf{F}^\top/(1+g)$, whereas the former only requires the inversion of an $m_X \times m_X$ matrix.

3.3 Implementation, results, and comparisons

Here, the GP with jumps to the LLM (hereafter GP LLM) is illustrated on synthetic and real data. Most of the experiments are in the context of applying the GP LLM at the leaves of the tree, upgrading the treed GP model of Chapter 2 to a treed GP LLM model. However, Section 3.3.4 shows an example without treed partitioning. Partition models are an ideal setting for evaluating the utility of the GP LLM as linearity can be extracted in large areas of the input space. The result is a uniquely tractable nonstationary semiparametric spatial model.

A separable correlation function is used throughout this section for brevity and consistency, even though in some cases the process which generated the data is clearly isotropic. Recall that experiments in Section 2.5 of the last chapter all used the isotropic power family. Proposals for the booleans $\mathbf{b}$ are drawn from the prior, conditional on $\mathbf{d}$, and accepted or rejected on the basis of the constructed covariance matrix $\mathbf{K}$. The same prior parameterizations are used for all experiments unless otherwise noted, the idea being to develop a method that works "right out of the box" as much as possible.

3.3.1 1-d Synthetic Sinusoidal data

Recall the synthetic sinusoidal data from Section 2.5.1. The *top* panel of Figure 3.11 shows a plot of $z(x)$ evaluated as in (2.20) (with noise) for $n = 100$ evenly spaced $x-$values. A posterior predictive surface is also shown, represented by the mean and 90% quantile lines which were estimated using the treed GP LLM model. In this example, the linear model was preferred for 42% of the input domain area on average over 5,000 MCMC samples from the posterior. It is known from (2.20) that the process is linear for exactly half of the domain. The *bottom* panel of Figure 3.11 shows a histogram of the areas under the LLM for each MCMC sample of 20 repeated draws of size $n = 100$ from (2.20). The mode can be seen to be near 0.5.

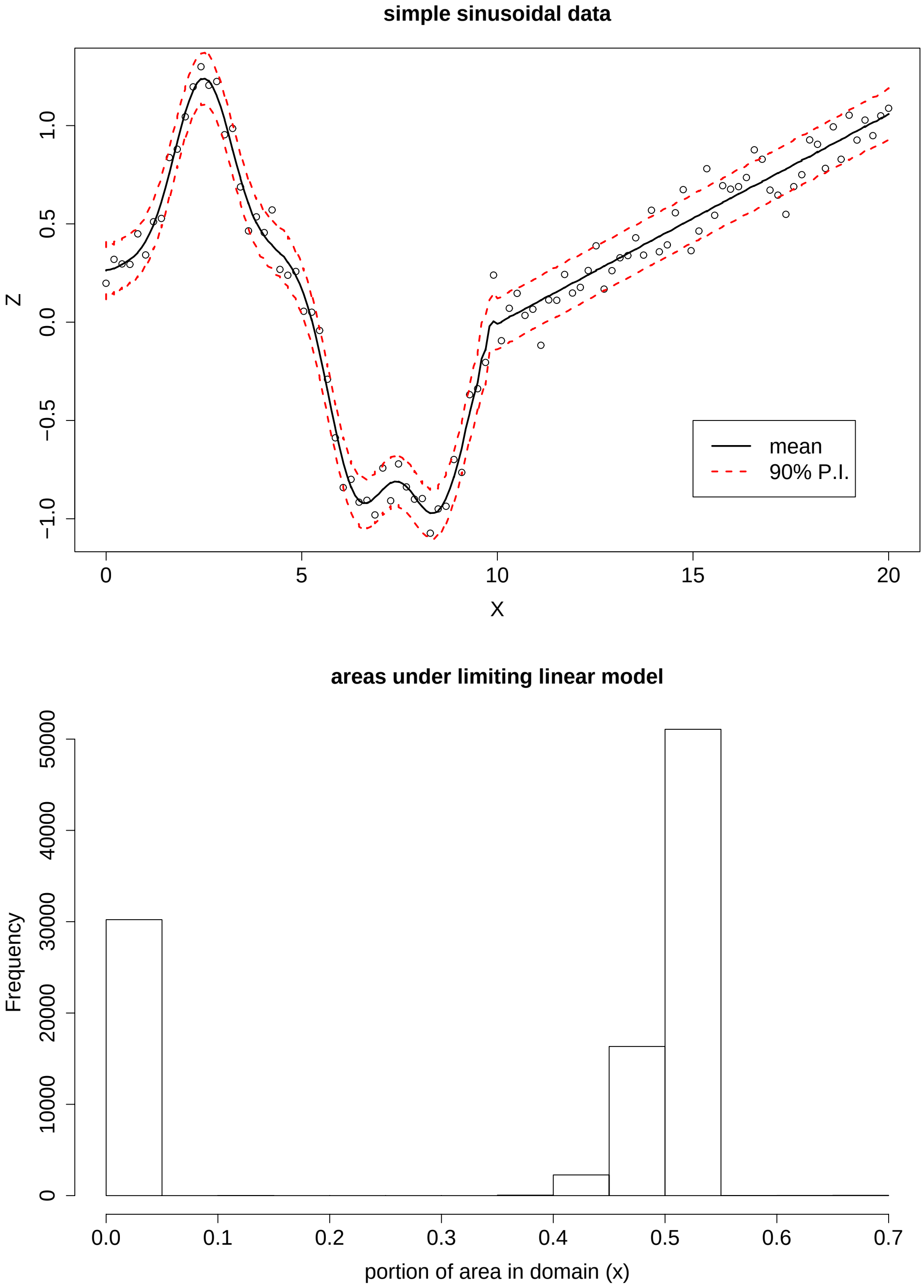


Figure 3.11: *Top:* sinusoidal data (2.20) fit with the treed GP LLM for $n = 100$ evenly spaced x-values. An average of $\sim$ 42% of the (domain) of the process was under the LLM. *bottom:* histogram of the areas of the domain under the LLM spread over 20 repeated $n = 100$ samples from (2.20).

A similar experiment of 20-fold repeated draws of size $n = 100$ and predicting at $n' = 200$ new locations, revealed that the treed GP LLM was 27% faster than treed GP alone.

3.3.2 2-d Synthetic Exponential data

Recall from Section 2.5.2 the 2-d input space $[-2, 6] \times [-2, 6]$ in which the true response is given by $z(\mathbf{x}) = x_1 \exp(-x_1^2 - x_2^2) + \epsilon$, where $\epsilon \sim N(0, \sigma = 0.001)$. Figure 3.12 summarizes the consequences of estimation and prediction with the treed GP LLM for a $n = 200$ sub-sample of this data from a regular grid of size 441. The partitioning structure of the treed GP LLM first splits the region into two halves, one of which can be fit linearly. It then recursively partitions the half with the "action" into a piece which requires a GP and another piece which is also linear. The *top* panel shows a mean predictive surface wherein the LLM was used in over 66% of the domain on average. This surface was obtained in less than ten seconds on a 1.8 GHz Athalon. The *bottom* panel shows a histogram of the areas of the domain under the LLM over 20-fold repeated experiments. The four modes of the histogram clump around 0%, 25%, 50%, and 75% showing that most often the obvious three-quarters of the space are under the LLM, although sometimes one of the two partitions will use a very smooth GP. The treed GP LLM was 40% faster than the treed GP alone when combining estimation and sampling from the posterior predictive distributions at the remaining $n' = 241$ points from the grid.

3.3.3 Motorcycle data

Recall the Motorcycle Accident Dataset from Section 2.5.3. Figure 3.13 shows the data, and a fit using the treed GP LLM. The *top* panel shows the mean predictive surface, with 90% quantile error-bars. From the *bottom* panel, which shows the difference in 95% and 5% quantiles, it is clear that the tree structure typically partitions the space into three parts. On

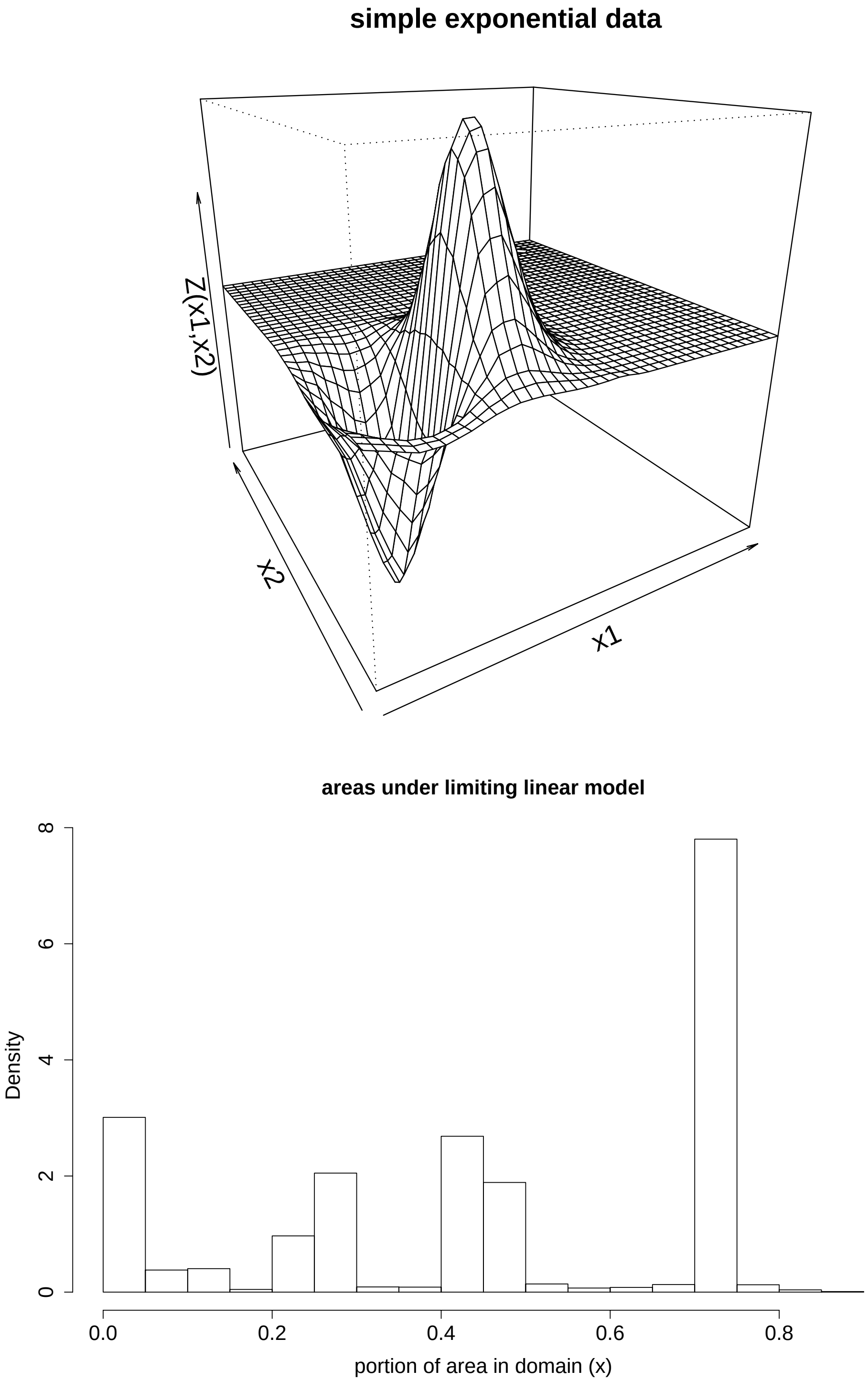


Figure 3.12: *Top:* exponential data treed GP LLM fit. *Bottom:* histogram of the areas under the LLM.

average, 29% of the domain was under the LLM, split between the left low-noise region (before impact) and the noisier right region. Visually, there is little difference between the fit in Figure 3.13 and the one in Figure 2.7, which did not use jumps to the LLM.

Rasmussen & Ghahramani (2002) analyzed this data by using a Dirichlet process mixture of Gaussian process (DPGP) experts which reportedly took one hour on a 1 GHz Pentium. Such times are typical of nonstationary modeling because of the computational effort required to construct and invert large covariance matrices. In contrast, the treed GP LLM fits this dataset with comparable accuracy but in less than one minute on a 1.8 GHz Athalon.

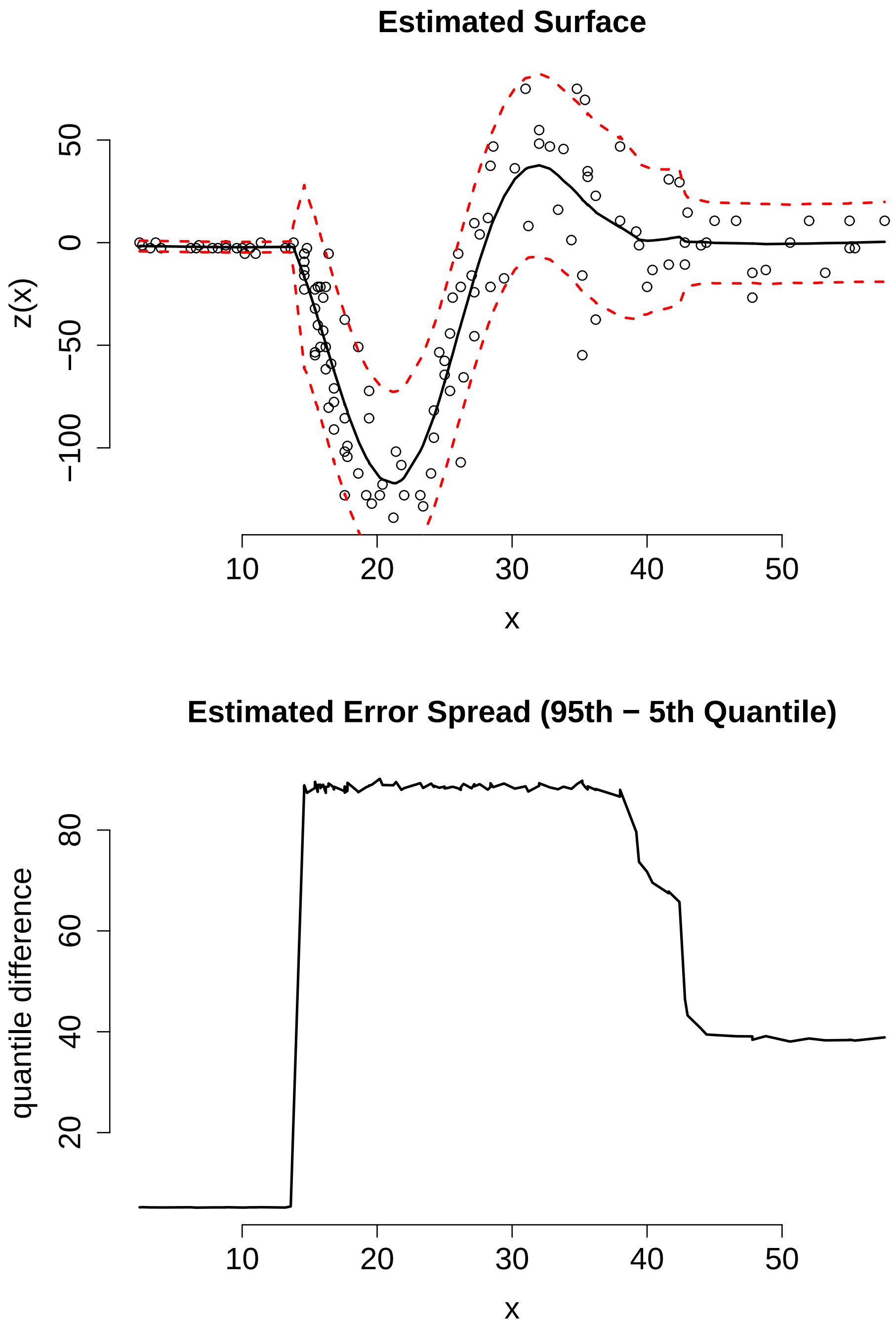
Three things make the treed GP LLM so fast relative to most nonstationary spatial models. (1) Partitioning fits models to less data, yielding smaller matrices to invert. (2) Jumps to the LLM mean fewer inversions all together. (3) MCMC mixes better because under the LLM the parameters $\mathbf{d}$ and g are out of the picture and all sampling can be performed via Gibbs steps.

3.3.4 Friedman data

This Friedman data set is the first one of a suite that was used to illustrate MARS (Multivariate Adaptive Regression Splines) (Friedman, 1991). There are 10 covariates in the data ($\mathbf{x} = \{x_1, x_2, \ldots, x_{10}\}$), but the function that describes the responses (Z), observed with standard Normal noise,

$$E(Z|\mathbf{x}) = \mu = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5, \tag{3.10}$$

depends only on $\{x_1, \ldots, x_5\}$, thus combining nonlinear, linear, and irrelevant effects. Comparisons are made on this data to results provided for several other models in recent literature. Chipman et al. (2002) used this data to compare their linear CART algorithm to four other



,

Figure 3.13: *Top:* Motorcycle Data fit by treed GP LLM. *Bottom:* and quantile differences.

methods of varying parameterization: linear regression, greedy tree, MARS, and neural networks. The statistic they use for comparison is root mean-square error (RMSE)

$$\mathrm{MSE} = \textstyle\sum_{i=1}^{n}(\mu_i - \hat{z}_i)^2/n \qquad\qquad \mathrm{RMSE} = \sqrt{\mathrm{MSE}}$$

where $\hat{z}_i$ is the model-predicted response for input $\mathbf{x}_i$. The $\mathbf{x}$'s are randomly distributed on the unit interval. RMSE's are gathered for fifty repeated simulations of size $n = 100$ from (3.10). Chipman et al. provide a nice collection of boxplots showing the results. However, they do not provide any numerical results, so I have extracted some key numbers from their plots and refer the reader to that paper for the full results.

I duplicated this experiment using the GP LLM. For this dataset, a single model was used, not a treed model, as the function is essentially stationary in the spatial statistical sense (so if I were to try to fit a treed GP, it would keep all of the data in a single partition). Linearizing boolean prior parameters $(\gamma, \theta_1, \theta_2) = (10, 0.2, 0.9)$ were used, which gave the LLM a relatively low prior probability of 0.35, for large range parameters d_i. The RMSEs obtained for the GP LLM are summarized in the table below.

	Min	1st Qu.	Median	Mean	3rd Qu.	Max
GP LLM	0.4341	0.5743	0.6233	0.6258	0.6707	0.7891
LM	1.710	2.165	2.291	2.325	2.500	2.794

Results on the linear model are reported for calibration purposes, and can be seen to be essentially the same as those reported by Chipman et al.. RMSEs for the GP LLM are on average significantly better than *all* of those reported for the above methods, with lower variance. For example, the best mean RMSE shown in the boxplot is ≈ 0.9. That is 1.4 times higher than the worst one obtained for GP LLM. Further comparison to the boxplots provided by Chipman et al. shows that the GP LLM is the clear winner.

In fitting the model, the Markov Chain quickly keyed in on the fact that only the first three covariates contribute nonlinearly. After burn-in, the booleans **b** almost never deviated from $(1,1,1,0,0,0,0,0,0,0)$. From the following table summarizing the posterior for the linear regression coefficients $\boldsymbol{\beta}$ it can be seen that the coefficients for x_4 and x_5 (between double-bars) were estimated accurately, and that the model correctly determined that $\{x_6, \dots x_{10}\}$ were irrelevant, i.e., not included in the GP, and had β's close to zero.

		x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
$\boldsymbol{\beta}$	5% Qu.	8.40	2.60	-1.23	-0.89	-1.82	-0.60	- 0.91
	Mean	9.75	4.59	-0.190	0.049	-0.612	0.326	0.066
	95% Qu.	10.99	9.98	0.92	1.00	0.68	1.21	1.02

For a final comparison, consider an SVM method (Drucker et al., 1996) illustrated on this data and compared to Bagging (Breiman, 1996) regression trees. Note that the SVM method required cross-validation (CV) to set some of its parameters. In the comparison, 100 randomized training sets of size $n = 200$ were used, and MSEs were collected for a (single) test set of size $n' = 1000$. An average MSE of 0.67 is reported, showing the SVM to be uniformly better than the Bagging method with an MSE of 2.26. I repeated the experiment for the GP LLM (which requires no CV!), and obtained an average MSE of 0.293, which is 2.28 times better than the SVM, and 7.71 times better than Bagging.

3.3.5 Boston housing data

A commonly used data set for validating multivariate models is the Boston Housing Data (Harrison & Rubinfeld, 1978) available from the UCI Machine Learning repository (Newman et al., 1998), which contains 506 responses over 13 covariates. Chipman et al. (2002) showed that their (Bayesian) linear CART model gave lower RMSEs, on average, compared to a number of popular techniques (the same ones listed above). The treed GP LLM is a generalization of the linear CART model, retaining the original linear CART as an accessible special case. Though computationally more intensive than linear CART, the treed GP LLM gives impressive results. To mitigate some of the computational demands, the LLM can be used to initialize the Markov Chain by breaking the larger data set into smaller partitions. Before treed GP burn-in begins, the model is fit using only the faster (limiting) linear CART model. Once the treed partitioning has stabilized, this fit is taken as the starting value for a full MCMC exploration of the posterior for the treed GP LLM. This initialization process allows fitting of GPs to smaller segments of the data, reducing the size of matrices that need to be inverted and greatly reducing computation time. For the Boston Housing data, the settings $(\gamma, \theta_1, \theta_2) = (10, 0.2, 0.95)$ were used, which gives the LLM a prior probability of $0.95^{13} \approx 0.51$, when the d_i's are large.

Experiments in the Bayesian linear CART paper (Chipman et al., 2002) consist of calculating RMSEs via 10-fold CV. The data are randomly partitioned into 10 groups, iteratively trained on 9/10 of the data, and tested on the remaining 1/10. This is repeated for 20 random partitions, and boxplots are shown. The logarithm of the response is used, and CV is only used to assess predictive error, not to tune parameters. Samples are gathered from the posterior predictive distribution of the linear CART model for six parameterizations using 20 restarts of 4000 iterations. This seemed excessive, but I followed suit for the treed GP LLM in order to obtain a fair comparison. My "boxplot" for training and testing RMSEs are summarized numerically in the table below. As before, linear regression (on the log responses) is used for calibration.

		Min	1st Qu.	Median	Mean	3rd Qu.	Max
train	GP LLM	0.0701	0.0716	0.0724	0.0728	0.0730	0.0818
	LM	0.1868	0.1869	0.1869	0.1869	0.1869	0.1870
test	GP LLM	0.1321	0.1327	0.1346	0.1346	0.1356	0.1389
	LM	0.1926	0.1945	0.1950	0.1950	0.1953	0.1982

The RMSEs for the linear model have extremely low variability. This is similar to the results provided by Chipman et al. and was a key factor in determining that the experiment was well-calibrated. Upon comparison of the above numbers with the boxplots in Chipman et al., it can readily be seen that the treed GP LLM is leaps and bounds better than linear CART, and *all* of the other methods in the study. The treed GP LLM's worst training RMSE is almost two times lower than the best ones from the boxplot. All testing RMSEs are lower than the lowest ones from the boxplot, and the median RMSE (0.1346) is 1.26 times lower than the lowest median RMSE (≈ 0.17) from the boxplot.

More recently, Chu et al. (Chu et al., 2004) [see Table V] performed a similar experiment, but instead of 10-fold CV, they randomly partitioned the data 100 times into training/test sets of size 481/25 and reported average MSEs on the un-transformed responses. They compare their Bayesian SVM regression algorithm (BSVR) to other high-powered techniques like Ridge Regression, Relevance Vector Machine, GPs, etc., with and without ARD (automatic relevance determination). Repeating their experiment for the treed GP LLM gave an average MSE of 6.96 compared to that of 6.99 for the BSVR with ARD, making the two algorithms by far the best in the comparison. However, without ARD the MSE of BSVR was 12.34, 1.77 times higher than the treed GP LLM, and the worst in the comparison. The reported results for a GP with (8.32) and without (9.13) ARD showed the same effect, but to a lesser degree. Thus the GP LLM might similarly benefit from an ARD-like approach. Perhaps not surprisingly, the average MSEs do not tell the whole story. The 1st, median, and 3rd quantile MSEs obtained for the treed GP LLM were 3.72, 5.32 and 8.48 respectively, showing that its distribution had a heavy right-hand tail. This may be an indication that several responses in the data are either misleading, noisy, or otherwise very hard to predict.

3.4 Conclusion

Gaussian processes are a flexible modeling tool which can be overkill for many applications. This chapter has shown how the limiting linear model parameterization of the GP can be both useful and accessible in terms of Bayesian posterior estimation and prediction. The benefits include speed, parsimony, and a relatively straightforward implementation of a semiparametric model. Combined with treed partitioning, the GP LLM extends linear CART, resulting in a uniquely nonstationary, semiparametric, tractable, and highly accurate regression tool.

The next chapter will demonstrate the impact of the treed GP LLM employed as a surrogate model for the sequential design of computer experiments. Empirical evidence suggests that many computer experiments are nearly linear. That is, either the response is linear in most of its input dimensions, or the process is entirely linear in a subset of the input domain. The Bayesian treed GP LLM provides a *full* posterior predictive distribution (particularly a nonstationary and thus region-specific estimate of predictive variance) that can be used towards active learning in the input domain. Exploitation of these characteristics can yield an efficient framework for the adaptive exploration of computer experiment parameter spaces.

Chapter 4

Adaptive Sampling

Much of the current work in large-scale computer models starts by evaluating the model over a hand-crafted grid of input configurations. After the full grid has been run, a human may identify interesting regions and perform additional runs if desired.

This chapter is concerned with developing improvements to this approach. The first task is to introduce the asynchronous distributed computer model commonly used to run complex computer codes. Protocols can then be developed and used to simulate state-of-the-art supercomputers. Methodologies can then explored for choosing new input configurations based on region-specific estimates of uncertainty, provided by the nonstationary treed GP, and/or GP LLM surrogate model. Illustrations are carried out on synthetic nonstationary data sets. Finally, a fully developed, asynchronous, Bayesian adaptive sampling (BAS) framework is interfaced with NASA supercomputers in order to sequentially design an experiment for a re-usable launch vehicle called the Langley Glide-Back booster (LGBB).

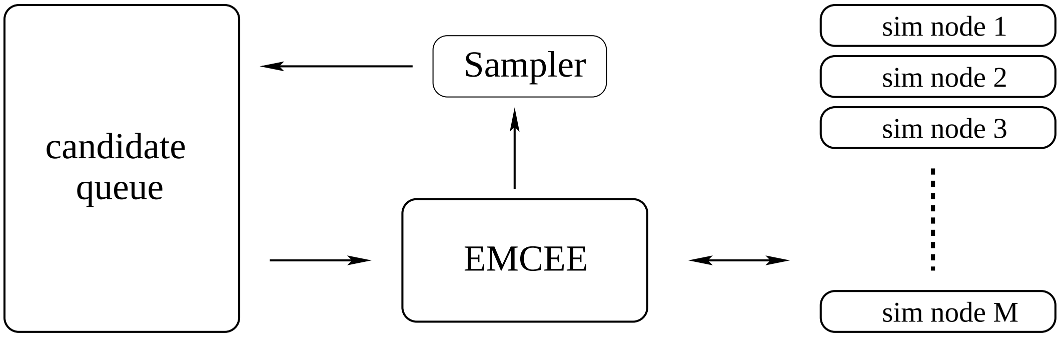


Figure 4.1: *Emcee* program feeds the adaptive sampler with finished responses, and selects new configurations from a queue constructed by the sampler.

4.1 Asynchronous distributed computing

High fidelity computer experiments are usually run on clusters of independent computing agents, or processors. A `Beowulf` cluster is a good example. At any given time, each agent is working on a single input configuration. Multiple agents allow several input configurations to be run in parallel. Simulations for new configurations begin when an agent finishes execution and becomes available. Therefore, simulations may start and finish at different, perhaps even random, times. The cluster is usually managed asynchronously by a master controller *(emcee)* program that gathers responses from finished simulations, and supplies free agents with new input configurations.

The goal is to have the *emcee* program interact with an adaptive sampling program that supplies it with well-chosen candidates. In turn, the *emcee* feeds the sampling program with finished responses when they become available, so that the surrogate model can be updated. A diagram of this process is shown in Figure 4.1. A treed GP or GP LLM is an ideal surrogate model because it is fast, nonstationary, semiparametric, and can provide region-specific estimates of uncertainty. The next section describes how the adaptive sampler can use the surrogate model to help populate the *emcee's* candidate queue, implementing an asynchronous sequential design of experiments.

4.2 Asynchronous sequential DOE via Active Learning

Active learning, or sequential design of experiments (DOE), in the context of estimating response surfaces, is called *adaptive sampling*. Adaptive sampling starts with a relatively small space-filling “peppering” of input data, and then proceeds by fitting a model, estimating predictive uncertainty, and then choosing future samples with the aim of minimizing some measure of uncertainty, or to try to maximize information. The process repeats until some threshold in predictive uncertainty or information is met, or a maximum number of samples have been taken. In this iterative fashion the model adapts to the data, and the new data either reinforces, or suggests a modification to, the old model.

Nonstationary models like the treed GP model from Chapter 2, and the treed GP LLM model of Chapter 3, fit independent stationary models in different regions of the input space. Uncertainty in partition models and uncertainty in the posterior predictive response distribution can vary over the input space. Region-specific uncertainty estimates can guide sampling. As responses become available, the adaptive sampler can update the model, and make more informed decisions in the future.

In the statistics community, there are a number of established methodologies for (sequentially) designing experiments [see Section 1.2.4]. However, some classic criticisms for traditional DOE approaches precluded such a canned approach. For example, the number of support points in an optimal design is often equal to the number of model parameters; these points are usually closer to the boundary of the region, where measurement error can be severe, and responses can be difficult to elicit, and model checking is often not feasible. Possible remedies may arise when one considers designs that account for model uncertainty (DuMouchel & Jones, 1994; DuMouchel & Jones, 1985; O'Hagan, 1985). According to Chaloner & Verdinelli (1995) "a tradeoff is recognized between choosing design points on the boundary ... to maximize information and choosing them toward the center ... where the model is believed to hold better approximation." Other reasons for not taking the standard statistical approach include speed, the difficulty inherent in using Monte Carlo to estimate the surrogate model, lack of support for partition models, and the desire to design for an asynchronous *emcee* interface where responses and computing nodes become available at random times.

My solution takes a hybrid approach that combines standard DOE with methods from the Active Learning literature [see Section 1.2.4]. The basic idea is to use optimal sequential designs from the DOE literature, like $D-$optimal, minimax, or LH, as candidates for future sampling. Then, the treed GP or GP LLM algorithm can provide Monte Carlo estimates of model uncertainty, via the ALM or ALC algorithm, which can be used to populate, and sequence, the candidate queue used by the *emcee* [see Figure 4.1]. That way, candidates are well-spaced out relative to themselves, and to the already sampled locations. Additionally, the most informative of these candidates can be first in line for simulation when agents become available. Fleshing these ideas out is the focus of the following two subsections. Details pertaining to implementation are left to Section 4.3.

For the remainder of this chapter I shall refer to the surrogate model as "treed GP", rather than "treed GP with or without jumps to the LLM". Most of the experimental results presented here use the treed GP LLM. Results without jumps to the LLM are strikingly similar, but with a slower implementation in some cases.

4.2.1 ALM and ALC algorithms

A hybrid approach to designing experiments employs active learning techniques. The idea is to choose a set of candidate input configurations $\tilde{\mathbf{X}}$ (say, a $D-$optimal or LH design) and an active learning rule for determining the order in which they should be added into the design. Two criteria for choosing new sampling locations have been proposed [see Section 1.2.4]. Both are based on the posterior predictive distribution $P(z|\mathbf{x})$. For example, consider an approach which maximizes the information gained about model parameters by selecting the location $\tilde{\mathbf{x}} \in \tilde{\mathbf{X}}$ which has the greatest standard deviation in predicted output. This approach has been called ALM for Active Learning–Mackay, and has been shown to approximate maximum expected information designs (MacKay, 1992). MCMC posterior predictive samples provide a convenient estimate of location-specific variance, namely the width of predictive quantiles.

An alternative algorithm is to select $\tilde{\mathbf{x}}$ minimizing the expected reduction in squared error averaged over the input space (Cohn, 1996), called ALC for Active Learning–Cohn. Rather than focusing on design points which have large predictive variance, ALC selects configurations that would lead to a global reduction in predictive variance. Conditioning on $\mathcal{T}$, the reduction in variance at a point $\mathbf{y} \in \mathbf{Y}_\nu$, given that the location $\tilde{\mathbf{x}} \in \tilde{\mathbf{X}}_\nu$ is added into the data, is defined as (region subscripts suppressed):

$$\begin{aligned} & \Delta\hat{\sigma}^2_{\mathbf{y}}(\mathbf{x}) = \hat{\sigma}^2_{\mathbf{y}} - \hat{\sigma}^2_{\mathbf{y}}(\mathbf{x}), \\ \text{where} \quad & \hat{\sigma}^2_{\mathbf{y}} = \sigma^2[\kappa(\mathbf{y},\mathbf{y}) - \mathbf{q}_N^\top(\mathbf{y})\mathbf{C}_N^{-1}\mathbf{q}_N^\top(\mathbf{y})], \\ \text{and} \quad & \hat{\sigma}^2_{\mathbf{y}}(\mathbf{x}) = \sigma^2[\kappa(\mathbf{y},\mathbf{y}) - \mathbf{q}_{N+1}(\mathbf{y})^\top\mathbf{C}_{N+1}^{-1}\mathbf{q}_{N+1}(\mathbf{y})]. \end{aligned}$$

The above equations use notation for the GP predictive variance for region r_ν given in (2.18). The partition inverse equations (Barnett, 1979), for a covariance matrix $\mathbf{C}_{N+1}$ in terms of $\mathbf{C}_N$, gives a means to arrive at a nice expression for $\Delta\sigma^2_{\mathbf{y}}(\mathbf{x})$:

$$\Delta\hat{\sigma}^2_{\mathbf{y}}(\mathbf{x}) = \frac{\sigma^2\left[\mathbf{q}_N^\top(\mathbf{y})\mathbf{C}_N^{-1}\mathbf{q}_N(\mathbf{x}) - \kappa(\mathbf{x},\mathbf{y})\right]^2}{\kappa(\mathbf{x},\mathbf{x}) - \mathbf{q}_N^\top(\mathbf{x})\mathbf{C}_N^{-1}\mathbf{q}_N(\mathbf{x})}. \tag{4.1}$$

The details of this derivation are included in Appendix C.1. For $\mathbf{y}$ and $\tilde{\mathbf{x}}$ not in the same region r_ν, let $\Delta\sigma^2_{\mathbf{y}}(\tilde{\mathbf{x}}) = 0$. The reduction in predictive variance that would be obtained by adding $\mathbf{x}$ into the data set is calculated by averaging over $\mathbf{y} \in \mathbf{Y}$:

$$\Delta\sigma^2(\mathbf{x}) = \frac{1}{|\mathbf{Y}|}\sum_{\mathbf{y}_i \in \mathbf{Y}} \Delta\hat{\sigma}^2_{\mathbf{y}_i}(\mathbf{x}) \tag{4.2}$$

The benefits of ALC include that $\Delta\sigma^2(\mathbf{x})$ is easily approximated using MCMC methods. Also, compared to ALM, adaptive samples under ALC are less heavily concentrated near the boundaries of partitions. The computational demands of both algorithms are a function of the number of candidate locations. However, ALC requires an order of magnitude more computing time than ALM. ALC is also more sensitive to the location and region-specific count of candidates, especially $\mathbf{Y}$. If the configurations in $\mathbf{Y}$ are not distributed uniformly thought the input space, then $\Delta\sigma^2(\tilde{\mathbf{x}})$ will be (artificially) magnified closer to high-density $\mathbf{Y}$ regions.

For a nice comparison between variance reduction techniques like ALM and ALC, including LH, on computer code data, see work by McKay et al. (1979). Seo et al. (2000) provide comparisons between ALC and ALM using standard GPs. In both papers, the model is assumed known in advance. Seo et al. take $\mathbf{Y} = \tilde{\mathbf{X}}$ to be the full set of un-sampled locations in a pre-specified uniform grid. Assuming that the model is known *a priori* is at loggerheads with adaptive sampling. If the goal of adaptive sampling is to learn the responses online, and adjust the model accordingly, then claiming to know the model ahead of time makes little sense. Also, obtaining samples from $\Delta\sigma^2_{\mathbf{y}}(\tilde{\mathbf{x}})$ via MCMC on a dense high-dimensional grid is computationally expensive.

In the treed GP application of ALC, the model is not assumed known *a priori*. Instead, Bayesian MCMC posterior inference on $\{\mathcal{T}, \boldsymbol{\theta}\}$ is performed, and then samples from $\Delta\sigma^2_{\mathbf{y}}(\tilde{\mathbf{x}})$ are taken conditional on samples from $\{\mathcal{T}, \boldsymbol{\theta}\}$. Candidates $\mathbf{Y} = \tilde{\mathbf{X}}$ can come from the sequential treed D-optimal design, described in the following subsection, so that they are well-spaced relative both to themselves and to the already sampled configurations, in order to encourage exploration.

Applying the ALC algorithm under the limiting linear model is computationally less intense compared to ALC under a full GP. Starting with the predictive variance given in (3.8), the expected reduction in variance under the linear model is:

$$\Delta\hat{\sigma}^2_{\mathbf{y}}(\mathbf{x}) = \frac{\sigma^2[\mathbf{f}^\top(\mathbf{y})\mathbf{V}_{\tilde{\beta}_N}\mathbf{f}(\mathbf{x})]^2}{1 + g + \mathbf{f}^\top(\mathbf{x})\mathbf{V}_{\tilde{\beta}_N}\mathbf{f}(\mathbf{x})}. \tag{4.3}$$

Appendix C.2 contains details of the derivation. Since only an $m_X \times m_X$ inverse is required, Eq. (4.3) is preferred over simply replacing $\mathbf{K}$ with $\mathbf{I}(1+g)$ in (4.1), which requires an $N \times N$ inverse. Averaging over $\mathbf{y}$ proceeds as in (4.2), above.

Given these two hybrid approaches to sequential design, constructing a list of input configurations for the *emcee* to send to available computing agents is simply a matter of sorting candidate locations ranked via either ALM or ALC. That way, the most informative locations are first in line for simulation when agents become available.

4.2.2 Choosing candidates

Configurations located close to one another in the input space have high likelihood of being clumped together when sorted by ALM or ALC. This means that the chances of sampling two geographically disparate input configurations is low unless their predicted uncertainties are more similar than other candidates in the neighborhood. The result can be a “clumping” of adaptive samples, rather than ones that better explore the input space. This phenomenon is largely an artifact of the delayed (and uncertain) response time for agent-based supercomputer simulation. That is, if responses were available immediately, and the model instantaneously updated, a local reduction in model uncertainty would lower the utility of neighboring configurations, giving way to exploration of other high-uncertainty regions of the input space.

Sub-sampling the remaining candidates from a grid, or choosing candidate locations randomly, are possible ways of generating candidate designs. However, they do not guard against a clumping of adaptive samples. A better approach would be to choose candidates from a sequential optimal design. That way, candidates will be spaced out relative to themselves, and relative to the configurations which have already been sampled. A sequential D-optimal design is a good first choice because it encourages exploration. But traditional D-optimal designs are based on a *known* parameterization of a single GP model, and are thus not well-suited to MCMC based treed-partition models. A D-optimal design may not choose candidates in the “interesting” part of the input space, because sampling is high there already. Classic optimal design criteria have not been designed for partition models, wherein “closeness” is not measured homogeneously across the input space.

Another disadvantage to D-optimal design is computational, for the same reason that GPs become intractable as the number of inputs gets large—namely decomposing and finding the determinant of a large covariance matrix. Since determinant space can have many local minima [see Section 1.2.4], a clever search strategy is required.

One possible solution to both computational and nonstationary modeling issues is to use treed sequential D-optimal design, outlined below.

Treed sequential optimal design

Instead of using a global sequential D-optimal design, consider computing a separate sequential D-optimal design in each of the partitions depicted by the maximum *a posteriori* (MAP) tree $\hat{\mathcal{T}}$. The number of candidates selected from each region, $\{\hat{r}_\nu\}_{\nu=1}^{\hat{R}}$ of $\hat{\mathcal{T}}$, can be proportional to the volume of the region. If working on a grid, the number can be proportional to the number of grid locations in the region. MAP parameters $\hat{\boldsymbol{\theta}}_\nu|\hat{\mathcal{T}}$ can be used in creating the candidate design, or "neutral" or "exploration encouraging" parameters can be used instead. Separating design from inference by using custom parameterizations in design steps, rather than inferred ones, is a common practice in the SDACE (sequential design and analysis of computer experiments) community (Santner et al., 2003). Small range parameters, for learning about the wiggliness of the response, and a modest nugget parameter, for numerical stability, tend to work well together.

Since optimal design is only used to select candidates, and is not the final step in adaptively choosing samples, employing a high-powered search algorithm, e.g., a genetic algorithm (Hamada et al., 2001), seems excessive. Finding a local maxima is generally sufficient to get well-spaced candidates. I chose a simple stochastic ascent algorithm which can find local maxima without calculating too many determinants. The $\hat{R}$ search algorithms can be run in parallel, and typically invert matrices much smaller than $N \times N$.

Figure 4.2 shows an example sequential treed D-optimal design for the 2-d Exponential data [Section 2.5.2 & Section 3.3.2], found by simple stochastic search. Input configurations are sub-sampled from the remaining locations in a 21×21 grid. Circles in the figure represent the chosen locations of the new candidate design $\tilde{\mathbf{X}}$ relative to the existing sampled locations $\mathbf{X}$ (dotted). There are roughly the same number of candidates in each quadrant, despite the fact that the density of samples in the first quadrant is already two-times that of the others. A

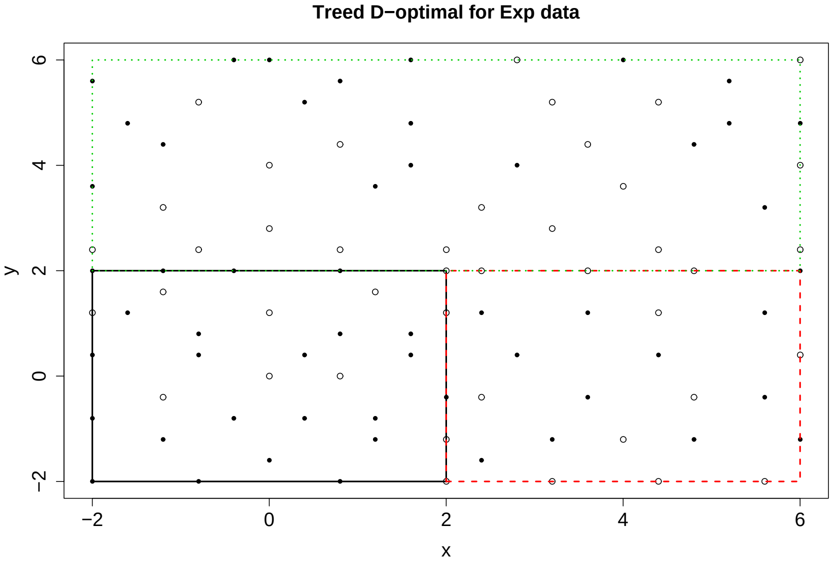


Figure 4.2: Example of a treed D-optimal design for the 2-d Exponential data sub-sampled from the remaining locations in a 21×21 grid. *Solid* dots represent previously sampled locations. *Circles* are the candidate design based on $\hat{\mathcal{T}}$, also shown.

classical (non-treed) D-optimal design would have chosen fewer points in the first quadrant in in order to equalize the density relative to the other three quadrants.

An alternative approach to using $\hat{\mathcal{T}}$ would be to average over the posterior distribution of the full surrogate model $\{\mathcal{T}, \boldsymbol{\theta}\}$, obtained via MCMC sampling. Such an approach is ambitious, but not impossible. Müller et al. (2004) show how optimal designs, posed as a decision problem, can be found via inhomogeneous Markov chain simulation. The idea is to set up a positive and bounded utility on design space which can be treated as a probability. Then, samples from the joint “posterior distribution” of design and surrogate model parameter space can be taken in a simulated annealing-*like* fashion, so that the Markov chain eventually concentrates on high utility designs [also see Section 1.2.4]. I made a serious effort in implementing this approach for obtaining optimal treed candidate designs, but had marginal success at best.

Compared to the relative simplicity and high quality of candidate designs obtained using the MAP $\hat{\mathcal{T}}$ method described above (and in Figure 4.2), the inhomogeneous Markov chain method is overkill.

4.3 Implementation methodology

Bayesian adaptive sampling (BAS) proceeds in trials. Suppose N samples and their responses have been gathered in previous trials, or from a small initial design before the first trial. In the current trial, a treed GP model is estimated for data $\{\mathbf{x}_i, z_i\}_{i=1}^N$. Samples are gathered, in accordance with the ALM or ALC algorithm conditional on $\{\boldsymbol{\theta}, \mathcal{T}\}$, at candidate locations $\tilde{\mathbf{X}}$ chosen from a sequential treed D-optimal design. The candidate queue is populated with a sorted list of candidates. BAS gathers finished and running input configurations from the *emcee* and adds them into the design. Predictive mean estimates are used as surrogate responses for unfinished (running) configurations until the true response is available. New trials start with fresh candidates.

I developed two implementations of an artificial clustered simulation environment, with a fixed number of agents, in order to simulate the parallel and asynchronous evaluation of input configurations, whose responses finish at random times. One implementation is in `C++` and uses the message passing features of `PVM` (Parallel Virtual Machine) to communicate with the adaptive sampler. The second implementation is in `Perl` and was designed to mimic, and interface with, the `Perl` modules at NASA which drive their experimental design software. Experiments on synthetic data, in the next section, will use this interface.

BAS, as used for the LGBB experiment in Section 4.4.3, interfaces with the `Perl` module developed at NASA to submit jobs to the supercomputer `Columbia`. Multi-dimensional responses, as in the LGBB experiment, are treated as independent, i.e. each response has its own treed GP surrogate model, m_Z surrogates total. Uncertainty estimates (via ALM or ALC) are pooled across the models for each response. The MAP tree $\hat{\mathcal{T}}$, used for creating sequential treed D-optimal candidates, is taken from the surrogates of each of the m_Z responses in turn.

Treating highly correlated physical measurements as independent is a crude approach. However, it still affords remarkable results, and allows the use of `PThreads` to get a highly parallel implementation. Coupled with the producer/consumer model for parallelizing prediction and estimation [from Section 2.4], a factor of $2m_Z$ speedup for $2m_Z$ processors can be obtained, where m_Z dimension of the response space. Cokriging, and other approaches to modeling multivariate (correlated) responses, are beyond the scope of this work.

Chipman et al. (1998) recommend running several parallel chains, and sub-sampling from all chains in order better explore the posterior distribution of the tree ($\mathcal{T}$). Rather than run multiple chains explicitly, the trial nature of adaptive sampling can be exploited: at the beginning of each trial the tree is restarted, or randomly pruned back. Although the tree chain associated with an individual trial may find itself stuck in a local mode of the posterior, in the aggregate of all trials the chain(s) explore the posterior of tree-space nicely. Random pruning represents a compromise between restarting and initializing the tree at a well-chosen starting place. This *tree inertia* usually affords shorter burn-in of the MCMC at the beginning of each trial. The tree can also be initialized with a run of the Bayesian Linear CART model, as in Section 3.3, for a faster burn-in of the treed GP chain.

Each trial executes at least B burn-in and T total MCMC sampling rounds. Samples are saved every E rounds in order to reduce the correlation between draws by thinning. Good default values are $B = 2000$, $T = 7000$, and $E = 2$ for a total of $(T - B)/E = 2500$ rounds in which samples are saved. Samples of ALM and ALC statistics only need be gathered every E rounds, so thinning cuts down on the computational burden as well. If the *emcee* has no responses waiting to be incorporated by BAS at the end of T MCMC rounds, then BAS can run more MCMC rounds, either continuing where it left off, or after re-starting the tree. New trials, with new candidates, start only when the *emcee* is ready with a new finished response. Such is the design so that the computing time of each BAS trial does not affect the rate of sampling. Rather, a slow BAS runs fewer MCMC rounds per finished response, and re-sorts candidates less often compared to a faster BAS. A slower adaptive sampler yields less optimal sequential samples, but always offers an improvement over naive gridding.

4.4 Results and discussion

In this section, sequential experimental designs are built for synthetic and real data with Bayesian Adaptive Sampling (BAS) and the treed GP as a surrogate model. The synthetic sinusoidal and exponential data from previous chapters facilitate illustration and comparison between ALC, ALM, and other approaches from the literature. Finally, Section 4.4.3 returns to the motivating NASA rocket boost experiment, the LGBB.

4.4.1 1-d Synthetic Sinusoidal data

Recall again the synthetic sinusoidal data from Sections 2.5.1 and 3.3.1. Figures 4.3, 4.4 & 4.5 show three snap-shots, illustrating the evolution BAS on this data using the the ALC algorithm with treed D-optimal candidates. The *top* panel of each figure plots the estimated surface in terms of posterior predictive means (solid-black) and 90% intervals (dashed-red). The MAP tree $\hat{\mathcal{T}}$ is shown as well. The *bottom* panel summarizes the ALM and ALC statistics (scaled to show alongside ALM) for comparison. Ten D-optimally spaced samples were used as an initial "peppering" design.

The snapshot in Figure 4.3 was taken after BAS had gathered a total of thirty samples.

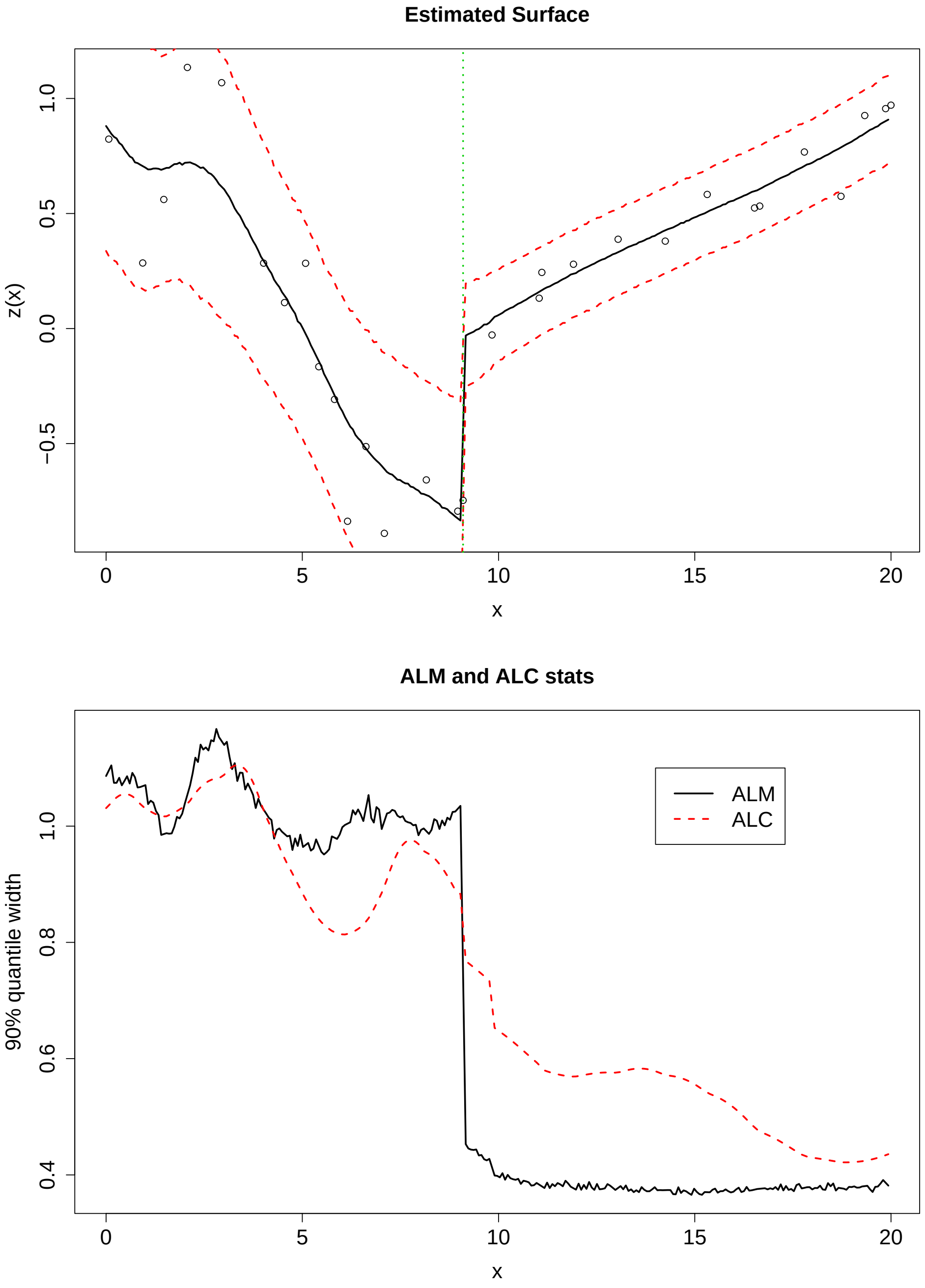


Figure 4.3: Sine data after 30 adaptively chosen samples. *Top:* posterior predictive mean and 90% quantiles, and MAP partition $\hat{\mathcal{T}}$. *Bottom:* ALM (black-solid) and ALC (red-dashed) statistics.

BAS recently learned that there is probably one partition near $x = 10$, with roughly the same number of samples on each side. The *bottom* of the figure shows that predictive uncertainty (under both ALM and ALC) is higher on the left side than on the right. ALM and ALC are in relative agreement, however the transition of ALC over the partition boundary is more smooth. The ALM statistics are "noisier" than ALC because the former is based on quantiles, and the latter on averages (4.2). Although both ALM and ALC are shown, only ALC was used to select adaptive samples.

Figure 4.4 shows a snapshot taken after 45 samples were gathered. BAS has sampled more heavily in the sinusoidal region, and learned a great deal. There are almost twice as many samples to the left of the partition, as compared to the right. ALM and ALC are in less agreement here than in Figure 4.3. In particular, they have different modes. Also, ALC is far less concerned with uncertainty near the partition boundary, than it is, say, near $x = 7$.

Finally, the snapshot in Figure 4.5 was taken after 97 samples had been gathered. By now, BAS has learned about the secondary cosine structure in the left-hand region. It has focused almost three-times more of its sampling effort to the left of the single partition in $\hat{\mathcal{T}}$, compared to the right. ALM and ALC both have high uncertainty near the partition boundary, but are otherwise in stark disagreement about where to sample next. ALM is larger everywhere on the left, than it is on the right. ALC has peaks on the left which are higher than on the right, but its valleys are lower.

In summary, the *top panels* of Figures 4.3, 4.4 & 4.5 track the the treed GP surrogate model's improvements in its ability to predict the mean, via the increase in resolution from one figure to the next. From the scale of y-axes on the *bottom* panels of the three figures, one can also see that as more samples are gathered, the variance in the posterior predictive distribution of the treed GP decreases as well.

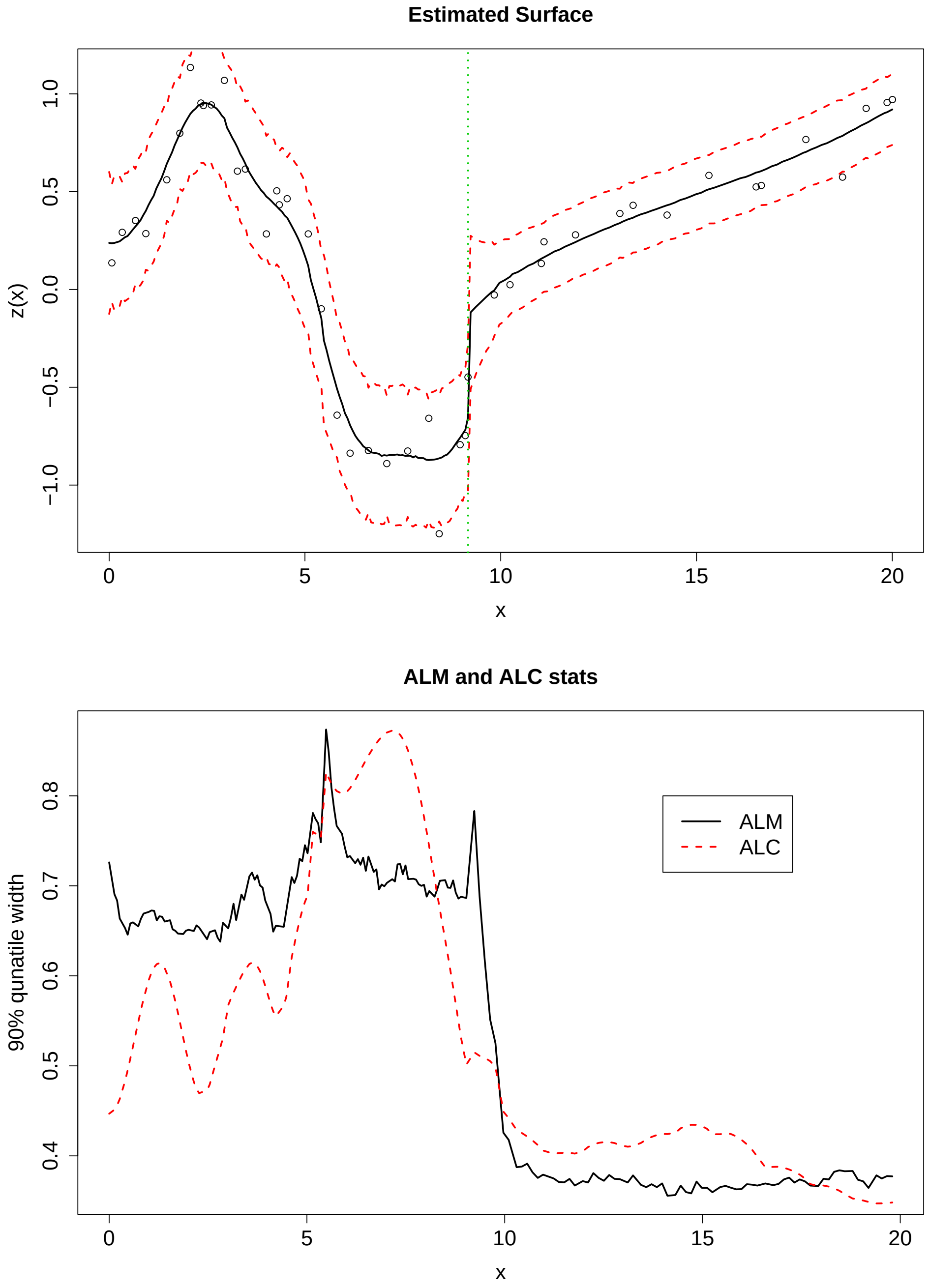


Figure 4.4: Sine data after 45 adaptively chosen samples. *Top:* posterior predictive mean and 90% quantiles, and MAP partition $\hat{\mathcal{T}}$. *Bottom:* ALM (black-solid) and ALC (red-dashed) statistics.

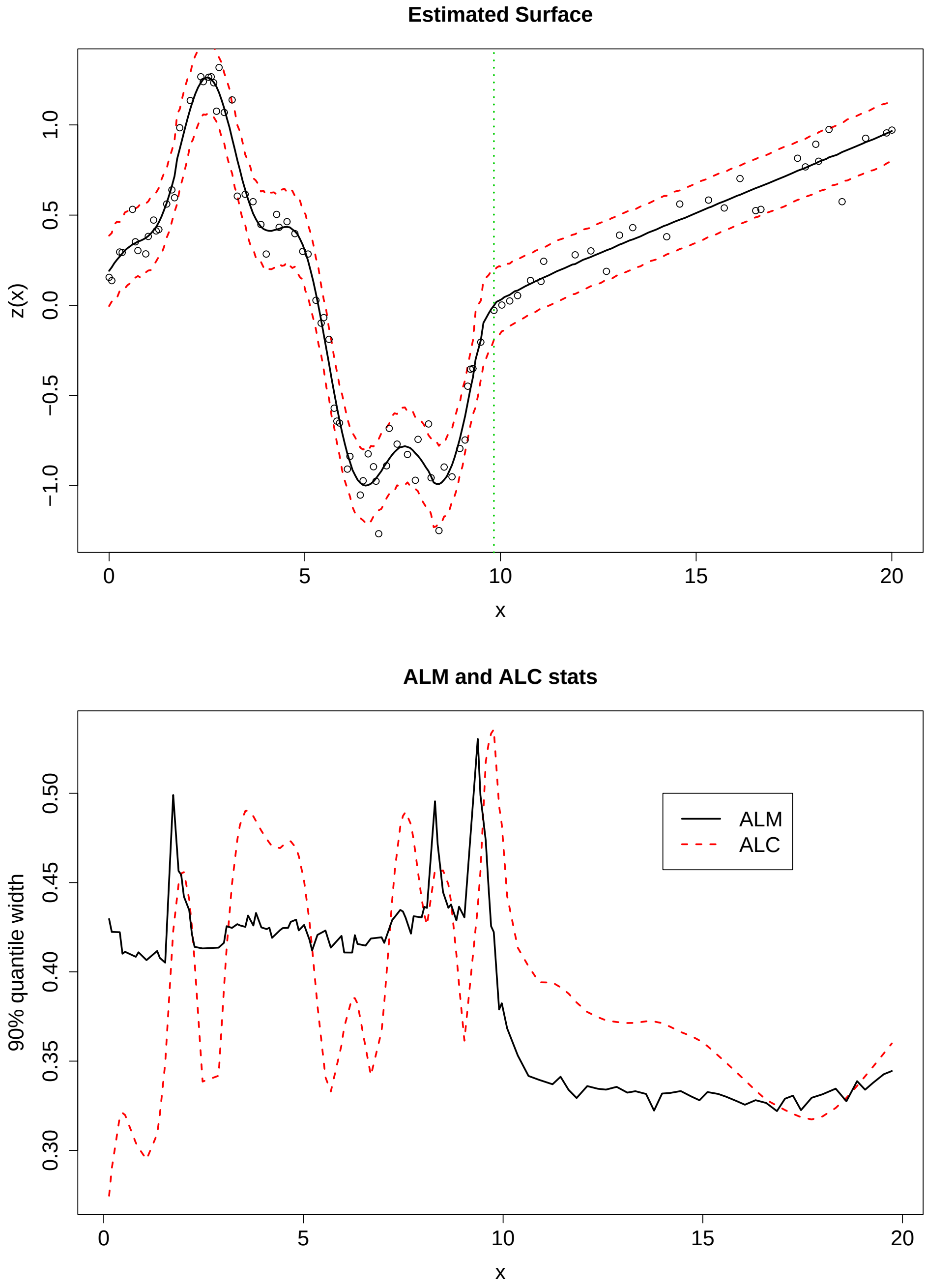


Figure 4.5: Sine data after 97 adaptively chosen samples. *Top:* posterior predictive mean and 90% quantiles, and MAP partition $\hat{\mathcal{T}}$. *Bottom:* ALM (black-solid) and ALC (red-dashed) statistics.

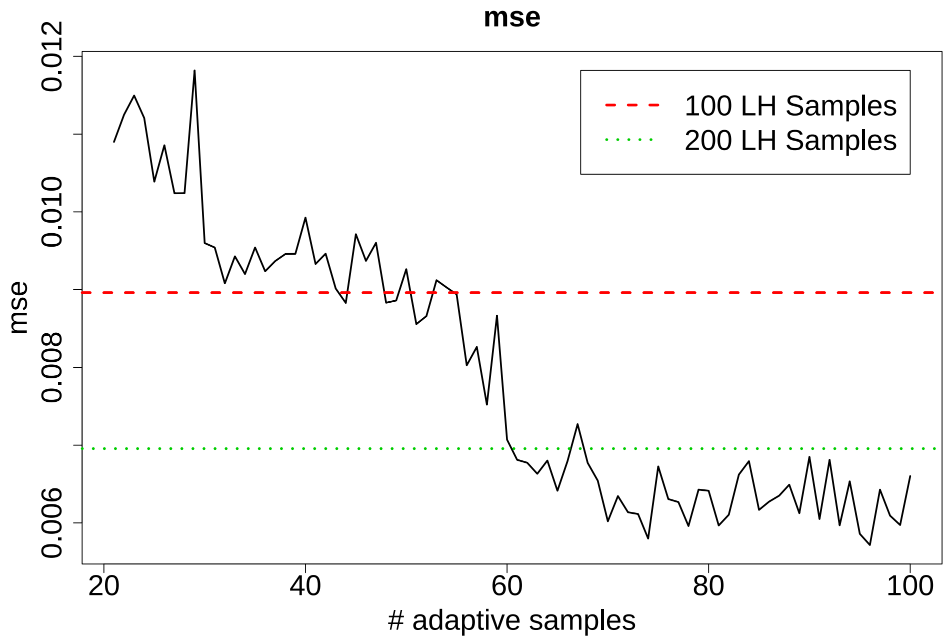


Figure 4.6: Mean-squared error (MSE) on the sinusoidal data compared to size 100 and 200 Latin Hypercube (LH) samples.

Despite the disagreements between ALM and ALC during the evolution of BAS, it is interesting to note that difference between using ALC and ALM on this data is negligible. This general theme will be noticed in other experiments as well. This is likely due to the high quality of candidates chosen using a treed D-optimal design. Treed D-optimal designs prevent the clumping behavior that tends to hurt ALM, but to which ALC is somewhat less prone.

Comparison

Perhaps the best illustration of how BAS learns and adapts over time is to compare it to something that is, ostensibly, less adaptive. Consider the plot in Figure 4.6. It shows mean-squared error (MSE) as a function of the size of the design (i.e. the number of adaptive samples). The plot shows that the MSE of BAS decreases steadily as samples are added, despite

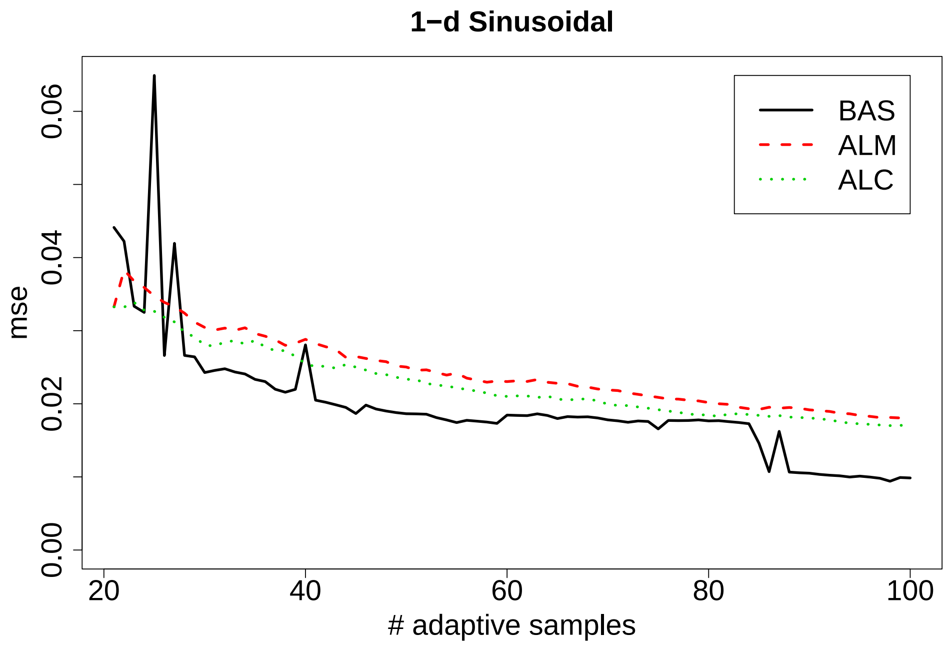


Figure 4.7: Mean-squared error (MSE) on the sinusoidal data using the ALM algorithm (labeled as BAS) compared to ALM and ALC with stationary GP's as in Seo et al. (2000).

the fact that fewer points are added in the linear region. Moreover it shows that BAS yields significantly lower MSE compared to using the same treed GP surrogate model with LH, rather than adaptive, sampling. In fact, the figure shows that adaptive sampling is at least two-times more efficient than LH sampling on this data.

Another constructive comparison is to show how BAS measures up against ALM and ALC, as implemented by Seo et al. (2000)—with a stationary GP surrogate model [see Section 1.2.4]. Seo et al. make the very powerful assumption that correct covariance structure is known at the start of sampling. Thus, the model need not be updated in light of new responses. Also, candidate locations $\mathbf{Y} = \tilde{\mathbf{X}}$ are taken to be the remaining unsampled locations from a predefined grid. Figure 4.7 shows an MSE plot comparing BAS to adaptive sampling with ALC and ALM based on an MAP parameterized—in hindsight—stationary GP model. The plot shows that as soon as BAS has enough samples to learn the partitioned covariance structure, it outperforms ALM and ALC based on a stationary model. Clearly, the treed GP is the right surrogate model for this dataset.

4.4.2 2-d Synthetic Exponential data

The nonstationary treed GP surrogate model has an even greater impact on adaptive sampling in a higher dimensional input space. For an illustration, consider again the synthetic exponential data from Sections 2.5.2 & 3.3.2. Figure 4.8 shows a snapshot after 30 adaptive samples have been gathered with BAS under the ALC algorithm. The *bottom* panel shows the single partition of $\hat{\mathcal{T}}$, with samples evenly split between the two regions. Room for improvement is evident in the mean predictive surface (*top* panel). Figure 4.9 shows surfaces for ALM (*top*) and ALC (*bottom*), which can be used as a guide to adaptive sampling.

After 72 adaptive samples have been selected, the situation is greatly improved. Figure 4.10 shows a posterior predictive surface with all the right ingredients: two partitions in $\hat{\mathcal{T}}$, and heavier sampling in the first quadrant compared to the rest of the input space. ALM and ALC, in Figure 4.11, agree that the first quadrant most interesting, although ALC is less confident. They disagree less about where, specifically, to sample next.

Finally, Figures 4.12 & 4.13 show a snapshot taken after the 123 adaptive samples. The predictive surface looks flawless. Almost 3/4 of the samples are located in the first quadrant which occupies only 1/4 of the total input space. The scale of the norm of predictive quantiles (ALM), in the *top* panel of Figure 4.13, is about half the height of those of the previous snapshot in Figure 4.11, everywhere except near the minima/maxima. ALM would choose to sample there next. ALC (*bottom*) shows that global uncertainty in the quadrant can be reduced so long as samples are taken in the interior of the region.

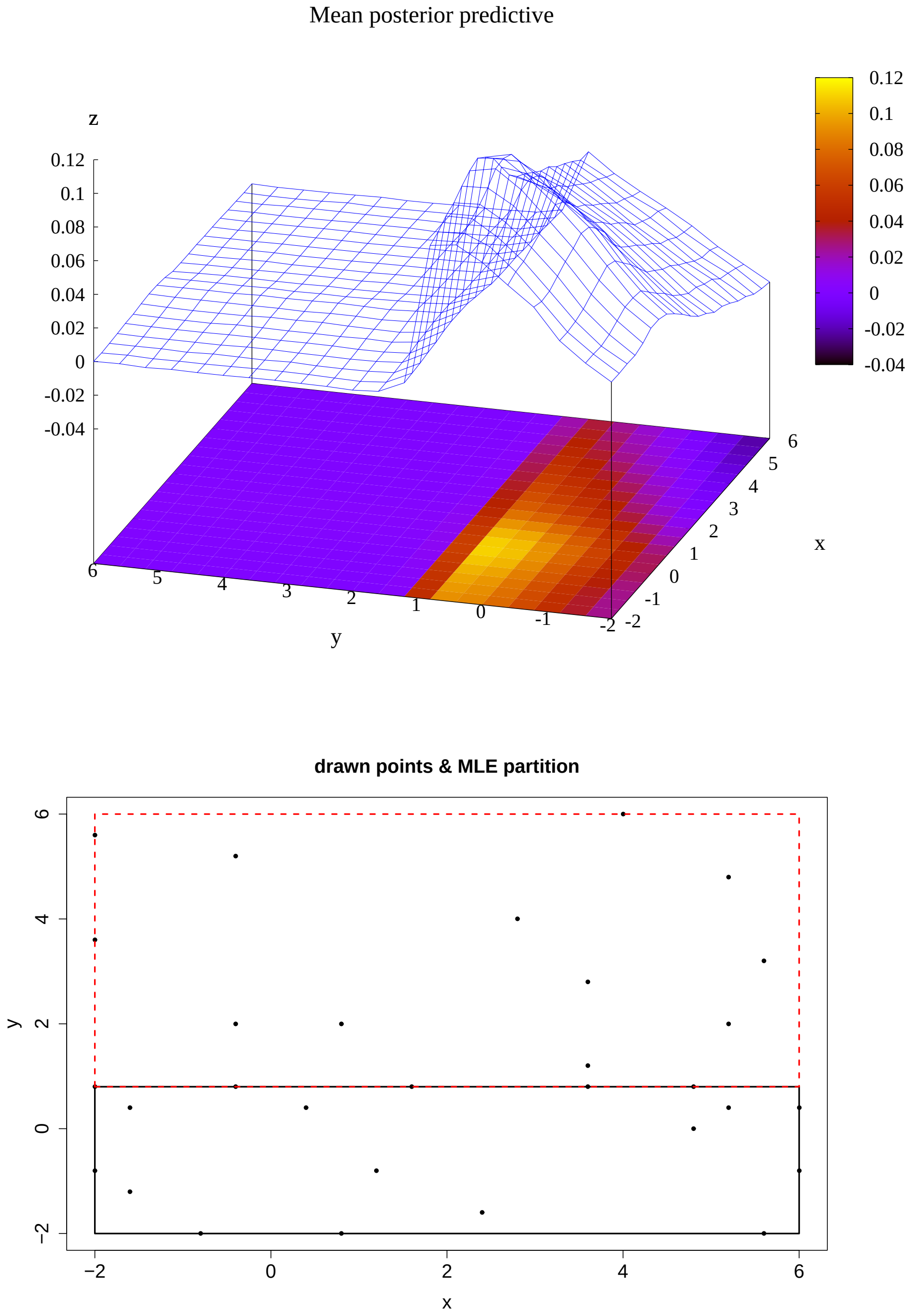
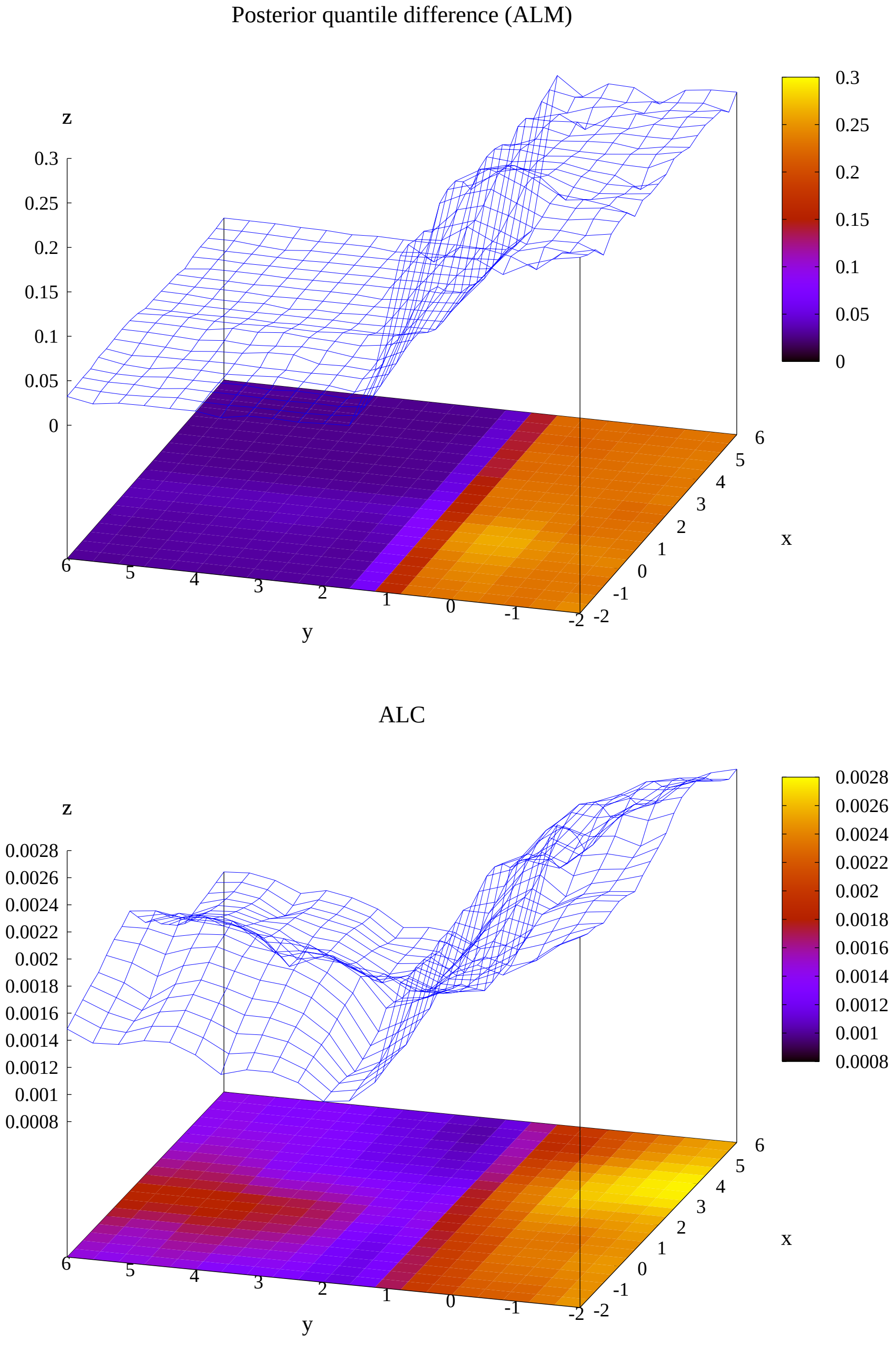


Figure 4.8: Exponential data after 30 adaptively chosen samples. *Top:* posterior predictive mean surface; *Bottom:* Sampled locations and MAP partition $\hat{\mathcal{T}}$.


Posterior quantile difference (ALM)

ALC

Figure 4.9: Continued from Figure 4.8 shows ALM *top* and ALC *bottom* surfaces after 30 adaptively chosen samples from the exponential data.

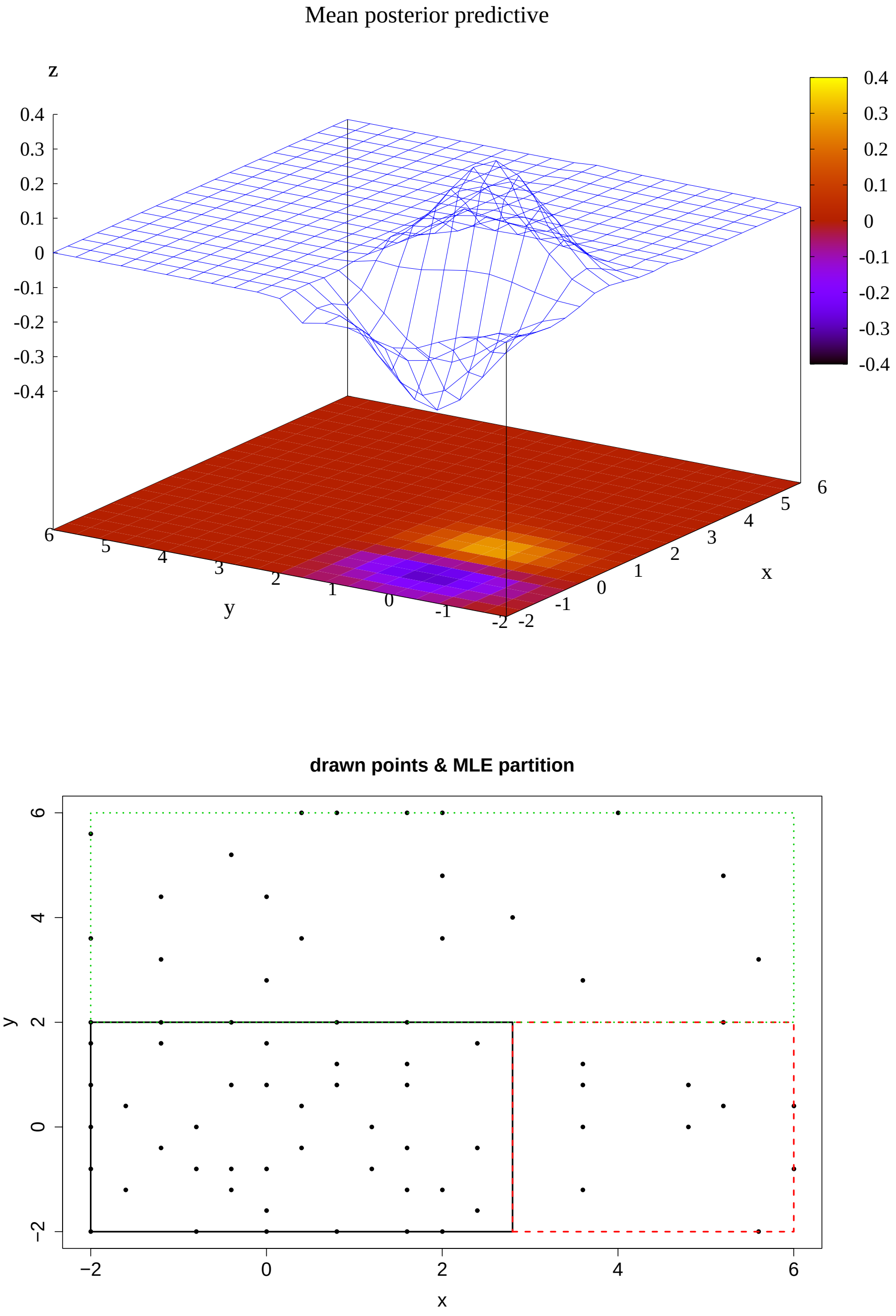
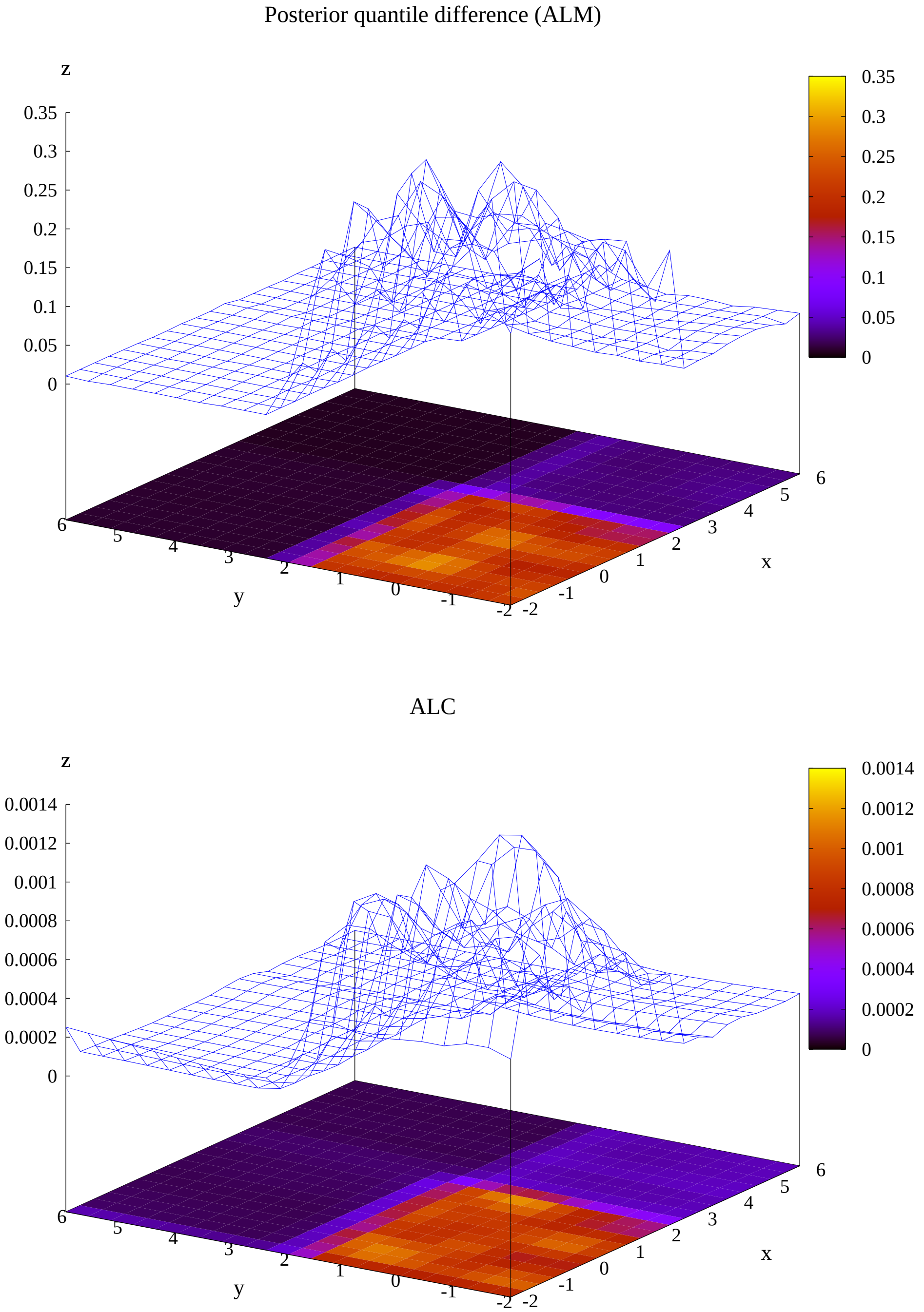


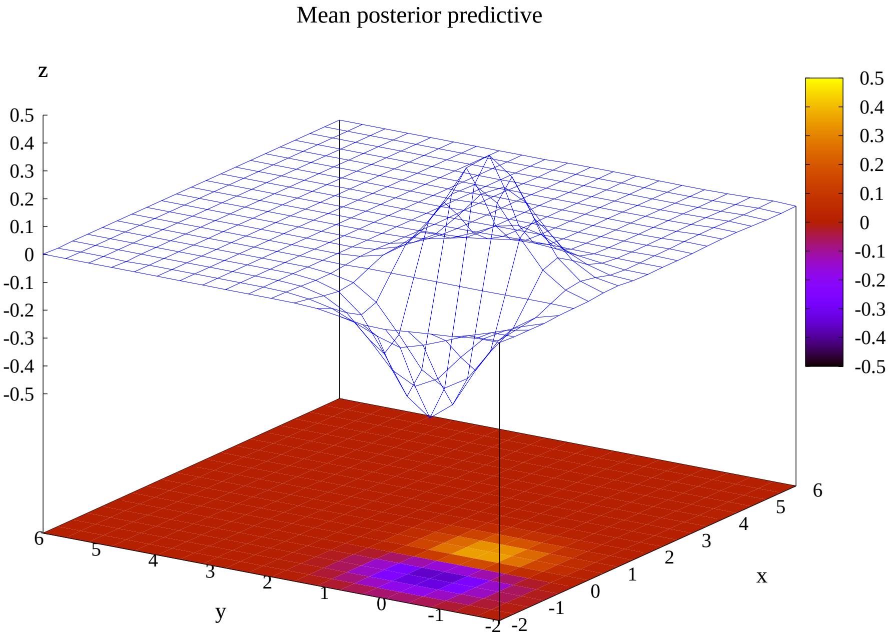
Figure 4.10: Exponential data after 72 adaptively chosen samples. *Top:* posterior predictive mean surface; *Bottom:* Sampled locations and MAP partition $\hat{\mathcal{T}}$.



Posterior quantile difference (ALM)

ALC

Figure 4.11: Continued from Figure 4.10 shows ALM *top* and ALC *bottom* surfaces after 72 adaptively chosen samples from the exponential data.



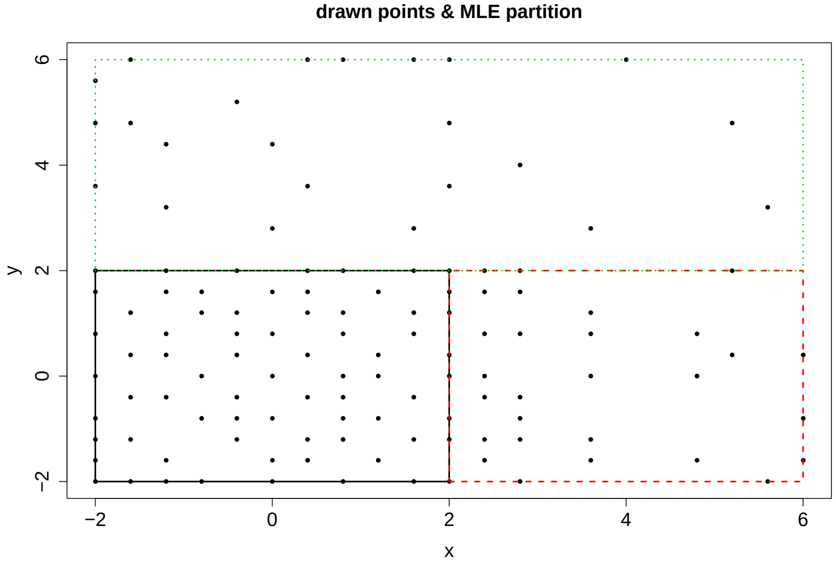
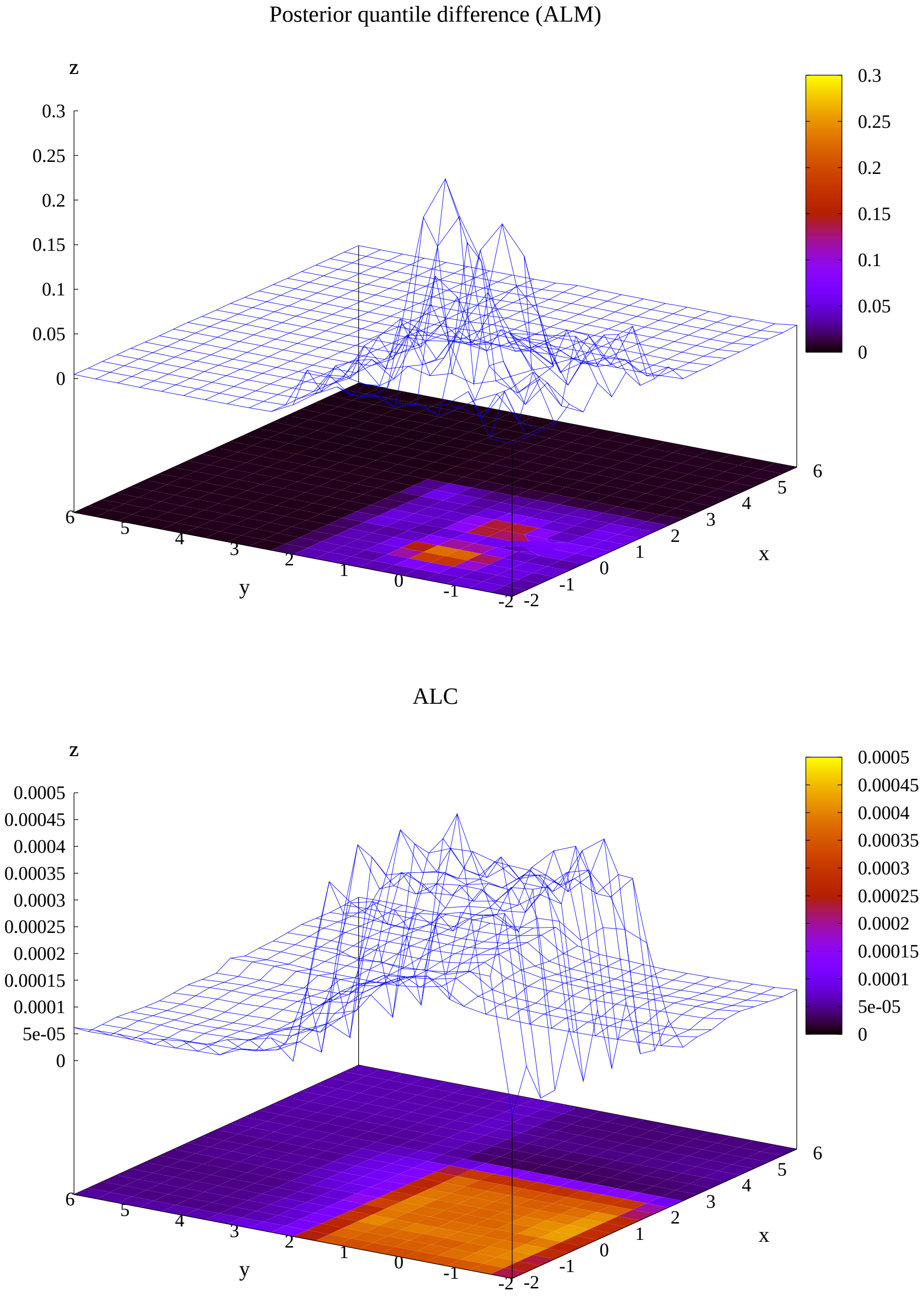


Figure 4.12: Exponential data after 123 adaptively chosen samples. *Top:* posterior predictive mean surface; *Bottom:* Sampled locations and MAP partition $\hat{\mathcal{T}}$.


Posterior quantile difference (ALM)

ALC

Figure 4.13: Continued from Figure 4.12 shows ALM *top* and ALC *bottom* surfaces after 123 adaptively chosen samples from the exponential data.

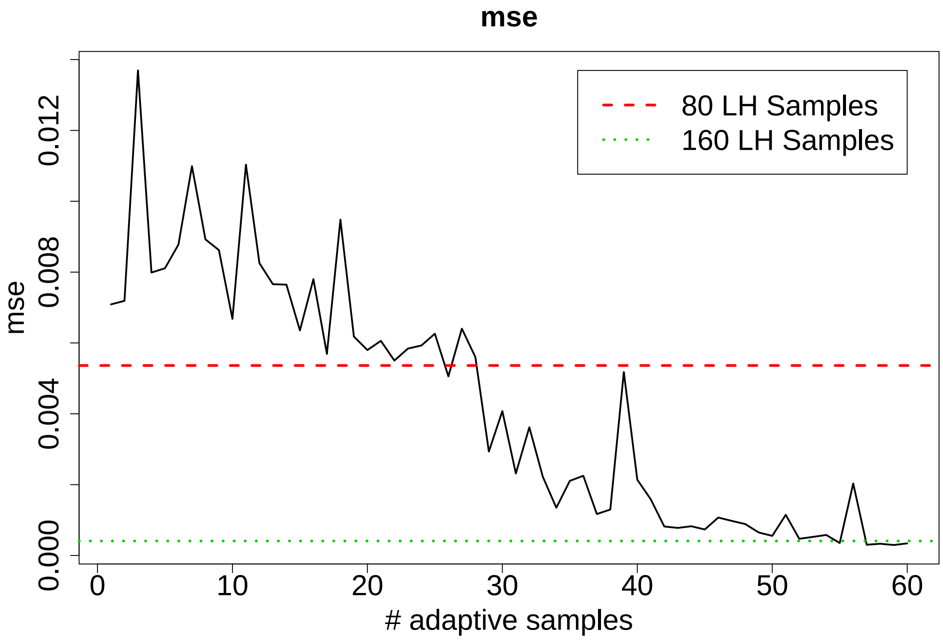


Figure 4.14: Mean-squared error (MSE) on the exponential data compared to size 100 and 200 Latin Hypercube (LH) samples.

As before, with the sinusoidal data of the previous section, despite the different recommendations made by ALM and ALC, the results are quite similar when ALM is used instead.

Comparison

As with the sinusoidal data of the last subsection, Figure 4.14 shows mean-squared error (MSE) as a function of the size of the design. Basically, the same conclusions can be drawn here: MSE of BAS decreases steadily as samples are added, despite that most of the sampling occurs in the first quadrant; adaptive sampling is at least two-times more efficient than LH sampling on this data.

Compared to the stationary GP implementation of ALC and ALM by Seo et al. (2000), BAS is again the winner. See Figure 4.15. Unlike the sinusoidal data, the exponential data is

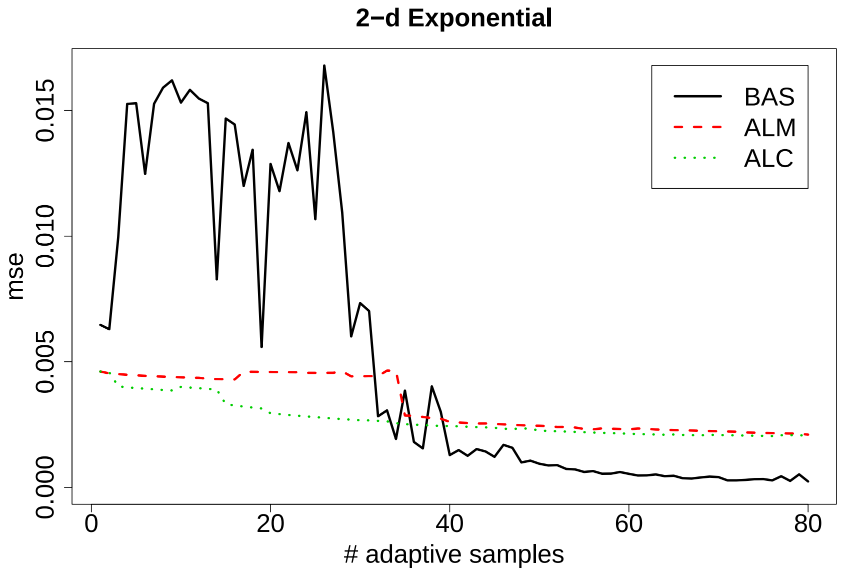


Figure 4.15: Mean-squared error (MSE) on the exponential data using the ALM algorithm (labeled as BAS) compared to ALM and ALC with stationary GP's as in Seo et al. (2000).

not defined by step functions. Transitions between partitions are more smooth. Thus it takes BAS longer to learn about $\mathcal{T}$, and the corresponding three GP models in each region of $\hat{\mathcal{T}}$. Once it does however—after about 50 samples—BAS outperforms the stationary model.

4.4.3 LGBB CFD Experiment

The final experiment is the motivating example for this work. It is the output from computational fluid dynamics simulations of a proposed reusable NASA launch vehicle, called the Langley Glide-Back Booster (LGBB). Simulations involved the integration of the inviscid Euler equations over a mesh of 1.4 million cells (0.8 million cells were used for the supersonic cases). A slice through some cells and the geometry of the LGBB is shown in Figure 4.16.

In the LGBB experiment, three input parameters are varied over (side slip angle, speed, and angle of attack), and for each setting of the input parameters, six outputs (lift,

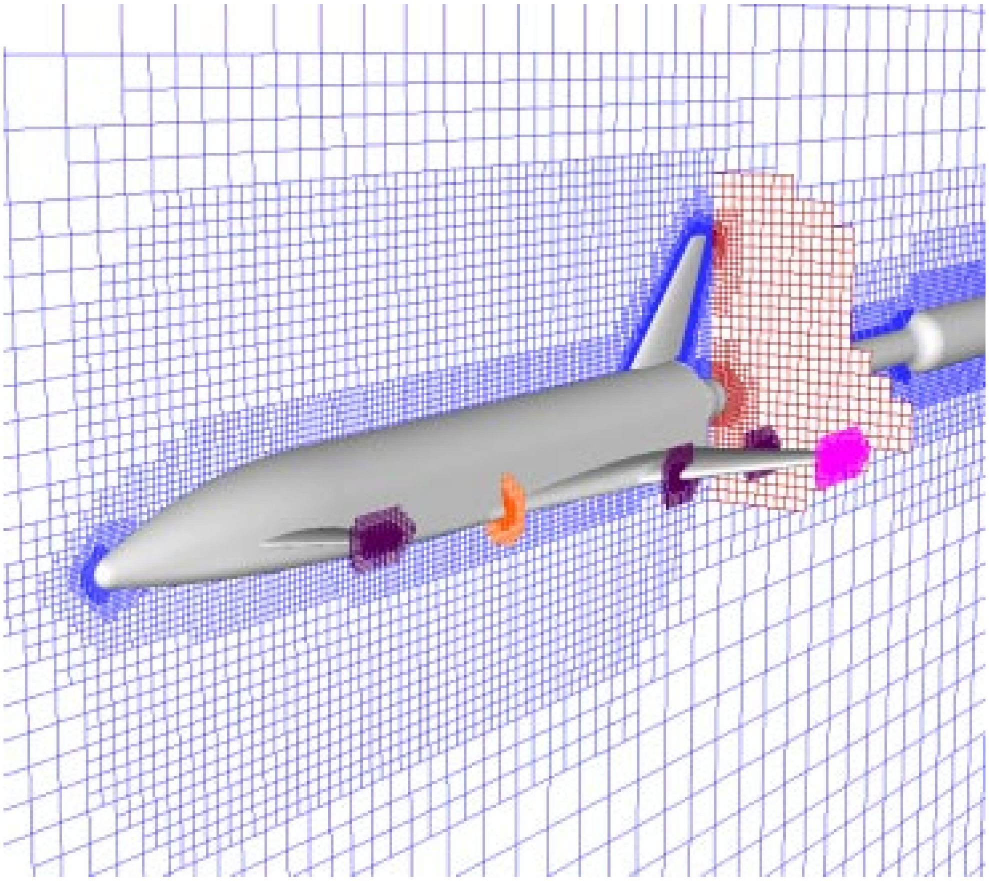


Figure 4.16: A slice through some cells and the geometry of the Langley Glide-Back Booster.

drag, pitch, side-force, yaw, and roll) are monitored. Each run of the Euler solver on an input triplet takes on the order of 5-20 hours on a high end workstation. All six responses are computed simultaneously. In a previous experiment, a supercomputer interface was used to launch runs at over 3,250 input configurations in several hand-crafted batches. The panels of Figure 1.1 [in Chapter 1] show plots of the resulting lift response as a function of Mach (speed) and alpha (angle of attack), with beta (side-slip angle) fixed to zero. A more detailed description of this system and its results are provided by Rogers et al. (2003). Some results therein will be summarized below, along with the comparisons to follow.

BAS for the LGBB is illustrated pictorially by the remaining figures in this section. The experiment was implemented on the NASA supercomputer `Columbia`—a fast and highly parallelized architecture, but with an extremely variable workload. The *emcee* algorithm of Section 4.1 was designed to interface with `AeroDB`, a database queuing system used by NASA to submit jobs to `Columbia`, and a set of CFD simulation codes called `cart3d`. To minimize impact on the queue, the *emcee* was restricted to ten submitted simulation jobs at a time. Candidate locations were sub-sampled from a 3-d grid consisting of 37,909 configurations.

Figure 4.17 shows the 780 configurations sampled by BAS for the LGBB experiment, in two projections. The *top* panel shows locations as a function of Mach (speed) and alpha (angle of attack), projecting over beta (side slip angle); the *bottom* panel shows Mach versus beta, projecting over alpha. NASA recommended restricting the number of possible beta settings to a handful, presumably to facilitate easy examination of the posterior surfaces in 2-d slices. The *top* panel in the figure shows that most of the configurations chosen by BAS were located near Mach one, with highest density for large alpha. Samples are scarce for Mach greater than two. The *bottom* panel shows uniform sampling across beta settings. A small amount of random noise has been added to the samples for visibility purposes.

After samples were gathered, the treed GP model was used in order to gather samples from the posterior predictive distribution at every location in the full 37,909 grid. Figure 4.18 shows a slice of the lift response, for beta $= 0$, plotted as a function of Mach and alpha. The *top* panel is a perspective and image plot, whereas the *bottom* panel shows results from the initial experiment for comparison. The plot in the *top* pane is more smooth, even though it is based on far fewer sampled locations. Figure 4.19 shows the sampled configurations and MAP tree $\hat{\mathcal{T}}$. The MAP partition separates out the near-Mach-one region. Samples are densely concentrated in this region—most heavily for large alpha.

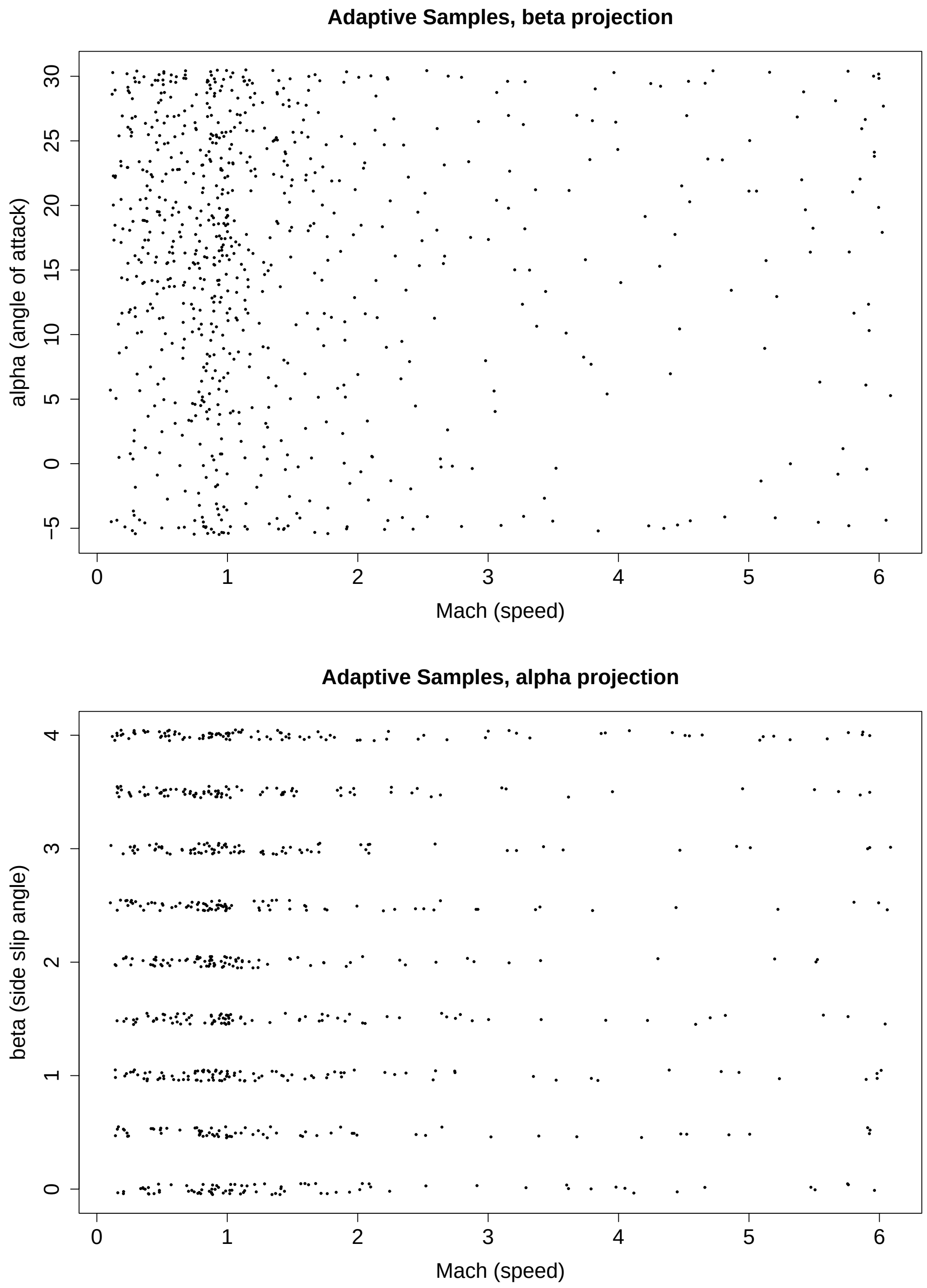


Figure 4.17: Full set of adaptively sampled configurations projected over beta (side-slip angle; *top*) and then over alpha (angle of attack; *bottom*).

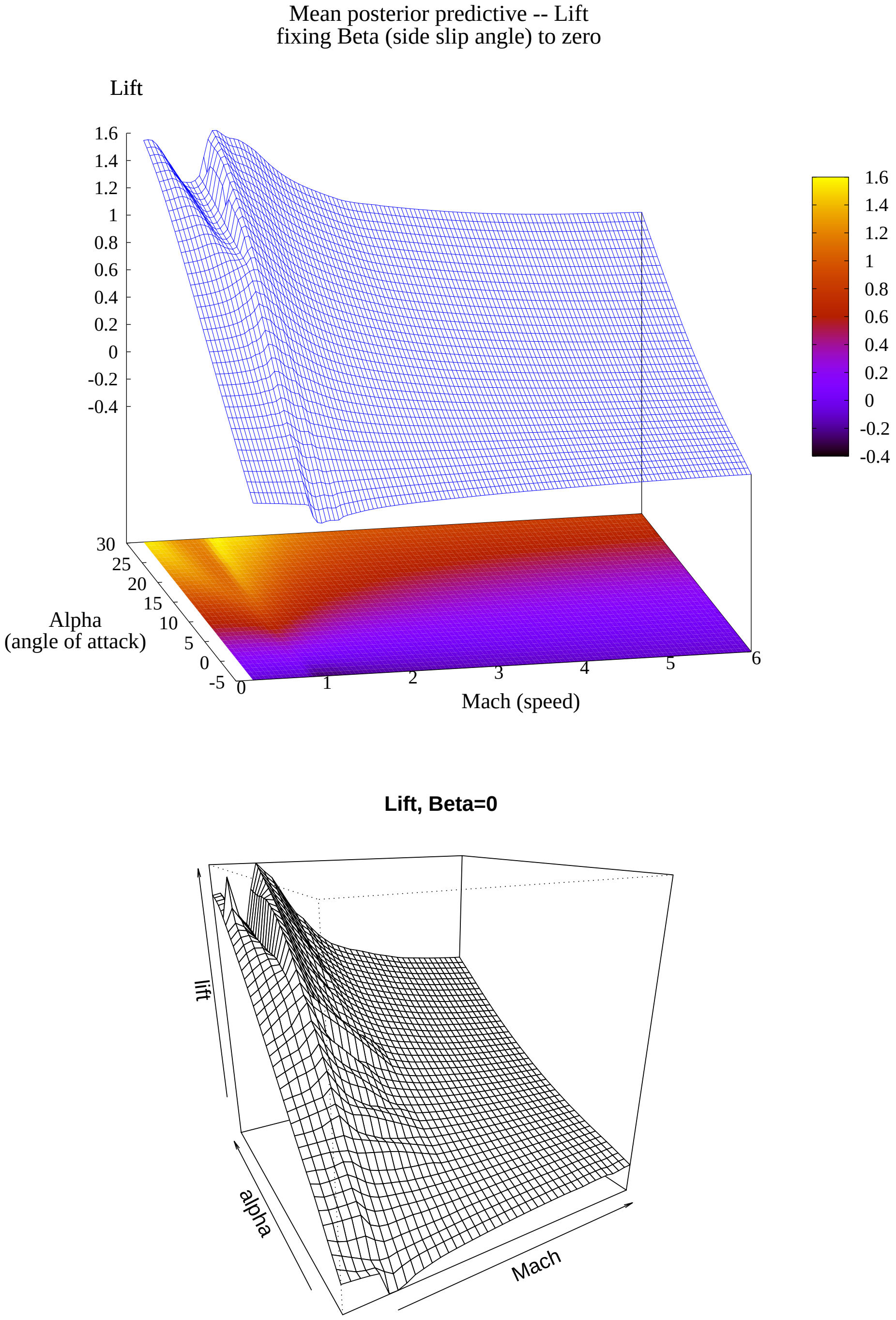


Figure 4.18: LGBB projection of the *lift* response plotted as a function of Mach (speed) and Alpha (angle of attach) with Beta (side slip angle) fixed at zero. *Top:* surface and image plot of mean posterior predictive surface; *bottom:* comparison to initial manually sampled experiment.

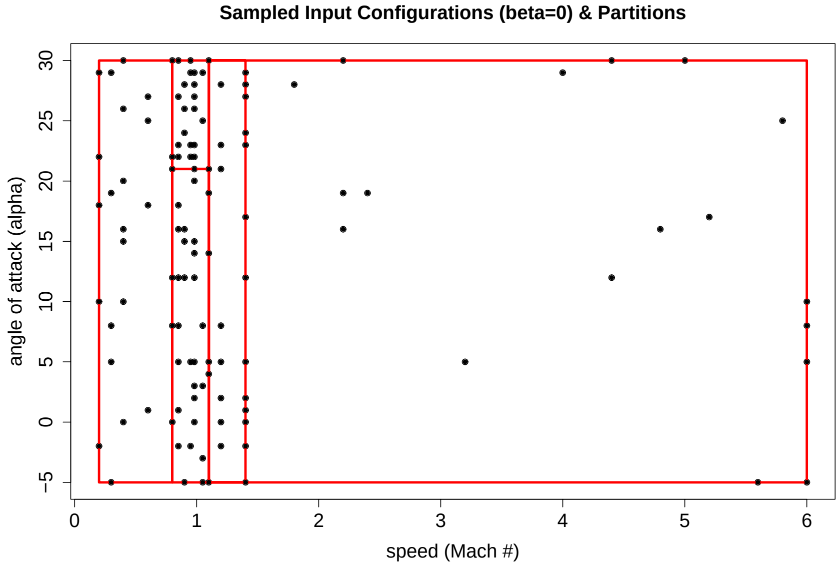
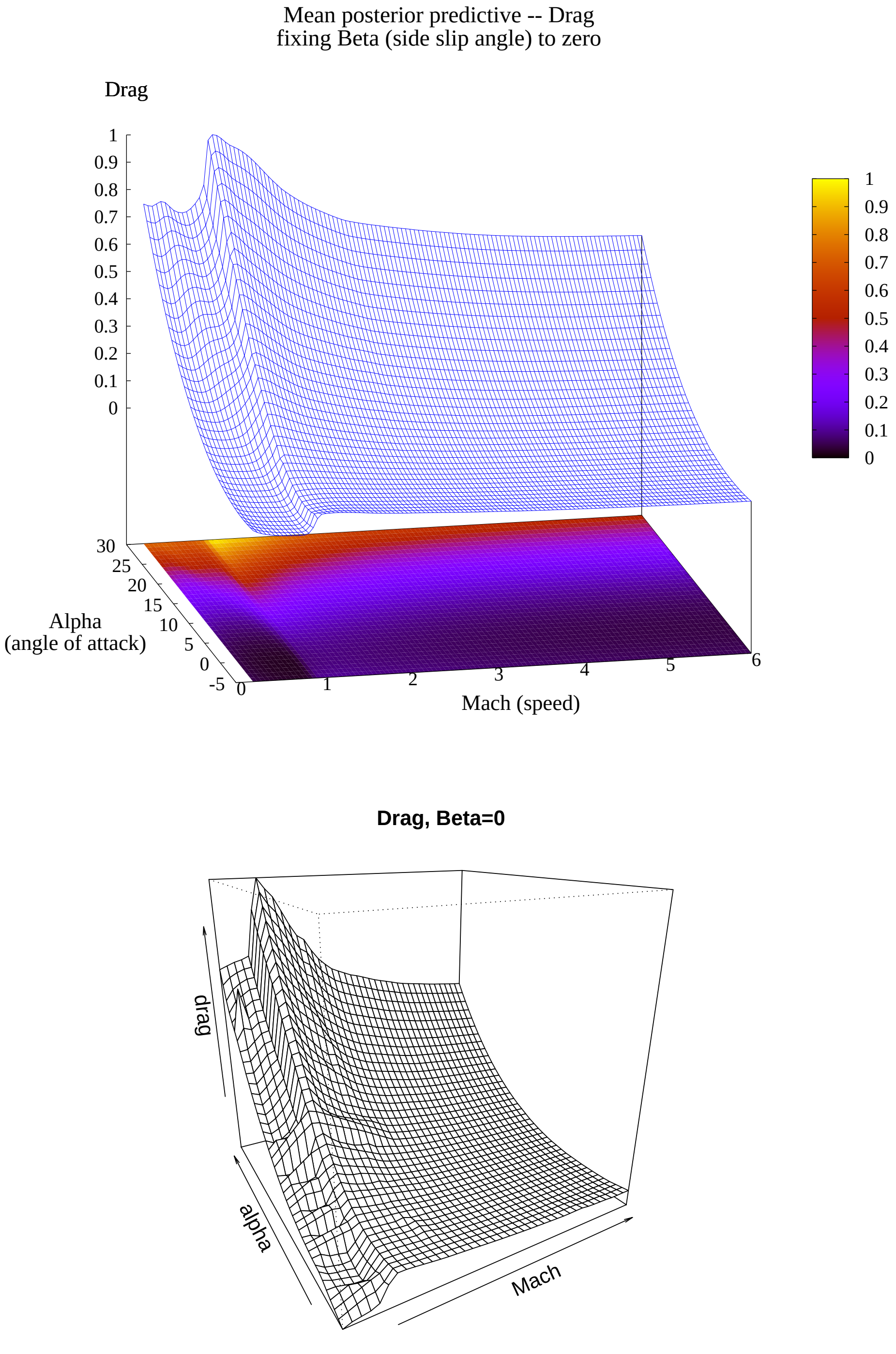


Figure 4.19: LGBB Beta = 0 slice of the *lift* adaptively sampled configurations and and MAP tree $\hat{\mathcal{T}}$.

Figures 4.20–4.24 show posterior predictive surfaces (*top*) and initial surfaces (*bottom*) for the remaining five responses. Drag and Pitch are shown for the beta = 0 slice. Other slices look strikingly similar. Side, yaw, and roll are shown for the beta = 2 slice, as beta = 0 slices for these responses are essentially zero. MAP partitions $\hat{\mathcal{T}}$ for these responses are similar to the ones shown in Figure 4.18, for the lift response. A common theme in these figures is that the posterior predictive surfaces are far smoother than comparative initial runs. There are three reasons for this. (1) The treed GP model has an explicit noise component, i.e., the nugget, included specifically to help in smoothing; (2) the posterior predictive distribution was produced by the nonlinear nonstationary treed GP model at the full grid of 37,909 candidates, whereas the initial run could only be interpolated linearly; and finally (3) the CFD codes used to evaluate the initial run were somewhat enhanced before building the adaptive design.


Mean posterior predictive -- Drag
fixing Beta (side slip angle) to zero

Drag, Beta=0

Figure 4.20: LGBB slice of the *drag* response plotted as a function of Mach (speed) and Alpha (angle of attach) with Beta (side slip angle) fixed at zero. *Top:* surface and image plot of mean posterior predictive surface; *bottom:* comparison to initial manually sampled experiment.

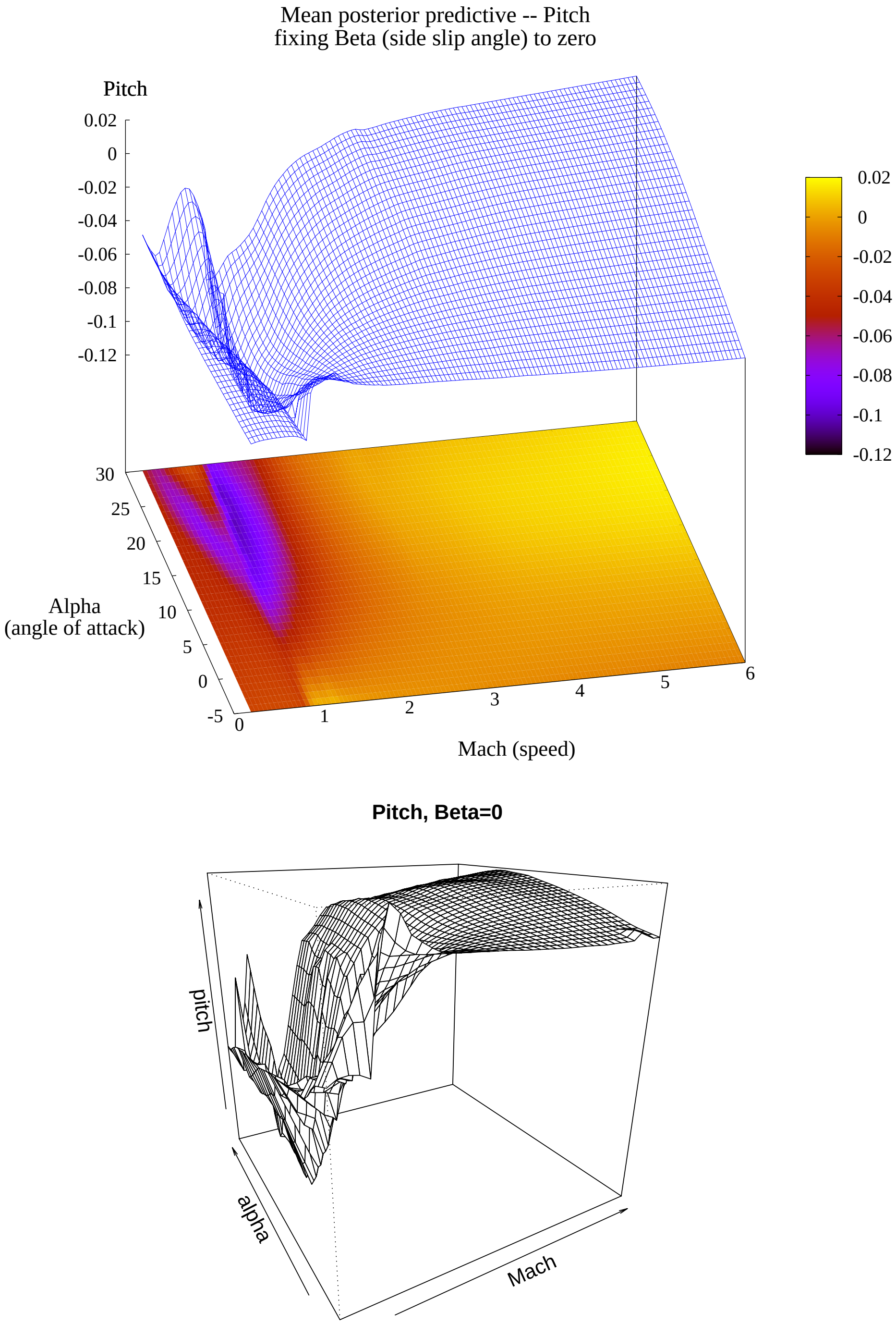


Figure 4.21: LGBB slice of the *pitch* response plotted as a function of Mach (speed) and Alpha (angle of attach) with Beta (side slip angle) fixed at zero. *Top:* surface and image plot of mean posterior predictive surface; *bottom:* comparison to initial manually sampled experiment.

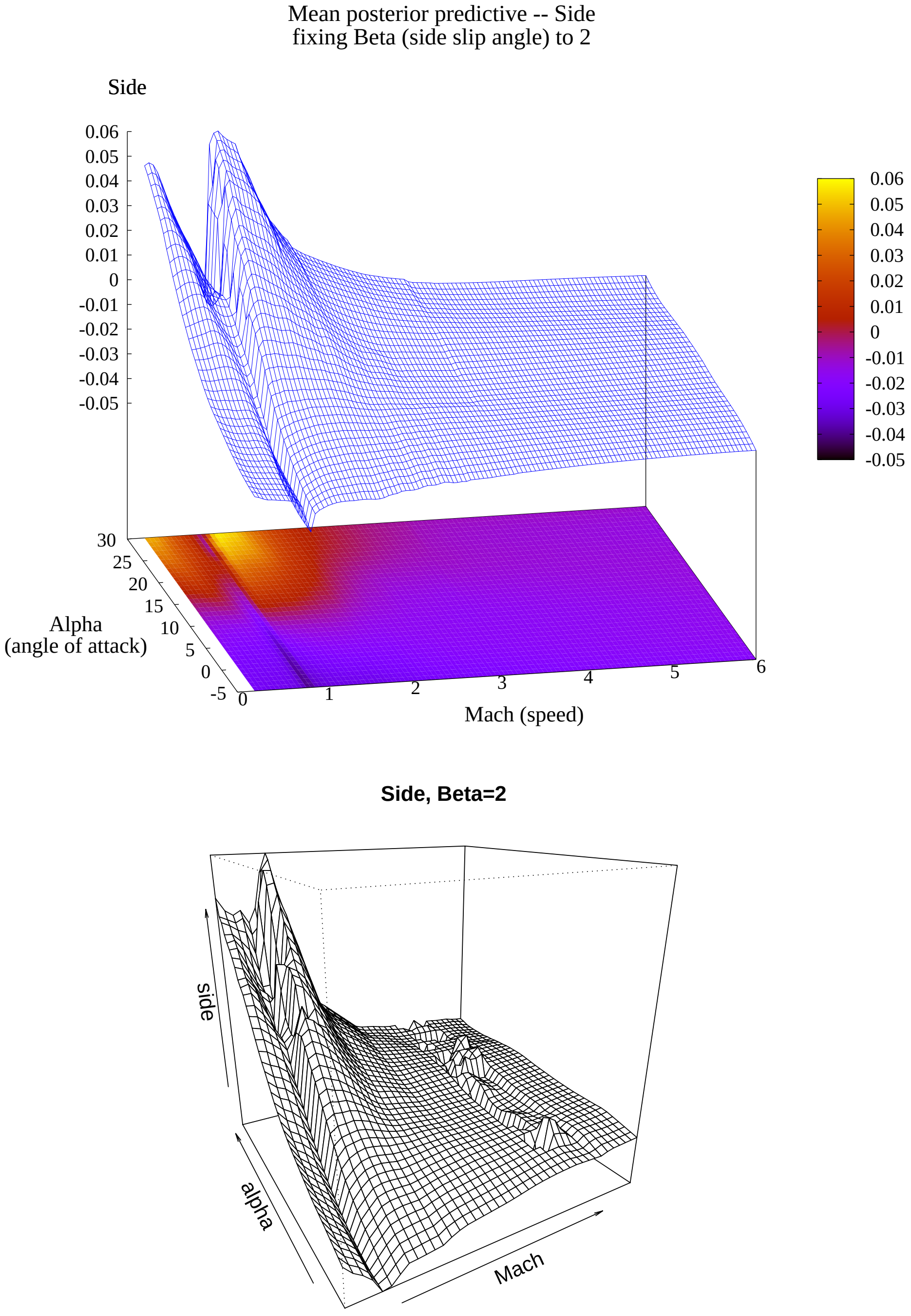


Figure 4.22: LGBB slice of the *side* response plotted as a function of Mach (speed) and Alpha (angle of attach) with Beta (side slip angle) fixed at 2. *Top:* surface and image plot of mean posterior predictive surface; *bottom:* comparison to initial manually sampled experiment.

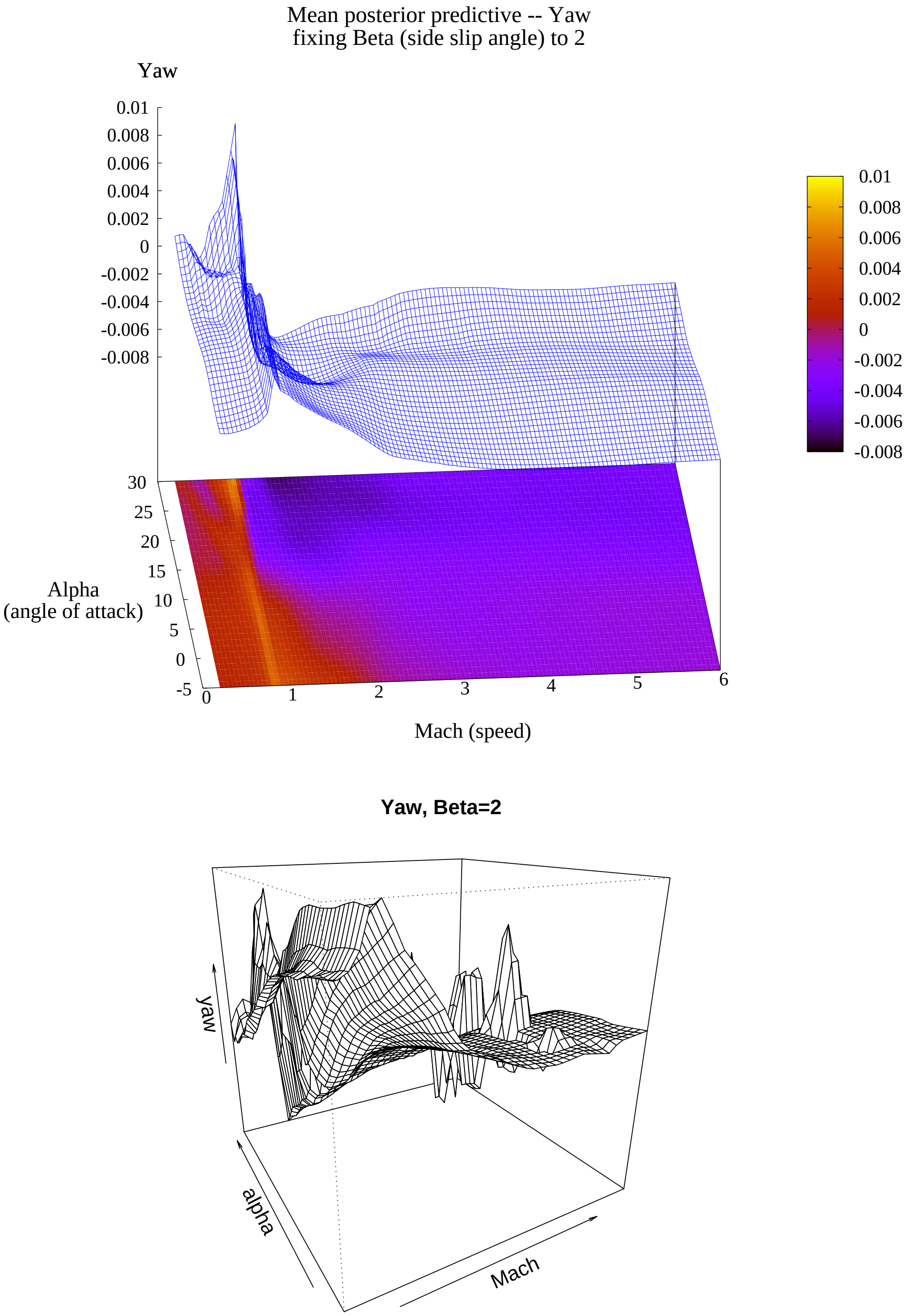


Figure 4.23: LGBB slice of the *yaw* response plotted as a function of Mach (speed) and Alpha (angle of attach) with Beta (side slip angle) fixed at 2. *Top:* surface and image plot of mean posterior predictive surface; *bottom:* comparison to initial manually sampled experiment.

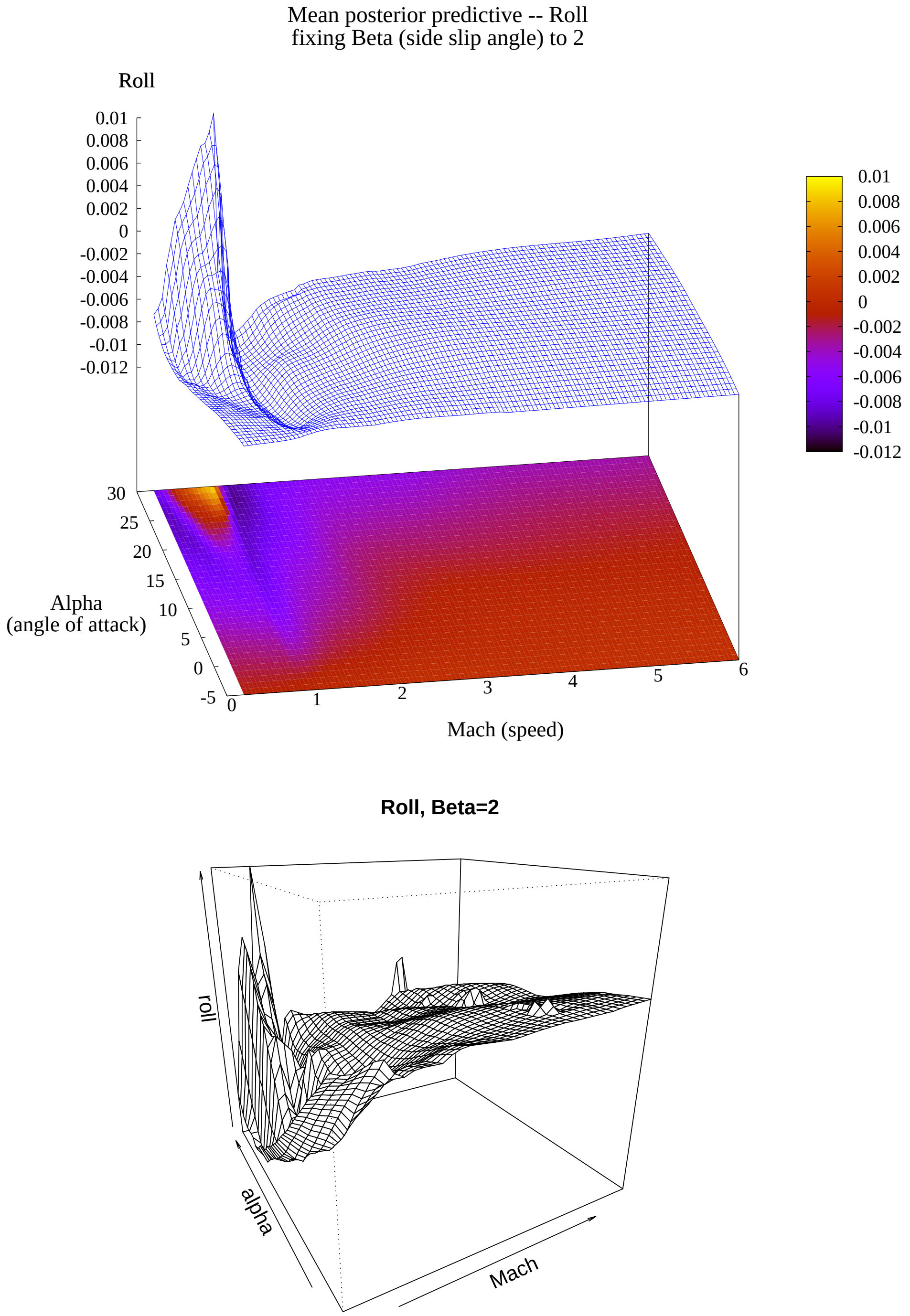


Figure 4.24: LGBB slice of the *roll* response plotted as a function of Mach (speed) and Alpha (angle of attach) with Beta (side slip angle) fixed at 2. *Top:* surface and image plot of mean posterior predictive surface; *bottom:* comparison to initial manually sampled experiment.

4.5 Conclusion

This chapter showed how the treed GP (and GP LLM) can be used as a surrogate model in the sequential design of computer experiments. A hybrid approach, combining Machine Learning and classical design methodologies, was taken in order to develop a flexible system for use in the highly variable environment of asynchronous agent-based supercomputing. In other words, a flexible and opportunistic approach was taken, rather than strictly "optimal" one.

Two sampling algorithms were proposed as adaptations to similar techniques developed for a simpler class of models. One chooses to sample configurations with high posterior predictive variance (ALM); the other uses a criteria based on an average global reduction in uncertainty (ALC). These model uncertainty statistics were used to determine which of a set optimally spaced candidate locations should go for simulation next. Optimal candidate designs were determined by adapting a classic optimal design methodology to Bayesian partition models. The result is a highly efficient Bayesian adaptive sampling strategy, representing an improvement on the state-of-the-art of computer experiment methodology at NASA.

Bayesian adaptive sampling (BAS) was illustrated on two nonstationary synthetic data sets. Finally, BAS was implemented on a supercomputer at NASA in order to sequentially design the computer experiment for a proposed reusable launch vehicle, called the Langley Glide-Back Booster (LGBB). With fewer than one-quarter of the samples, BAS was able to produce superior response surfaces by sampling more heavily in the interesting, or challenging, parts of the input space, relying on treed GP model to fill in the gaps in less-sampled regions.

Chapter 5

Conclusion

The novelty in the work presented here is in combining, enhancing, and exploiting established techniques from a number of different communities. The motivating NASA computer experiment provided an interesting problem that had not been addressed anywhere in the literature. The need for an efficient nonstationary model, and a flexible interface for sequentially designing an experiment to be run on a modern supercomputer, naturally led to a sampling and enhancing of existing techniques. The resulting models and methods, I believe, are truly unique and should be of general interest to the communities out of which the initial ideas were born.

The main contributions of this thesis were divided up into three chapters (2,3, & 4). Chapter 2 introduced the treed Gaussian process (GP) model as a nonparametric extension of the Bayesian Linear CART model. The need for such a model was motivated through illustration on synthetic and real data. Chapter 3 exploited the limiting linear model (LLM) parameterization of the GP, and showed how it can be both useful and accessible in terms of Bayesian posterior estimation and prediction. The benefits include speed, parsimony, and a relatively straightforward implementation of a semiparametric model. Together with treed partitioning, the result was a uniquely nonstationary, semiparametric, tractable, and highly accurate regression tool.

Chapter 4 showed show the treed GP (and GP LLM) could be used as a surrogate model in the sequential design of experiments. Creating a surrogate model for computer experiments is a problem that will continue to be of interest as additional computing resources are put toward more accurate simulations rather than faster results. The Bayesian approach allows a natural mechanism for building a sequential design based on the current estimated uncertainty. A hybrid approach to sequential experimental design was taken in order to develop a flexible system for use in the highly variable environment of asynchronous agent-based supercomputing. So called Bayesian adaptive sampling (BAS) was illustrated on synthetic data and on the motivating NASA experiment which involved computationally expensive computational fluid dynamics codes.

5.1 Future work

Not unlike many theses—though complete in most respects—there is always more work that can be done. In general, I look forward to future collaborations with research labs like NASA who have interesting experiments to run. It seems that many such collaborations should be on the horizon as computer experiments become more and more commonplace as surrogates for costly physical experimentation, as computing becomes more and more distributed and asynchronous, and as simulation codes become more and more complex.

Several small enhancements would increase the visibility and usability of the work presented here. As mentioned in Chapter 2, an interface in `R` to the treed GP (and GP LLM) code has been developed. I look forward to releasing an `R` package in the near future.

Some multi-platform issues need addressing—particularly in terms of linear algebra libraries—and documentation needs to be written. To my observation there is currently a void in the industry—that will hopefully soon be filled—for tools which implement nonstationary, fully-Bayesian, model fitting, inference, forecasting, and design, that are easy to use, efficient, and free. A well-packaged treed GP implementation with sampling libraries should go a long way towards filling that void.

Simple statistical analysis of the experimental apparatus, i.e., the supercomputer, may improve adaptive sampling. Accurate forecasts of how many agents will complete their runs before the next adaptive sampling trial could be used to tune the size of candidate designs. Configurations which are likely start running before the next round can be incorporated into the design ahead of time, with mean-predictive responses as surrogates, so that future candidates can be focused on other parts of the input space. Initial experiments on the NASA supercomputer suggest that accurate forecasts may be possible with something like an autoregressive Poisson process model.

For the longer term there are some enhancements which can be made towards applying the methods of this thesis to a broader array of problems. Three such related problems are of sampling to find extrema (Schonlau, 1997; Jones et al., 1998; Huang et al., 2005b), to find contours (Ranjan, 2005) generally, or to find boundaries, i.e., contours with large gradients (Banerjee & Gelfand, 2005), a.k.a. Wombling. Another problem is that of learning about, or finding extrema in, computer experiments with multi-fidelity codes of varying execution costs (Huang et al., 2005a), or those which are paired with a *physical* experiment (Reese et al., 2005).

Finally, an interesting undertaking would be to explore how the hierarchical structure of treed models can be exploited for fitting multi-resolution data (Ferreira et al., 2005), or as a modeling tool for sharing information about parameters at the leaves of the tree, across partitions. One idea might be to allow configurations which lie on the the boundary between two regions to contribute to inference on models in both regions, for example, in order to encourage continuity in the predictive surface between regions.

Appendix A

Estimating Parameters: Details

The following sections show full derivations of conditional and marginalized posteriors of the parameters to the Gaussian processes at the leaves of the tree.

A.1 Full Conditionals

$\boxed{\boldsymbol{\beta}:}$

$$
\begin{aligned}
& p(\boldsymbol{\beta}_\nu|\text{rest}) \\
& \propto p(\mathbf{Z}_\nu|\boldsymbol{\beta}_\nu, \sigma^2_\nu, d_\nu, g_\nu) p(\boldsymbol{\beta}_\nu|\boldsymbol{\beta}_0, \sigma^2_\nu, \tau^2_\nu, \mathbf{W}) \\
& = N(\mathbf{Z}_\nu|\mathbf{F}_\nu\boldsymbol{\beta}_\nu, \sigma^2_\nu\mathbf{K}_\nu) \cdot N(\boldsymbol{\beta}_\nu|\boldsymbol{\beta}_0, \sigma^2_\nu\tau^2\mathbf{W}) \\
& \propto \exp\left\{\frac{-1}{2\sigma^2_\nu}\left[(\mathbf{Z}_\nu - \mathbf{F}_\nu\boldsymbol{\beta}_\nu)'\mathbf{K}_\nu^{-1}(\mathbf{Z}_\nu - \mathbf{F}_\nu\boldsymbol{\beta}_\nu) + (\boldsymbol{\beta}_\nu - \boldsymbol{\beta}_0)'\,\frac{\mathbf{W}^{-1}}{\tau^2_\nu}\,(\boldsymbol{\beta}_\nu - \boldsymbol{\beta}_0)\right]\right\} \\
& \propto \exp\left\{-\frac{1}{2\sigma^2_\nu}\left[-2\mathbf{Z}'_\nu\mathbf{K}_\nu^{-1}\mathbf{F}_\nu\boldsymbol{\beta}_\nu + \boldsymbol{\beta}'_\nu\mathbf{F}'_\nu\mathbf{K}_\nu^{-1}\mathbf{F}_\nu\boldsymbol{\beta}_\nu + \boldsymbol{\beta}'_\nu\frac{\mathbf{W}^{-1}}{\tau^2_\nu}\boldsymbol{\beta}_\nu - 2\boldsymbol{\beta}'_\nu\frac{\mathbf{W}^{-1}}{\tau^2_\nu}\boldsymbol{\beta}_0\right]\right\} \\
& = \exp\left\{-\frac{1}{2\sigma^2_\nu}\left[\boldsymbol{\beta}'_\nu(\mathbf{F}'_\nu\mathbf{K}_\nu^{-1}\mathbf{F}_\nu + \mathbf{W}^{-1}/\tau^2)\boldsymbol{\beta}_\nu - 2\boldsymbol{\beta}'_\nu(\mathbf{F}'_\nu\mathbf{K}_\nu^{-1}\mathbf{Z}_\nu + \mathbf{W}^{-1}\boldsymbol{\beta}_0/\tau^2)\right]\right\}
\end{aligned}
$$

giving

$$\boxed{\boldsymbol{\beta}_\nu|\text{rest} \sim N(\tilde{\boldsymbol{\beta}}_\nu, \sigma_\nu^2 \mathbf{V}_{\tilde{\beta}_\nu})} \tag{A.1}$$

where

$$\mathbf{V}_{\tilde{\beta}_\nu} = (\mathbf{F}_\nu' \mathbf{K}_\nu^{-1} \mathbf{F}_\nu + \mathbf{W}^{-1}/\tau^2)^{-1} \qquad \tilde{\boldsymbol{\beta}}_\nu = \mathbf{V}_{\tilde{\beta}_\nu}(\mathbf{F}_\nu' \mathbf{K}_\nu^{-1} \mathbf{Z}_\nu + \mathbf{W}^{-1}\boldsymbol{\beta}_0/\tau^2),$$

which can be sampled using the Gibbs algorithm.

$\boxed{\boldsymbol{\beta}_0\text{:}}$

$$\begin{aligned}
&p(\boldsymbol{\beta}_0|\text{rest})\\
&= p(\boldsymbol{\beta}|\boldsymbol{\beta}_0, \sigma^2, \tau^2, \mathbf{W})p(\boldsymbol{\beta}_0)\\
&= p(\boldsymbol{\beta}_0)\prod_{i=1}^{r} p(\boldsymbol{\beta}_\nu|\boldsymbol{\beta}_0, \sigma_\nu^2, \tau_\nu^2, \mathbf{W})\\
&= N(\boldsymbol{\beta}_0|\mu, \mathbf{B})\prod_{i=1}^{r} N(\boldsymbol{\beta}_\nu|\boldsymbol{\beta}_0, \sigma_\nu^2\tau_\nu^2\mathbf{W})\\
&\propto \exp\left\{-\frac{1}{2}(\boldsymbol{\beta}_0-\mu)'\mathbf{B}^{-1}(\boldsymbol{\beta}_0-\mu)\right\}\prod_{i=1}^{r}\exp\left\{-\frac{1}{2\sigma_\nu^2\tau_\nu^2}(\boldsymbol{\beta}_\nu-\boldsymbol{\beta}_0)'\mathbf{W}^{-1}(\boldsymbol{\beta}_\nu-\boldsymbol{\beta}_0)\right\}\\
&= \exp\left\{-\frac{1}{2}\left[(\boldsymbol{\beta}_0-\mu)'\mathbf{B}^{-1}(\boldsymbol{\beta}_0-\mu)+\sum_{i=1}^{r}\frac{1}{\sigma_\nu^2\tau_\nu^2}(\boldsymbol{\beta}_\nu-\boldsymbol{\beta}_0)'\mathbf{W}^{-1}(\boldsymbol{\beta}_\nu-\boldsymbol{\beta}_0)\right]\right\}\\
&\propto \exp\left\{-\frac{1}{2}\left[\boldsymbol{\beta}_0'\mathbf{B}^{-1}\boldsymbol{\beta}_0 - 2\boldsymbol{\beta}_0'\mathbf{B}^{-1}\mu + \boldsymbol{\beta}_0'\mathbf{W}^{-1}\sum_{i=1}^{r}\frac{\boldsymbol{\beta}_0}{\sigma_\nu^2\tau_\nu^2} - 2\boldsymbol{\beta}_0'\mathbf{W}^{-1}\sum_{i=1}^{r}\frac{\boldsymbol{\beta}_\nu}{\sigma_\nu^2\tau_\nu^2}\right]\right\}\\
&\propto \exp\left\{-\frac{1}{2}\left[\boldsymbol{\beta}_0'\left(\mathbf{B}^{-1}+\sum_{i=0}^{r}\frac{\mathbf{W}^{-1}}{\sigma_\nu^2\tau_\nu^2}\right)\boldsymbol{\beta}_0 - 2\boldsymbol{\beta}_0'\left(\mathbf{B}^{-1}\mu + \mathbf{W}^{-1}\sum_{i=1}^{r}\frac{\boldsymbol{\beta}_\nu}{\sigma_\nu^2\tau_\nu^2}\right)\right]\right\}
\end{aligned}$$

giving

$$\boxed{\boldsymbol{\beta}_0|\text{rest} \sim N(\tilde{\boldsymbol{\beta}}_0, \boldsymbol{V}_{\tilde{\beta}_0})} \tag{A.2}$$

where

$$\mathbf{V}_{\tilde{\beta}_0} = \left(\mathbf{B}^{-1} + \mathbf{W}^{-1} \sum_{i=0}^{r} (\sigma_\nu \tau_\nu)^{-2} \right)^{-1} \qquad \tilde{\boldsymbol{\beta}}_0 = \mathbf{V}_{\tilde{\beta}_0} \left(\mathbf{B}^{-1}\mu + \mathbf{W}^{-1} \sum_{i=1}^{r} \boldsymbol{\beta}_\nu (\sigma_\nu \tau_\nu)^{-2} \right).$$

which can be sampled using Gibbs.

$\boxed{\tau^2:}$

$$\begin{aligned}
p(\tau_\nu^2|\text{rest}) &= p(\boldsymbol{\beta}_\nu|\boldsymbol{\beta}_0, \sigma_\nu^2, \tau_\nu^2, \mathbf{W}) p(\tau_\nu^2) \\
&= N(\boldsymbol{\beta}_\nu|\boldsymbol{\beta}_0, \sigma^2\tau_\nu^2\mathbf{W}) IG(\tau_\nu^2|\alpha_\tau/2, q_\tau/2) \\
&\propto (2\pi)^{-m/2} (\tau_\nu^2)^{-\frac{m}{2}} |\mathbf{W}|^{-1} \exp\left\{ -\frac{(\boldsymbol{\beta}_\nu - \boldsymbol{\beta}_0)^\top \mathbf{W}^{-1} (\boldsymbol{\beta}_\nu - \boldsymbol{\beta}_0)}{2\sigma_\nu^2\tau_\nu^2} \right\} \times \\
&\qquad \frac{(q_\tau/2)^{\alpha_\tau/2}}{\Gamma(\alpha_\tau/2)} (\tau_\nu^2)^{-(\alpha_\tau/2+1)} \exp\left\{ -\frac{q_\tau}{2\tau^2} \right\} \\
&\propto (\tau_\nu^2)^{-(\frac{\alpha_\tau+m}{2}+1)} \exp\left\{ -\frac{q_\tau + (\boldsymbol{\beta}_\nu - \boldsymbol{\beta}_0)^\top \mathbf{W}^{-1} (\boldsymbol{\beta}_\nu - \boldsymbol{\beta}_0)/\sigma_\nu^2}{2\tau_\nu^2} \right\}
\end{aligned}$$

which means

$$\boxed{\tau_\nu^2 \sim IG\left(\frac{\alpha_\tau + m}{2}, \frac{q_\tau + (\boldsymbol{\beta}_\nu - \boldsymbol{\beta}_0)^\top \mathbf{W}^{-1} (\boldsymbol{\beta}_\nu - \boldsymbol{\beta}_0)/\sigma_\nu^2}{2} \right)} \qquad \text{(A.3)}$$

$\boxed{\mathbf{W^{-1}}\text{:}}$

$$\begin{aligned}
&p(\mathbf{W}^{-1}|\text{rest})\\
&= p(\mathbf{W})p(\boldsymbol{\beta}|\boldsymbol{\beta}_0, \sigma^2, \tau_2, \mathbf{W})\\
&= W(\mathbf{W}^{-1}|(\rho\mathbf{V})^{-1}, \rho) \cdot \prod_{i=1}^{r} N(\boldsymbol{\beta}_\nu|\boldsymbol{\beta}_0, \sigma_\nu^2\tau_\nu^2\mathbf{W})\\
&\propto |\mathbf{W}^{-1}|^{(\rho-m-1)/2} \exp\left\{-\frac{1}{2}\text{tr}((\rho\mathbf{V})\mathbf{W}^{-1})\right\} \times\\
&\qquad |\mathbf{W}^{-1}|^{r/2} \exp\left\{-\frac{1}{2}\sum_{i=1}^{r}\frac{1}{\sigma_\nu^2}(\beta_\nu-\beta_0)'\mathbf{W}^{-1}(\beta_\nu-\beta_0)\}\right\}\\
&= |\mathbf{W}^{-1}|^{(\rho+r-m-1)/2}\times\\
&\quad \exp\left\{-\frac{1}{2}\left[\text{tr}((\rho\mathbf{V})\mathbf{W}^{-1}) + \text{tr}\left(\sum_{i=1}^{r}\frac{1}{\sigma_\nu^2\tau_\nu^2}(\beta_\nu-\beta_0)'\mathbf{W}^{-1}(\beta_\nu-\beta_0)\right)\right]\right\}
\end{aligned}$$

obtained because a scalar is equal to its trace. Applying more properties of the trace operation gives

$$\begin{aligned}
&p(\mathbf{W}^{-1}|\text{rest})\\
&\propto |\mathbf{W}^{-1}|^{\frac{\rho+r-k-1}{2}} \exp\left\{-\frac{1}{2}\left[\text{tr}\left(\left(\rho\mathbf{V}+\sum_{i=1}^{r}\frac{(\beta_\nu-\beta_0)(\beta_\nu-\beta_0)'}{\sigma_\nu^2\tau_\nu^2}\right)\mathbf{W}^{-1}\right)\right]\right\}
\end{aligned}$$

which means

$$\boxed{\mathbf{W}^{-1}|\text{rest} \sim W\left(\rho\mathbf{V}+\sum_{i=1}^{r}\frac{1}{\sigma_\nu^2\tau_\nu^2}(\beta_\nu-\beta_0)(\beta_\nu-\beta_0)', \rho+r\right)} \tag{A.4}$$

and that Gibbs sampling is appropriate.

A.2 Marginalized Conditional Posteriors

Complete conditional posteriors for the parameters to the correlation function $K(\cdot,\cdot)$ can be obtained by analytically integrating out $\boldsymbol{\beta}$ and σ^2 to get a marginal posterior.

$$
\begin{aligned}
&p(\mathbf{K}|\mathbf{Z},\boldsymbol{\beta}_0,\mathbf{W},\tau^2)\\
&=\prod_\nu p(\mathbf{K}_\nu|\mathbf{Z}_\nu,\boldsymbol{\beta}_0,\tau^2,\mathbf{W})\\
&\propto\prod_\nu\int\int p(\mathbf{Z}_\nu|d_\nu,g_\nu,\boldsymbol{\beta}_\nu,\sigma_\nu^2)p(\mathbf{K}_\nu,\boldsymbol{\beta}_\nu,\sigma_\nu^2|\boldsymbol{\beta}_0,\tau_\nu^2,\mathbf{W})\;d\boldsymbol{\beta}_\nu\;d\sigma_\nu^2\\
&=\prod_\nu p(\mathbf{K}_\nu)\int p(\sigma_\nu^2)\int p(\mathbf{Z}_\nu|d_\nu,g_\nu,\boldsymbol{\beta}_\nu,\sigma_\nu^2)p(\boldsymbol{\beta}_\nu|\sigma_\nu^2,\boldsymbol{\beta}_0,\tau_\nu^2,\mathbf{W})\;d\boldsymbol{\beta}_\nu\;d\sigma_\nu^2\\
&=\prod_\nu p(\mathbf{K}_\nu)\int p(\sigma_\nu^2)\int N(\boldsymbol{\beta}_\nu|\tilde{\boldsymbol{\beta}}_\nu,\sigma_\nu^2\mathbf{V}_{\tilde{\beta}_\nu})\;d\boldsymbol{\beta_\nu}\\
&\qquad\qquad\times(2\pi)^{-\frac{n_\nu}{2}}\sigma_\nu^{-n_\nu}|\mathbf{K}_\nu|^{-\frac{1}{2}}\tau_\nu^{-m}|\mathbf{W}|^{-\frac{1}{2}}|\mathbf{V}_{\tilde{\beta}_\nu}|^{\frac{1}{2}}\\
&\qquad\qquad\times\exp\left\{-\frac{1}{2\sigma_\nu^2}\left[\mathbf{Z}_\nu'\mathbf{K}^{-1}\mathbf{Z}_\nu+\boldsymbol{\beta}_0'\mathbf{W}^{-1}\boldsymbol{\beta}_0/\tau^2-\tilde{\boldsymbol{\beta}}_\nu'\mathbf{V}_{\tilde{\beta}_\nu}^{-1}\tilde{\boldsymbol{\beta}}_\nu)\right]\right\}\;d\sigma^2.\\
&=\prod_\nu p(\mathbf{K}_\nu)\times\left(\frac{|\mathbf{V}_{\tilde{\beta}_\nu}|}{(2\pi)^{n_\nu}\tau_\nu^{2m}|\mathbf{K}_\nu||\mathbf{W}|}\right)^{\frac{1}{2}}\int\sigma_\nu^{-n_\nu}p(\sigma_\nu^2)\exp\left\{-\frac{\psi_\nu}{2\sigma_\nu^2}\right\}\;d\sigma_\nu^2,
\end{aligned}
$$

where

$$
\psi_\nu=\mathbf{Z}_\nu'\mathbf{K}^{-1}\mathbf{Z}_\nu+\boldsymbol{\beta}_0'\mathbf{W}^{-1}\boldsymbol{\beta}_0/\tau_\nu^2-\tilde{\boldsymbol{\beta}}_\nu'\mathbf{V}_{\tilde{\beta}_\nu}^{-1}\tilde{\boldsymbol{\beta}}_\nu. \tag{A.5}
$$

Expanding the prior for σ_ν^2 gives:

$$\begin{aligned}
&p(\mathbf{K}_\nu|\mathbf{Z},\boldsymbol{\beta}_0,\tau^2,\mathbf{W})\\
&\propto \prod_\nu p(\mathbf{K}) \times \left(\frac{|\mathbf{V}_{\tilde{\beta}_\nu}|}{(2\pi)^{n_\nu}\tau_\nu^{2m}|\mathbf{K}_\nu||\mathbf{W}|}\right)^{\frac{1}{2}}\\
&\qquad \times \int (\sigma_\nu^2)^{-\frac{n_\nu}{2}} \frac{\left(\frac{q_\sigma}{2}\right)^{\frac{\alpha_\sigma}{2}}}{\Gamma(\frac{\alpha_\sigma}{2})}(\sigma_\nu^2)^{-(\frac{\alpha_\sigma}{2}+1)} \exp\left\{-\frac{q_\sigma}{2\sigma_\nu^2}\right\} \exp\left\{-\frac{\psi_\nu}{2\sigma_\nu^2}\right\}\ d\sigma_\nu^2\\
&= \prod_\nu p(\mathbf{K}_\nu) \times \left(\frac{|\mathbf{V}_{\tilde{\beta}_\nu}|}{(2\pi)^{n_\nu}\tau_\nu^{2m}|\mathbf{K}_\nu||\mathbf{W}|}\right)^{\frac{1}{2}} \times \frac{\left(\frac{q_\sigma}{2}\right)^{\frac{\alpha_\sigma}{2}}}{\Gamma(\frac{\alpha_\sigma}{2})} \times \frac{\Gamma(\frac{\alpha_\sigma+n_\nu}{2})}{(\frac{q_\sigma+\psi_\nu}{2})^{\frac{\alpha_\sigma+n_\nu}{2}}}\\
&\qquad \times \int \frac{(\frac{q_\sigma+\psi_\nu}{2})^{\frac{\alpha_\sigma+n_\nu}{2}}}{\Gamma(\frac{\alpha_\sigma+n_\nu}{2})}(\sigma_\nu^2)^{-(\frac{\alpha_\sigma+n_\nu}{2}+1)} \times \exp\left\{-\frac{q_\sigma+\psi_\nu}{2\sigma_\nu^2}\right\}\ d\sigma_\nu^2,
\end{aligned}$$

since the integrand above is really $IG((\alpha_\sigma+n_\nu)/2,(q_\sigma+\psi_\nu)/2)$, the integral evaluates to 1, giving:

$$\begin{aligned}
&p(\mathbf{K}|\mathbf{Z},\boldsymbol{\beta}_0,\tau^2,\mathbf{W}) \\
&\propto \prod_\nu p(\mathbf{K}_\nu) \times \left(\frac{|\mathbf{V}_{\tilde{\beta}_\nu}|}{(2\pi)^{n_\nu}\tau_\nu^{2m}|\mathbf{K}_\nu||\mathbf{W}|}\right)^{\frac{1}{2}} \times \frac{\left(\frac{q_\sigma}{2}\right)^{\frac{\alpha_\sigma}{2}}}{\left(\frac{q_\sigma+\psi_\nu}{2}\right)^{\frac{\alpha_\sigma+n_\nu}{2}}} \times \frac{\Gamma\left(\frac{\alpha_\sigma+n_\nu}{2}\right)}{\Gamma\left(\frac{\alpha_\sigma}{2}\right)}.
\end{aligned} \tag{A.6}$$

Eq. (A.6) can be used in place of the likelihood of the data conditional on all parameters. It can be thought of as a likelihood of the data, conditional on only the parameterization of $K(\cdot,\cdot)$. When computing a Metropolis-Hastings acceptance ratio for proposed $\mathbf{K}_\nu$ in a particular region r_ν, it suffices to use only the terms in (A.6) which contain some function of the imputed correlation matrix $\mathbf{K}_\nu$:

$$p(\mathbf{K}_\nu|\mathbf{Z}_\nu,\boldsymbol{\beta}_0,\tau_\nu^2,\mathbf{W}) \propto p(\mathbf{K}_\nu) \times \left(\frac{|\mathbf{V}_{\tilde{\beta}_\nu}|}{|\mathbf{K}_\nu|}\right)^{\frac{1}{2}} \times \left(\frac{q_\sigma+\psi_\nu}{2}\right)^{-\frac{\alpha_\sigma+n_\nu}{2}}. \tag{A.7}$$

Using the same ideas one can obtain the complete conditional of σ_ν^2 with $\boldsymbol{\beta}_\nu$ integrated out, which strangely enough involves the same ψ_ν quantity:

$$\begin{aligned}
p(\sigma_\nu^2|\mathbf{Z}_\nu, d_\nu, g_\nu, \boldsymbol{\beta}_0, \tau^2, \mathbf{W}) &= \int p(\boldsymbol{\beta}_\nu, \sigma_\nu^2|\mathbf{Z}_\nu, d_\nu, g_\nu, \boldsymbol{\beta}_0, \tau^2, \mathbf{W}) \; d\boldsymbol{\beta}_\nu \\
&= p(\sigma_\nu^2) \int p(\mathbf{Z}_\nu|d_\nu, g_\nu, \boldsymbol{\beta}_\nu, \sigma_\nu^2) p(\boldsymbol{\beta}_\nu|\sigma_\nu^2, \boldsymbol{\beta}_0, \mathbf{W}) \; d\boldsymbol{\beta}_\nu \\
&= \left(\frac{|\mathbf{V}_{\tilde{\beta}_\nu}|}{(2\pi)^{n_\nu} \tau_\nu^{2m} |\mathbf{K}_\nu||\mathbf{W}|} \right)^{\frac{1}{2}} \sigma_\nu^{-n_\nu} p(\sigma_\nu^2) \exp\left\{ -\frac{\psi_\nu}{2\sigma_\nu^2} \right\} \\
&\propto \sigma_\nu^{-n_\nu} (\sigma_\nu^2)^{-(\alpha_\sigma/2+1)} \exp\left\{ -\frac{q_\sigma}{2\sigma_\nu^2} \right\} \exp\left\{ -\frac{\psi_\nu}{2\sigma_\nu^2} \right\} \\
&= (\sigma_\nu^2)^{-((\alpha_\sigma + n_\nu)/2+1)} \exp\left\{ -\frac{q_\sigma + \psi_\nu}{2\sigma_\nu^2} \right\},
\end{aligned}$$

which means that

$$\boxed{\sigma_\nu^2|d, g, \boldsymbol{\beta}_0, \mathbf{W} \sim IG((\alpha_\sigma + n_\nu)/2, (q_\sigma + \psi_\nu)/2).} \tag{A.8}$$

In addition to improving mixing, (A.8) will be useful for obtaining Gibbs draws for σ^2 after accepted *grow* or *prune* tree operations or when $\boldsymbol{\beta}$ may not be available.

Appendix B

Thoughts on the nugget

Following the development in Hjort & Omre (Hjort & Omre, 1994), a Gaussian process is often written as

$$Z(\mathbf{X}) = m(\mathbf{X}, \boldsymbol{\beta}) + \varepsilon(\mathbf{X}). \tag{B.1}$$

The mean function $m(\mathbf{X}, \boldsymbol{\beta})$ is taken to be linear in $\mathbf{X}$:

$$m(\mathbf{X}, \boldsymbol{\beta}) = \boldsymbol{F}\boldsymbol{\beta}.$$

$\boldsymbol{\beta}$ are coefficient parameters and $\mathbf{F} = (\mathbf{1}, \mathbf{X}^\top)^\top$. The process variance, governed by $\varepsilon(\mathbf{X})$ is such that

$$\text{cov}(Z(\mathbf{X}), Z(\mathbf{X}')) = \text{cov}(\varepsilon(\mathbf{X}), \varepsilon(\mathbf{X})) = \sigma^2 \mathbf{K}(\mathbf{X}, \mathbf{X}'),$$

where $\mathbf{K}$ is a correlation function depicting the smoothness of the process.

Accordingly, observations $z_i \times \mathbf{x}_i$ for $i = 1, \ldots, n$ are said to form a Gaussian process

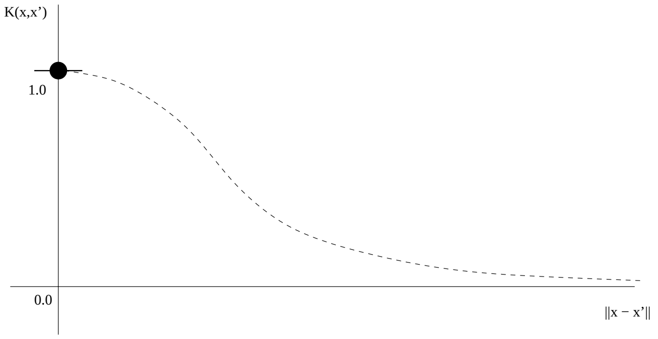


Figure B.1: Graphical depiction of the correlation function (B.3).

if they satisfy

$$(Z_1, \ldots, Z_n)^\top \sim N_n[\mathbf{F}\boldsymbol{\beta}, \sigma^2\mathbf{K}], \tag{B.2}$$

where correlation matrix $\mathbf{K}$ is constructed using a one of a family of parameterized correlation functions, such as the power family:

$$K(\mathbf{x}_i, \mathbf{x}_j|d) = \exp\left\{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{d}\right\}. \tag{B.3}$$

Such correlation matrices should be positive definite with all entries less than or equal to one. Figure B.1 shows an illustration of how correlation, like that described by (B.3), decays as the distance $||\mathbf{x}_i - \mathbf{x}_j||$ increases.

Relations for interpolation in terms of predicted mean and errors can be obtained using multivariate normal theory (see Hjort & Omre). This kind of spatial interpolation is commonly called *Kriging*. It easily seen that using a parameterized correlation function like the one in (B.3) results in a predictive mean of $\hat{Z}(\mathbf{x}_i) = z_i$, and error

$$\hat{\sigma}^2(\mathbf{x_i}) = E[\hat{Z}(\mathbf{x}_i) - Z(\mathbf{x}_i)]^2 = 0$$

when $\mathbf{x}_i$ corresponds to any of the input data locations $\mathbf{x}_1, \ldots, \mathbf{x}_n$. For $\mathbf{x} \neq \mathbf{x}_i$, $\hat{\sigma}^2(\mathbf{x}) > 0$, increasing as the distance from $\mathbf{x}$ to closest $\mathbf{x}_i$ gets large. An example interpolation is shown in Figure B.2.

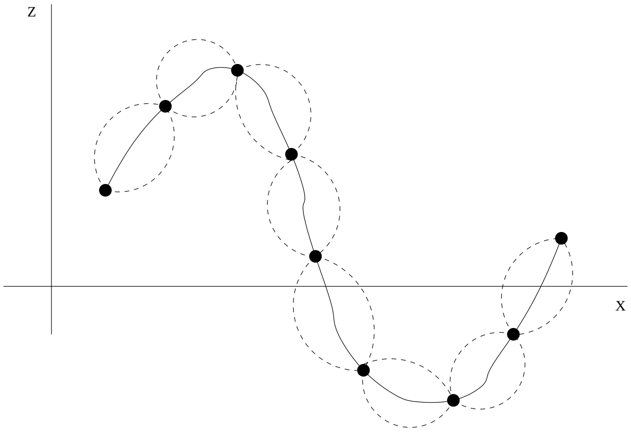


Figure B.2: Data interpolated by Kriging, using a correlation function like that in (B.3) with a model like that in (B.1).

However, if the modeler believes that the observations are subject to *measurement error*, then smoothing rather than interpolation is the goal. Figure B.3 shows what a possible smoothing of the data presented in Figure B.2 might look like.

To smooth the data, the model (B.1) must be augmented to include an additional variance term account for "measurement error". However, this is not the approach taken by everyone in the Geostatistical community. Instead, a common approach is to add a so-called *nugget* term (η) directly into the definition of the correlation function (B.3), leaving the underlying model formulation (B.1) unchanged:

$$K(\mathbf{x}_i, \mathbf{x}_j | d, \eta) = \exp\left\{ -\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{d} \right\} + \eta_{I\{i=j\}}, \tag{B.4}$$

where $I\{\cdot\}$ is the boolean indicator function. Note that the matrix $\mathbf{K}$ resulting from (B.4) is no

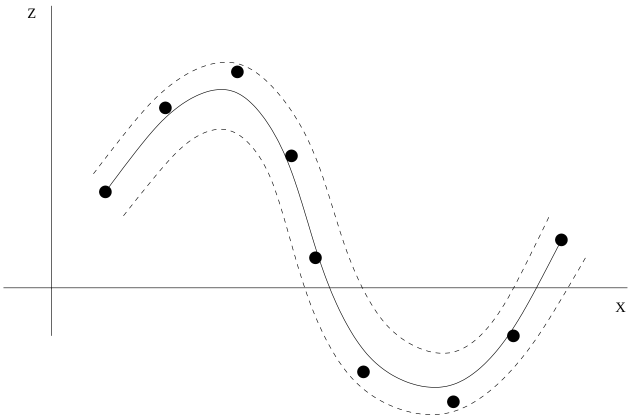


Figure B.3: A smooth alternative to interpolation of the data presented in Figure B.2

longer a correlation matrix (in the strictest sense) because its diagonal may have entries which are greater than one. Figure B.4 shows the resulting (dis-continuous) exponential correlation function graphically.

To my knowledge, the parameter η does not have a straightforward statistical interpretation. In fact, several authors advise against using this approach for this very reason. However, (B.4) gives that $K(\mathbf{x}, \mathbf{x}|d, \eta) = 1 + \eta$, which makes the prediction error non-zero for data locations $\mathbf{x}_i$ (see Hjort & Omre):

$$\hat{\sigma}^2(\mathbf{x}_i) = \sigma^2\eta, \quad \text{and} \quad \hat{Z}(\mathbf{x}_i) \neq z_i \quad (\text{unless } \eta = 0),$$

provided that one is careful about the bookkeeping for the standard Kriging equations (see Section B.1). Thus, the nugget accomplishes the goal of smoothing the data rather than interpolating. Predictive means and error-bars look similar to those drawn in Figure B.3, although uncertainty is usually somewhat lower near observed data locations.

The orthodox statistical way to account for measurement error is to augment the

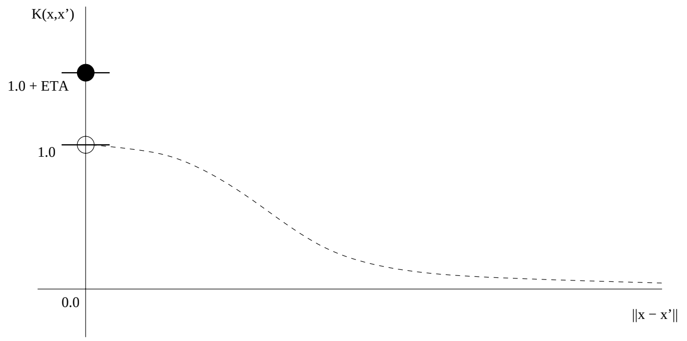


Figure B.4: Graphical depiction of the (dis-continuous) correlation function (B.4) with nugget.

model (B.1):

$$Z(\mathbf{X}) = m(\mathbf{X}, \boldsymbol{\beta}) + \varepsilon(\mathbf{X}) + \eta(\mathbf{x}), \tag{B.5}$$

where m and ε are as before, and $\eta(\mathbf{x})$ is an independent zero-mean noise process, usually Gaussian. Given observations $z_i \times \mathbf{x}_i$ for $i = 1, \ldots, n$, the corresponding Gaussian process can be written as a sum of independent normals:

$$\begin{aligned}(Z_1, \ldots, Z_m)^\top &\sim N_n[\mathbf{F}\boldsymbol{\beta}, \sigma^2\mathbf{K}] + N_n[\mathbf{0}, \tau^2\mathbf{I}] && \text{(B.6)}\\ &\sim N_n[\mathbf{F}\boldsymbol{\beta}, (\sigma^2 + \tau^2)\mathbf{K}'] && \text{(B.7)}\end{aligned}$$

where $\mathbf{K}'$ is a (true) correlation matrix defined in terms of $K(\mathbf{x}, \mathbf{x}'|d)$ from (B.3) by

$$K'(\mathbf{x}_i, \mathbf{x}_j | d, \sigma^2, \tau^2) = \frac{\sigma^2}{\sigma^2 + \tau^2}(K(\mathbf{x}, \mathbf{x}'|d) + \tau^2\mathbf{I}). \tag{B.8}$$

Equivalently:

$$K'(\mathbf{x}_i, \mathbf{x}_j | d, \sigma^2, \tau^2) = \frac{\sigma^2}{\sigma^2 + \tau^2}(K(\mathbf{x}, \mathbf{x}') + \tau^2_{I\{i=j\}}). \tag{B.9}$$

This is essentially the a scaled version of (B.4). Thus, there is really no difference between the model in (B.1) with a correlation that includes a nugget term (the *nugget model*), and a model like that in (B.5) which includes an explicit noise parameter. The main difference is that now all three ingredients (σ^2, τ^2, and $\mathbf{K}'$) have arguably more meaningful statistical interpretations.

Despite its less than satisfactory interpretability or statistical meaning, many authors have chosen the *nugget model* because its parameters are easy to estimate using Maximum Likelihood and Monte carlo based methods. Since σ^2 is, in a sense, de-coupled from the nugget it is possible to obtain Gibbs draws for σ^2 which would not be possible under (B.5) using (B.7). Moreover, the simplified structure allows an integrating out of $\boldsymbol{\beta}$ in the conditional posterior for both σ^2 and $\boldsymbol{\beta}$ in the full conditional posterior for $\mathbf{K}$.

B.1 Careful bookkeeping when predicting with the *nugget model.*

One has to be careful when applying the Kriging equations (like those in Hjort & Omre) when using the *nugget model* formulation mentioned above. To help illustrate, below we will re-write the prediction equations: The predicted value of $z(\mathbf{x})$ at $\mathbf{x}$ is normally distributed with mean and variance

$$\hat{z}(\mathbf{x}) = \mathbf{f}^\top(\mathbf{x})\boldsymbol{\beta} + \mathbf{k}(\mathbf{x})^\top \mathbf{K}^{-1}(\mathbf{t} - \mathbf{F}\boldsymbol{\beta}),$$

$$\hat{\sigma}(\mathbf{x})^2 = \sigma^2[\kappa(\mathbf{x},\mathbf{x}) - \mathbf{q}^\top(\mathbf{x})\mathbf{C}^{-1}\mathbf{q}(\mathbf{x})],$$

where $\mathbf{C}^{-1} = (\mathbf{K} + \mathbf{F}\mathbf{W}\mathbf{F}^\top)^{-1}$, $\mathbf{q}(\mathbf{x}) = \mathbf{k}(\mathbf{x}) + \mathbf{F}\mathbf{W}\mathbf{f}(\mathbf{x})$, $\mathbf{f}^\top(\mathbf{x}) = (1, \mathbf{x}^\top)$, $\kappa(\mathbf{x},\mathbf{y}) = K(\mathbf{x},\mathbf{y}) + \mathbf{f}^\top(\mathbf{x})\mathbf{W}\mathbf{f}(\mathbf{x})$, and $\mathbf{k}(\mathbf{x})$ is a $n-$vector with $\mathbf{k}_j(\mathbf{x}) = K(\mathbf{x}, \mathbf{x}_j)$, for all $\mathbf{x}_j \in \mathbf{X}$.

Here, the focus is mainly on the definitions of $\mathbf{k}(\mathbf{x})$, $\kappa(\mathbf{x},\mathbf{y})$, and $K(\mathbf{x},\mathbf{y})$ which are measurements of the correlation of a predictive location $\mathbf{x}$ and other locations $\mathbf{y}$.

Remember that the covariance matrix $\mathbf{K}$ is constructed using the definition for $K(\cdot,\cdot)$ from (B.4) giving correlations between the data locations $\mathbf{x}_i \in \mathbf{X}$, and results in a covariance matrix $\mathbf{K}$ which has $1 + \eta$ along the diagonal. According to (B.4) $K(\mathbf{x}_i, \mathbf{x}_j) = 1 + \eta$ when $i = j$. But when $i \neq j$ we have that $K(\mathbf{x}_i, \mathbf{x}_j) \leq 1$ even in the case where $\mathbf{x}_i = \mathbf{x}_j$ whence $K(\mathbf{x}_i, \mathbf{x}_j) = 1$.

Therefore, when computing $\mathbf{k}(\mathbf{x})$, $\kappa(\mathbf{x}, \mathbf{y})$ one has to be careful to make the distinction between the covariance between a point and itself, versus the covariance between multiple points with the same configurations *(because they are different)*. For example, if one is considering a new set of predictive locations, $\mathbf{y}_i \in \mathbf{Y}$, then $\mathbf{k}(\mathbf{y}_i)$ has entries less than or equal to 1 (no nugget), with equality only when $\mathbf{y}_i = \mathbf{x}_j$ for some $\mathbf{x}_j \in \mathbf{X}$. Alternatively, the correlation matrix between pairs of predictive locations from $\mathbf{Y}$ satisfies the same properties as that of $\mathbf{K}$, the correlation matrix of for the data locations ($\mathbf{X}$).

Appendix C

Active Learning – Cohn (ALC)

Section C.1 derives the ALC algorithm (Chapter 4) for a hierarchical Gaussian process (Chapter 2), and following in Section C.2 for a linear model (Chapter 3).

C.1 For Hierarchical Gaussian Process

The partition inverse equations (Barnett, 1979) can be used to write a covariance matrix $\mathbf{C}_{N+1}$ in terms of $\mathbf{C}_N$, so to obtain an equation for $\mathbf{C}_{N+1}^{-1}$ in terms of $\mathbf{C}_N^{-1}$:

$$\mathbf{C}_{N+1} = \begin{bmatrix} \mathbf{C}_N & \mathbf{m} \\ \mathbf{m}^\top & \kappa \end{bmatrix} \qquad \mathbf{C}_{N+1}^{-1} = \begin{bmatrix} [\mathbf{C}_N^{-1} + \mathbf{g}\mathbf{g}^\top \mu^{-1}] & \mathbf{g} \\ \mathbf{g}^\top & \mu \end{bmatrix} \tag{C.1}$$

where $\mathbf{m} = [C(\mathbf{x}_1, \mathbf{x}), \dots, C(\mathbf{x}_N, \mathbf{x})]$, $\kappa = C(\mathbf{x}, \mathbf{x})$, for an $N + 1^{\text{st}}$ point $\mathbf{x}$ where $C(\cdot, \cdot)$ is the covariance function, and

$$\mathbf{g} = -\mu \mathbf{C}_N^{-1}\mathbf{m} \qquad \mu = (\kappa - \mathbf{m}^\top \mathbf{C}_N^{-1}\mathbf{m})^{-1}.$$

If $\mathbf{C}_N^{-1}$ is available, these partitioned inverse equations allow one to compute $\mathbf{C}_{N+1}^{-1}$, without explicitly constructing $\mathbf{C}_{N+1}$. Moreover, the partitioned inverse can be used to compute $\mathbf{C}_{N+1}^{-1}$ with time in $O(n^2)$ rather than the usual $O(n^3)$.

Using notation for a hierarchically specified Gaussian process, in the context of ALC sampling, the matrix which requires an inverse is

$$\mathbf{K}_{N+1} + \mathbf{F}_{N+1}\mathbf{W}\mathbf{F}_{N+1}^\top$$

This matrix is key to the computation of the predictive variance $\hat{\sigma}(\mathbf{x})^2$.

$$\begin{aligned}\mathbf{K}_{N+1} + \mathbf{F}_{N+1}^\top\mathbf{W}\mathbf{F}_{N+1} &= \begin{bmatrix} \mathbf{K}_N & \mathbf{k}_N(\mathbf{x}) \\ \mathbf{k}_N^\top(\mathbf{x}) & K(\mathbf{x},\mathbf{x}) \end{bmatrix} + \begin{bmatrix} \mathbf{F}_N\mathbf{W}\mathbf{F}_N^\top & \mathbf{F}_N\mathbf{W}\mathbf{f}(\mathbf{x}) \\ \mathbf{f}(\mathbf{x})^\top\mathbf{W}\mathbf{F}_N^\top & \mathbf{f}(\mathbf{x})^\top\mathbf{W}\mathbf{f}(\mathbf{x}) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{K}_N + \mathbf{F}_N\mathbf{W}\mathbf{F}_N^\top & \mathbf{k}_N(\mathbf{x}) + \mathbf{F}_N\mathbf{W}\mathbf{f}(\mathbf{x}) \\ \mathbf{k}_N^\top(\mathbf{x}) + \mathbf{f}(\mathbf{x})^\top\mathbf{W}\mathbf{F}_N^\top & K(\mathbf{x},\mathbf{x}) + \mathbf{f}(\mathbf{x})^\top\mathbf{W}\mathbf{f}(\mathbf{x}) \end{bmatrix}.\end{aligned}$$

(*) Using the notation $\mathbf{C}_N = \mathbf{K}_N + \mathbf{F}_N\mathbf{W}\mathbf{F}_N^\top$, $\mathbf{q}_N(\mathbf{x}) = \mathbf{k}_N(\mathbf{x}) + \mathbf{F}_N\mathbf{W}\mathbf{f}(\mathbf{x})$, and $\kappa(\mathbf{x},\mathbf{y}) = K(\mathbf{x},\mathbf{y}) + \mathbf{f}(\mathbf{x})^\top\mathbf{W}\mathbf{f}(\mathbf{y})$ yields some simplification:

$$\mathbf{C}_{N+1} = \mathbf{K}_{N+1} + \mathbf{F}_{N+1}\mathbf{W}\mathbf{F}_{N+1}^\top = \begin{bmatrix} \mathbf{C}_N & \mathbf{q}_N(\mathbf{x}) \\ \mathbf{q}_N(\mathbf{x})^\top & \kappa(\mathbf{x},\mathbf{y}) \end{bmatrix}.$$

Applying the partitioned inverse equations (C.1) gives the following nice expression for $(\mathbf{K}_{N+1} + \mathbf{F}_{N+1}^\top\mathbf{W}\mathbf{F}_{N+1})^{-1}$:

$$\mathbf{C}_{N+1}^{-1} = (\mathbf{K}_{N+1} + \mathbf{F}_{N+1}^\top\mathbf{W}\mathbf{F}_{N+1})^{-1} = \begin{bmatrix} [\mathbf{C}_N^{-1} + \mathbf{g}\mathbf{g}^\top\mu^{-1}] & \mathbf{g} \\ \mathbf{g}^\top & \mu \end{bmatrix} \tag{C.2}$$

where

$$\mathbf{g} = -\mu \mathbf{C}_N^{-1} \mathbf{q}_N(\mathbf{x}) \qquad\qquad \mu = (\kappa(\mathbf{x},\mathbf{x}) - \mathbf{q}_N(\mathbf{x})^\top \mathbf{C}_N^{-1} \mathbf{q}_N(\mathbf{x}))^{-1}$$

using the most recent definitions of $\mathbf{C}_N$ and $\kappa(\cdot,\cdot)$, see (*).

From here an expression for the key quantity of the ALC algorithm from Seo et al. (2000) can be obtained. The expression calculates the reduction in variance at a point $\mathbf{y}$ given that the location $\mathbf{x}$ is added into the data:

$$\begin{aligned} &\Delta\hat{\sigma}^2_{\mathbf{y}}(\mathbf{x}) = \hat{\sigma}^2_{\mathbf{y}} - \hat{\sigma}^2_{\mathbf{y}}(\mathbf{x}), \\ \text{where} \qquad &\hat{\sigma}^2_{\mathbf{y}} = \sigma^2[\kappa(\mathbf{y},\mathbf{y}) - \mathbf{q}_N^\top(\mathbf{y})\mathbf{C}_N^{-1}\mathbf{q}_N^\top(\mathbf{y})], \\ \text{and} \qquad &\hat{\sigma}^2_{\mathbf{y}}(\mathbf{y}) = \sigma^2[\kappa(\mathbf{y},\mathbf{y}) - \mathbf{q}_{N+1}(\mathbf{y})^\top\mathbf{C}_{N+1}^{-1}\mathbf{q}_{N+1}(\mathbf{y})]. \end{aligned}$$

Now,

$$\begin{aligned} \Delta\hat{\sigma}^2_{\mathbf{y}}(\mathbf{x}) &= \sigma^2[\kappa(\mathbf{y},\mathbf{y}) - \mathbf{q}_N^\top(\mathbf{y})\mathbf{C}_N^{-1}\mathbf{q}_N(\mathbf{y})] - \sigma^2[\kappa(\mathbf{y},\mathbf{y}) - \mathbf{q}_{N+1}^\top(\mathbf{y})\mathbf{C}_{N+1}^{-1}\mathbf{q}_{N+1}(\mathbf{y})] \\ &= \sigma^2[\mathbf{q}_{N+1}(\mathbf{y})^\top\mathbf{C}_{N+1}^{-1}\mathbf{q}_{N+1}(\mathbf{y}) - \mathbf{q}_N^\top(\mathbf{y})\mathbf{C}_N^{-1}\mathbf{q}_N(\mathbf{y})]. \end{aligned}$$

Focusing on $\mathbf{q}_{N+1}^{\top}(\mathbf{y})\mathbf{C}_{N+1}^{-1}\mathbf{q}_{N+1}(\mathbf{y})$, first decompose $\mathbf{q}_{N+1}$:

$$\begin{aligned}
\mathbf{q}_{N+1} &= \mathbf{k}_{N+1}(\mathbf{y}) + \mathbf{F}_{N+1}\mathbf{W}\mathbf{f}(\mathbf{y}) \\
&= \begin{bmatrix} \mathbf{k}_N(\mathbf{y}) \\ K(\mathbf{y},\mathbf{x}) \end{bmatrix} + \begin{bmatrix} \mathbf{F}_N \\ \mathbf{f}^{\top}(\mathbf{x}) \end{bmatrix} \mathbf{W}\mathbf{f}(\mathbf{y}) \\
&= \begin{bmatrix} \mathbf{k}_N(\mathbf{y}) + \mathbf{F}_N\mathbf{W}\mathbf{f}(\mathbf{y}) \\ K(\mathbf{y},\mathbf{x}) + \mathbf{f}^{\top}(\mathbf{x})\mathbf{W}\mathbf{f}(\mathbf{y}) \end{bmatrix} = \begin{bmatrix} \mathbf{q}_N(\mathbf{y}) \\ \kappa(\mathbf{x},\mathbf{y}) \end{bmatrix}.
\end{aligned}$$

Turning attention back to $\mathbf{C}_{N+1}^{-1}\mathbf{q}_{n+1}(\mathbf{y})$, with the help of (C.2):

$$\begin{aligned}
\mathbf{C}_{N+1}^{-1}\mathbf{q}_{N+1}(\mathbf{y}) &= \begin{bmatrix} [\mathbf{C}_N^{-1} + \mathbf{g}\mathbf{g}^{\top}\mu^{-1}] & \mathbf{g} \\ \mathbf{g}^{\top} & \mu \end{bmatrix} \begin{bmatrix} \mathbf{q}_N(\mathbf{y}) \\ \kappa(\mathbf{x},\mathbf{y}) \end{bmatrix} \\
&= \begin{bmatrix} [\mathbf{C}_N^{-1} + \mathbf{g}\mathbf{g}^{\top}\mu^{-1}]\mathbf{q}_N(\mathbf{y}) + \mathbf{g}\kappa(\mathbf{x},\mathbf{y})) \\ \mathbf{g}^{\top}\mathbf{q}_N(\mathbf{y}) + \mu\kappa(\mathbf{x},\mathbf{y})] \end{bmatrix}.
\end{aligned}$$

Then, another multiplication:

$$\begin{aligned}
&\mathbf{q}_{N+1}^{\top}(\mathbf{y})\mathbf{C}_{N+1}^{-1}\mathbf{q}_{N+1}(\mathbf{y}) \\
&= \begin{bmatrix} \mathbf{q}_N(\mathbf{y}) \\ \kappa(\mathbf{x},\mathbf{y}) \end{bmatrix}^{\top} \begin{bmatrix} (\mathbf{C}_N^{-1} + \mathbf{g}\mathbf{g}^{\top}\mu^{-1})\mathbf{q}_N(\mathbf{y}) + \mathbf{g}\kappa(\mathbf{x},\mathbf{y})) \\ \mathbf{g}^{\top}\mathbf{q}_N(\mathbf{y}) + \mu\kappa(\mathbf{x},\mathbf{y}) \end{bmatrix} \\
&= \mathbf{q}_N^{\top}(\mathbf{y})[(\mathbf{C}_N^{-1} + \mathbf{g}\mathbf{g}^{\top}\mu^{-1})\mathbf{q}_N(\mathbf{y}) + \mathbf{g}\kappa(\mathbf{x},\mathbf{y})] \\
&+ \kappa(\mathbf{x},\mathbf{y})[\mathbf{g}^{\top}\mathbf{q}_N(\mathbf{y}) + \mu\kappa(\mathbf{x},\mathbf{y})].
\end{aligned}$$

Finally:

$$
\begin{aligned}
\Delta\hat{\sigma}^2_{\mathbf{y}}(\mathbf{x}) &= \sigma^2[\mathbf{q}_{N+1}(\mathbf{y})^\top \mathbf{C}^{-1}_{N+1}\mathbf{q}_{N+1}(\mathbf{y}) - \mathbf{q}^\top_N(\mathbf{y})\mathbf{C}^{-1}_N\mathbf{q}_N(\mathbf{y})]. \\
&= \sigma^2[\mathbf{q}^\top_N(\mathbf{y})\mathbf{g}\mathbf{g}^\top\mu^{-1}\mathbf{q}_N(\mathbf{y}) + 2\kappa(\mathbf{x},\mathbf{y})\mathbf{g}^\top\mathbf{q}_N(\mathbf{y}) + \mu\kappa(\mathbf{x},\mathbf{y})^2] \\
&= \sigma^2\mu[\mathbf{q}^\top_N(\mathbf{y})\mathbf{g}\mathbf{g}^\top\mu^{-2}\mathbf{q}_N(\mathbf{y}) + 2\mu^{-1}\kappa(\mathbf{x},\mathbf{y})\mathbf{g}^\top\mathbf{q}_N(\mathbf{y}) + \kappa(\mathbf{x},\mathbf{y})^2] \\
&= \sigma^2\mu\left[\mathbf{q}^\top_N(\mathbf{y})\mathbf{g}\mu^{-1} - \kappa(\mathbf{x},\mathbf{y})\right]^2,
\end{aligned}
$$

and some minor re-arranging after plugging in for μ and $\mathbf{g}$ gives:

$$
\Delta\hat{\sigma}^2_{\mathbf{y}}(\mathbf{x}) = \frac{\sigma^2\left[\mathbf{q}^\top_N(\mathbf{y})\mathbf{C}^{-1}_N\mathbf{q}_N(\mathbf{x}) - \kappa(\mathbf{x},\mathbf{y})\right]^2}{\kappa(\mathbf{x},\mathbf{x}) - \mathbf{q}^\top_N(\mathbf{x})\mathbf{C}^{-1}_N\mathbf{q}_N(\mathbf{x})}.
$$

C.2 For Hierarchical (Limiting) Linear Model

Under the (limiting) linear model, computing the ALC statistic is somewhat more straightforward. Starting back at the beginning; now with the predictive variance under the limiting linear model (3.8):

$$
\begin{aligned}
\Delta\hat{\sigma}^2_{\mathbf{y}}(\mathbf{x}) &= \hat{\sigma}^2_{\mathbf{y}} - \hat{\sigma}^2_{\mathbf{y}}(\mathbf{x}) \\
&= \sigma^2[1 - \mathbf{f}^\top(\mathbf{y})\mathbf{V}_{\tilde{\beta}_N}\mathbf{f}(\mathbf{y}) - 1 - \mathbf{f}^\top(\mathbf{y})\mathbf{V}_{\tilde{\beta}_{N+1}}\mathbf{f}(\mathbf{y})] \\
&= \sigma^2\mathbf{f}^\top(\mathbf{y})[\mathbf{V}_{\tilde{\beta}_N} - \mathbf{V}_{\tilde{\beta}_{N+1}}]\mathbf{f}(\mathbf{y}),
\end{aligned}
$$

where $\mathbf{V}_{\tilde{\beta}_{N+1}}$ from Eq. (2.4) includes $\mathbf{x}$, and $\mathbf{V}_{\tilde{\beta}_N}$ does not. Expanding out $\mathbf{V}_{\tilde{\beta}_{N+1}}$:

$$\begin{aligned}
&\Delta\hat{\sigma}^2_{\mathbf{y}}(\mathbf{x}) \\
&= \sigma^2 \mathbf{f}^\top(\mathbf{y}) \left[\mathbf{V}_{\tilde{\beta}_N} - \left(\frac{\mathbf{W}^{-1}}{\tau^2} + \frac{\mathbf{F}_{N+1}^\top \mathbf{F}_{N+1}}{1+g} \right)^{-1} \right] \mathbf{f}^\top(\mathbf{y}) \\
&= \sigma^2 \mathbf{f}^\top(\mathbf{y}) \left[\mathbf{V}_{\tilde{\beta}_N} - \left(\frac{\mathbf{W}^{-1}}{\tau^2} + \frac{1}{1+g} \begin{bmatrix} \mathbf{F}_N \\ \mathbf{f}^\top(\mathbf{x}) \end{bmatrix}^\top \begin{bmatrix} \mathbf{F}_N \\ \mathbf{f}^\top(\mathbf{x}) \end{bmatrix} \right)^{-1} \right] \mathbf{f}(\mathbf{y}) \\
&= \sigma^2 \mathbf{f}^\top(\mathbf{y}) \left[\mathbf{V}_{\tilde{\beta}_N} - \left(\frac{\mathbf{W}^{-1}}{\tau^2} + \frac{\mathbf{F}_N^\top \mathbf{F}_N}{1+g} + \frac{\mathbf{f}(\mathbf{x})\mathbf{f}^\top(\mathbf{x})}{1+g} \right)^{-1} \right] \mathbf{f}(\mathbf{y}) \\
&= \sigma^2 \mathbf{f}^\top(\mathbf{y}) \left[\mathbf{V}_{\tilde{\beta}_N} - \left(\mathbf{V}^{-1}_{\tilde{\beta}_N} + \frac{\mathbf{f}(\mathbf{x})\mathbf{f}^\top(\mathbf{x})}{1+g} \right)^{-1} \right] \mathbf{f}(\mathbf{y}).
\end{aligned}$$

Using the Sherman-Morrison-Woodbury formula (Bernstein, 2005) [see Section 3.2.1], where $\mathbf{V} \equiv \mathbf{f}^\top(\mathbf{x})(1+g)^{-\frac{1}{2}}$ and $\mathbf{A} \equiv \mathbf{V}^{-1}_{\tilde{\beta}_N}$ gives

$$\Delta\hat{\sigma}^2_{\mathbf{y}}(\mathbf{x}) = \sigma^2 \mathbf{f}^\top(\mathbf{y}) \left[\left(1 + \frac{\mathbf{f}^\top(\mathbf{x})\mathbf{V}_{\tilde{\beta}_N}\mathbf{f}(\mathbf{x})}{1+g} \right)^{-1} \mathbf{V}_{\tilde{\beta}_N} \frac{\mathbf{f}(\mathbf{x})\mathbf{f}^\top(\mathbf{x})}{1+g} \mathbf{V}_{\tilde{\beta}_N} \right] \mathbf{f}(\mathbf{y}).$$

Combining and rearranging gives

$$\Delta\hat{\sigma}^2_{\mathbf{y}}(\mathbf{x}) = \frac{\sigma^2 [\mathbf{f}^\top(\mathbf{y})\mathbf{V}_{\tilde{\beta}_N}\mathbf{f}(\mathbf{x})]^2}{1 + g + \mathbf{f}^\top(\mathbf{x})\mathbf{V}_{\tilde{\beta}_N}\mathbf{f}(\mathbf{x})}.$$

Bibliography

Abrahamsen, P. (1997). *A review of Gaussian random fields and correllation functions* (Technical Report 917). Norwegian Computing Center, Box 114 Blindern, N-0314 Oslo, Norway.

Abramowitz, M., & Stegun, I. A. (1964). *Handbook of mathematical functions with formulas, graphs, and mathematical tables*. New York: Dover. 9th dover printing, 10th gpo printing–edition.

Adler, R. J. (1990). *An introduction to continuity, extrema, and related topics for general Gaussian processes* (Technical Report). Institute of Mathematical Statistics, Hayward, CA.

Andrieu, C., de Freitas, N., & Jordan, M. (2003). An introduction to MCMC for Machine Learning. *Machine Learning*, *50*, 5–43.

Angluin, D. (1987). Queries and concept learning. *Machine Learning*, *2*, 319–342.

Atlas, L., Cohn, D., Ladner, R., El-Sharkawi, M., Marks, R., Aggoune, M., & Park, D. (1990). Training connectionist networks with queries and selective sampling. *Advances in Neural Information Processing Systems*, 566–753.

Banerjee, S., & Gelfand, A. (2005). Boundary analysis: significance and construction of curvilinear boundaries. *under revision*.
http://www.biostat.umn.edu/∼sudiptob/ResearchPapers/BanerjeeGelfandJASA1.pdf.

Barnett, S. (1979). *Matrix methods for engineers and scientists*. McGraw-Hill.

Bates, R. A., Buck, R. J., Riccomagno, E., & Wynn, H. P. (1996). Experimental design and observation for large systems. *Journal of the Royal Statistical Society, Series B.*, *58*, 77–94.

Berger, J., Oliveira, V. D., & Sansó, B. (2001). *Objective Bayesian analysis of spatially correlated data* (Technical Report). CESMa. To appear in Journal of the American Statistical Association.

Bernardo, J., & Smith, A. (1994). *Bayesian theory*. Chichester: John Wiley & Sons.

Bernstein, D. (2005). *Matrix mathematics*. Princeton, NJ: Princeton University Press.

Box, G. E. P., Hunter, W. G., & Hunter, J. S. (1978). *Statistics for experimenters*. New York: Wiley.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*, 123–140.

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

Carlin, B., & Louis, T. (2000). *Bayes and emperical Bayes methods for data analysis*. Boca Raton, FL: Chapman & Hall/CRC.

Chaloner, K., & Verdinelli, I. (1995). Bayesian experimental design, a review. *Statistical Science*, *10 No. 3*, 273–1304.

Chipman, H., George, E., & McCulloch, R. (1998). Bayesian CART model search (with discussion). *Journal of the American Statistical Association*, *93*, 935–960.

Chipman, H., George, E., & McCulloch, R. (2002). Bayesian treed models. *Machine Learning*, *48*, 303–324.

Chu, W., Keerthi, S. S., & Ong, C. J. (2004). Bayesian support vector regression using a unified loss function. *IEEE Transactions on Neural Networks*, *15(1)*, 29–44.

Cogdon, P. (2001). *Bayesian statistical modelling*. New York, NY: John Wiley & Sons.

Cohn, D. A. (1996). Neural network exploration using optimal experimental design. *Advances in Neural Information Processing Systems* (pp. 679–686). Morgan Kaufmann Publishers.

Cormen, T. H., Leiserson, C. E., & Rivest, R. L. (1990). *Introduction to algorithms*. The MIT Electrical Engineering and Computer Science Series. MIT Press/McGraw Hill.

Cressie, N. (1991). *Statistics for spatial data*. John Wiley and Sons, Inc.

Currin, C., Mitchell, T., Morris, M., & Ylvisaker, D. (1988). *A Bayesian approach to the design and analysis of computer experiments* (Technical Report 6498). Oak Ridge National Laboratory.

Currin, C., Mitchell, T., Morris, M., & Ylvisaker, D. (1991). Bayesian prediction of deterministic functions, with applicatons to the design and analysis of computer experiments. *Journal of the American Statistical Association*, *86*, 953–963.

Damian, D., Sampson, P. D., & Guttorp, P. (2001). Bayesian estimation of semiparametric nonstationary spatial covariance structure. *Environmetrics*, *12*, 161–178.

Denison, D., Adams, N., Holmes, C., & Hand, D. (2002). Bayesian partition modelling. *Computational Statistics and Data Analysis*, *38*, 475–485.

Denison, D., Mallick, B., & Smith, A. (1998). A Bayesian CART algorithm. *Biometrika*, *85*, 363–377.

Dey, D., Müeller, P., & Sinha, D. (1998). *Practical nonparametric and semiparametric Bayesian statistics*. New York, NY, USA: Springer-Verlag New York, Inc.

Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A. J., & Vapnik, V. (1996). Support vector regression machines. *Advances in Neural Information Processing Systems* (pp. 155–161). MIT Press.

DuMouchel, W., & Jones, B. (1985). Model robust response surface designs: scaling two-level factorials. *Biometrika*, *72*, 513–526.

DuMouchel, W., & Jones, B. (1994). A simple Bayesian modification of D-optimal designs to reduce dependence on and assumed model. *Technometrics*, *36*, 37–47.

Ferreira, M. A., Higdon, D., Lee, H. K., & West, M. (2005). Multi-scale random field models. DME–UFRG Technical report, *submitted*; http://www.dme.ufrj.br/ marco/msmrf.pdf.

Fields Development Team (2004). fields: Tools for spatial data. National Center for Atmospheric Research, Boulder CO. URL: http://www.cgd.ucar.edu/Software/Fields.

Fine, S. (1999). *Knowledge acquisition in statistical learning theory.* Doctoral dissertation, Hebrew University, Jerusalem, Israel.

Fine, S., Gilad-Bachrach, R., & Shamir, E. (2000). Learning using query by committee, linear separation and random walks. *Eurocolt '99*, *1572 of LNAI*, 34–49.

Friedman, J. H. (1991). Multivariate adaptive regression splines. *Annals of Statistics*, *19, No. 1*, 1–67.

Fuentes, M., & Smith, R. L. (2001). *A new class of nonstationary spatial models* (Technical Report). North Carolina State University, Raleigh, NC.

Gamerman, D. (1997). *Markov chain Monte Carlo.* London: Chapman & Hall.

Gelman, A., Carlin, J. B., Stern, H. S., & Rubin, D. B. (1995). *Bayesian data analysis.* London: Chapman and Hall.

Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *6*, 721–741.

George, E. I., & McCulloch, R. E. (1994). Variable selection via Gibbs sampling. *Journal of the American Statistical Association*, *85*, 389–409.

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

Glover, F. W., & Laguna, M. (1997). *Tabu search.* Springer. 1 edition, ISBN: 079239965X.

Golub, G. H., & Van Loan, C. F. (1996). *Matrix computations.* Baltimore, MD: Johns Hopkins.

Hamada, M., Martz, H., Reese, C., & Wilson, A. (2001). Finding near-optimal Bayesian experimental designs by genetic algorithms. *American Statistical Association*, *55–3*, 175–181.

Harrison, D., & Rubinfeld, D. L. (1978). Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, *5*, 81–102.

Hartigan, J. (1964). *Bayes theory.* New York, NY: Springer-Verlag.

Hastings, W. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, *57*, 97–109.

Higdon, D. (2002). Space and space-time modeling using process convolutions. *Quantitative Methods for Current Environmental Issues* (pp. 37–56). London: Springer-Verlag.

Higdon, D., Swall, J., & Kern, J. (1999). Non-stationary spatial modeling. In J. M. Bernardo, J. O. Berger, A. P. Dawid and A. F. M. Smith (Eds.), *Bayesian statistics 6*, 761–768. Oxford University Press.

Hjort, N. L., & Omre, H. (1994). Topics in spatial statistics. *Scandinavian Journal of Statistics*, *21*, 289–357.

Hoeting, J., Madigan, D., Raftery, A., & Volinsky, C. (1999). Bayesian model averaging: A tutorial (with discussion). *Statistical Science*, *14*, 382–417.

Huang, D., Allen, T., Notz, W., & MIller, R. (2005a). Sequential kriging optimization using multiple fidelity evaluations. *submitted to:* Structural and Multidisciplinary Optimization.

Huang, D., Allen, T., Notz, W., & Zheng, N. (2005b). Global optimization of stochastic black-box systems via sequential kriging meta-models. *accepted to:* Journal of Global Optimization.

Jeffreys, H. (1961). *Theory of probability.* New York, NY: Oxford University Press.

Jones, D., Schonlau, M., & Welch, W. J. (1998). Efficient global optimization of expensive black box functions. *Journal of Global Optimization*, *13*, 455–492.

Kass, R., & Raftery, A. (1995). Bayes factors. *Journal of the American Statistical Association*, *90*, 773–795.

Kennedy, M., & O'Hagan, A. (2000). Predicting the output from a complex computer code when fast approcimations are avaliable. *Biometrika*, *87*, 1–13.

Kennedy, M., & O'Hagan, A. (2001). Bayesian calibration of computer models (with discussion). *Journal of the Royal Statistical Society, Series B*, *63*, 425–464.

Kim, H.-M., Mallick, B. K., & Holmes, C. C. (2002). *Analyzing non-stationary spatial data using piecewise Gaussian processes* (Technical Report). Texas A&M University – Corpus Christi.

MacKay, D. J. C. (1992). Information-based objective functions for active data selection. *Neural Computation*, *4*, 589–603.

Matheron, G. (1963). Principles of geostatistics. *Economic Geology*, *58*, 1246–1266.

McKay, M. D., Conover, W. J., & Beckman, R. J. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, *21*, 239–245.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., & Teller, R. (1953). Equations of state calculations by fast computing machine. *Journal of Chemical Physics*, *21*, 1087–1091.

Mitchell, T. (1974). An algorithm for the construction of *D*-optimal experimental designs. *Technometrics*, *16*, 203–210.

Mitchell, T. J., & Morris, M. D. (1992). Bayesian design and analysis of computer experiments: Two examples. *Statistica Sinica*, *2*, 359–379.

Müeller, P. (1999). Simulation based optimal design. In J. Berger, J. Bernardo, A. Dawid and A. Smith (Eds.), *Bayesian stastics*, 459–474. Oxford University Press.

Müeller, P., & Parmigiani, G. (1995). Optimal design via curve fitting of Monte carlo experiment. *Journal of the American Statistical Association*, *90*, 1322–1330.

Müller, P., Sansó, B., & de Iorio, M. (2004). Optimal Bayesian design by inhomogeneous Markov chain simulation. *Journal of the American Statistical Association, 99(467), Theory and Methods*, 788–798.

Neal, R. (1997). *Monte carlo implementation of Gaussian process models for Bayesian regression and classification"* (Technical Report CRG–TR–97–2). Dept. of Computer Science, University of Toronto.

Newman, D., Hettich, S., Blake, C., & Merz, C. (1998). UCI repository of machine learning databases. *url:* http://www.ics.uci.edu/∼mlearn/MLRepository.html.

O'Hagan, A. (1985). Curve fitting and optimal design for prediction (with discussion). *Journal of the Royal Statistical Society, Series B*, *40*, 1–41.

O'Hagan, A., Kennedy, M. C., & Oakley, J. E. (1999). Uncertainty analysis and other inference tools for complex computer codes. *Bayesian Statistics 6* (pp. 503–524). Oxford University Press.

Paciorek, C. (2003). *Nonstationary Gaussian processes for regression and spatial modelling.* Doctoral dissertation, Carnegie Mellon University, Pittsburgh, Pennsylvania.

Press, S. (1989). *Bayesian statistics: Principles, models, and applications.* New York, NY: John Wiley & Sons.

R Development Core Team (2004). *R: A language and environment for statistical computing.* R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-00-3.

Ramakrishnan, N., Bailey-Kellogg, C., Tadepalli, S., & Pandey, V. (2005). Gaussian processes for active data mining of spatial aggregates. *Proceedings of the SIAM Data Mining Conference.* http://www.cs.dartmouth.edu/∼cbk/papers/sdm05.pdf.

Ranjan, P. (2005). Sequential experiment design for contour estimation from complex computer codes. *Talk:* Design and Analysis of Experiments Conference 2005, Santa Fe, NM; http://www.stat.lanl.gov/DAE2005/index.html.

Rasmussen, C., & Ghahramani, Z. (2002). Infinite mixtures of Gaussian process experts. *Advances in Neural Information Processing Systems* (pp. 881–888). MIT Press.

Reese, C., Wilson, A., Hamada, M., Martz, H., & Ryan, K. (2005). Integrated analysis of computer and physical experiments. *Accepted to:* Technometrics.

Richardson, S., & Green, P. J. (1997). On Bayesian analysis of mixtures with an unknown number of components. *Journal of the Royal Statistical Society, Series B, Methodological*, *59*, 731–758.

Robert, C. (2001). *The Bayesian choice: From decision-theoretic foundations to computational implementation.* New York, NY: Springer-Verlag.

Robert, C., & Castella, G. (2000). *Monte carlo statistical methods.* New York, NY: Springer-Verlag.

Rogers, S. E., Aftosmis, M. J., Pandya, S. A., N. M. Chaderjian, E. T. T., & Ahmad, J. U. (2003). Automated CFD parameter studies on distributed parallel computers. *16th AIAA Computational Fluid Dynamics Conference.* AIAA Paper 2003-4229.

Sacks, J., Welch, W. J., Mitchell, T. J., & Wynn, H. P. (1989). Design and analysis of computer experiments. *Statistical Science*, *4*, 409–435.

Sampson, P. D., & Guttorp, P. (1992). Nonparametric estimation of nonstationary spatial covariance structure. *Journal of the American Statistical Association*, *87(417)*, 108–119.

Santner, T. J., Williams, B. J., & Notz, W. I. (2003). *The design and analysis of computer experiments*. New York, NY: Springer-Verlag.

Schmidt, A., & Gelfand, A. (2003). A Bayesian coregionalization approach for multivariate pollutant data. *Journal of Geophysical Research–Atmospheres*, *D24*, 108.

Schmidt, A. M., & O'Hagan, A. (2003). Bayesian inference for nonstationary spatial covariance structure via spatial deformations. *Journal of the Royal Statistical Society, Series B*, *65*, 745–758.

Schonlau, M. (1997). *Computer experiments and global optimiszation*. Doctoral dissertation, University of Waterloo, Waterloo, Ontario, Canada.

Sebastiani, P., & Wynn, H. P. (2000). Maximum entropy sampling and optimal Bayesian experimental design. *Journal of the Royal Stastical Society, Series B*, *62*, 145–157.

Seo, S., Wallat, M., Graepel, T., & Obermayer, K. (2000). Gaussian process regression: Active data selection and test point rejection. *Proceedings of the International Joint Conference on Neural Networks* (pp. 241–246). IEEE.

Silverman, B. W. (1985). Some aspects of the spline smoothing approach to non-parametric curve fitting. *Journal of the Royal Statistical Society Series B*, *47*, 1–52.

Stein, M. L. (1999). *Interpolation of spatial data*. New York, NY: Springer.

Thiébaux, H. J. (1997). A Bayesian coregionalization approach for multivariate pollutant data. *Journal of Climatology*, *10*, 567–573.

Thiébaux, H. J., & Pedder, M. A. (1987). *Spatial objective analysis: with applications in atmospheric science*. London Academic. 3rd edition.

Ver Hoef, J., & Barry, R. P. (1998). Constructing and fitting models for cokriging and multivariate spatial prediction. *Journal of Statistical Planning and Inference*, *69*, 275–294.

Wackernagel, H. (2003). *Multivariate geostatistics*. Berlin Heidelberg: Springer-Verlag.

Warmuth, M. K., Liao, J., Ratsch, G., Mathieson, M., Putta, S., & Lemmen, C. (2003). Support vector machines for active learning in the drug discovery process. *Journal of Chemical Information Sciences*, *43(2)*, 667–672.

Warmuth, M. K., Ratsch, G., Mathieson, M., Liao, J., & Lemmen, C. (2001). Active learning in the drug discovery process. *Advances in Neural Information Processing Systems, Vancouvner BC, Canada*.

Welch, W. J., Buck, R. J., Sacks, J., Wynn, H. P., Mitchell, T., & Morris, M. D. (1992). Screening, predicting, and computer experiment. *Technometrics*, *34*, 15–25.

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

Zhang, H. (2004). Inconsistent estimation and asymptotically equal interpolations in model-based geostatistics. *Journal of the American Statistical Association, Theory and Methods*, *99*, 250–261.