Designing and analyzing a circuit device experiment using treed Gaussian processes

Herbert K. H. Lee, Matthew Taddy, Robert B. Gramacy, and Genetha A. Gray*

Abstract

The development of circuit devices can involve both physical and computer simulation experiments. Here, we discuss statistical solutions for both design of the physical experiment and optimization for the calibration of the computer model. In both cases, we can view the problem as one of optimization, and we rely upon treed Gaussian processes to model the data and guide our design choices.

1 Introduction

This chapter describes work in the development of a new circuit device, and is part of a collaboration with scientists at Sandia National Laboratories. Circuit devices need to be tested for various capabilities during the development phase, in order to eventually create a device that will be effective in a range of operational environments. Both physical and computer simulation experiments are used. Our work here focuses on two parts of this process: creation of the design for the physical experiment, and optimization during the calibration of the computer model. Our key statistical tool is the treed Gaussian process.

The goal of our collaborators was to both "calibrate" and "validate" computer models of the electrical circuit devices. These models can then be used for the next stage of the design

*This work was partially supported by NASA awards 08008-002-011-000 and SC 2003028 NAS2-03144, Sandia National Laboratories grants 496420 and 673400, and National Science Foundation grants DMS 0233710 and 0504851. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the funding organizations.

process. In this context, calibration uses data from physical experiments to inform upon uncertain input parameters of a computer simulation model (Kennedy and O'Hagan, 2001, for example). Model parameters are tuned so that the code calculations in the simulator most closely match the observed behavior of the experimental data. Accurate calibration both improves predictive capabilities and minimizes the information lost by using a numerical model instead of the actual system.

In contrast, the validation process is applied in order to quantify the degree to which a computational model is an accurate representation of the real world phenomena it seeks to represent (Oberkampf et al., 2003). Validation is critical to ensure some level of predictive capability of the simulation codes so that these codes can be subsequently used to study or predict situations for which experimental data are unavailable due to environmental or economic limitations. Model validation is a major effort regularly undertaken by numerous government agencies and private companies. Our role as statisticians focused on the calibration aspects, and did not deal directly with validation.

1.1 Circuit Experiments

The circuit devices under study are bipolar junction transistors (bjt), which are used to amplify electrical current. They exist in both PNP and NPN constructions (Sedra and Smith, 1997). Twenty different such devices were studied; in this chapter we present results only for a few of them. For example, we will consider the *bft92a*, which is a PNP, and the *bfs17a* which is an NPN. The primary interest is in understanding current output as a function of the intensity of a pulse of gamma radiation applied to the devices, where

the current output is characterized by the peak amplitude reached during the experiment. Because of the physical setup of the experiment, a particular testing facility is capable of only a rather limited range of possible radiation doses, so experiments were run at three different facilities to span a broader range of possible dose rates. Although in principle the relationship between dose rate and peak amplitude should not depend on the facility, some of the results do appear to show a facility effect. Experimental runs were done at three different temperature settings. The scientists do allow for the results to depend on temperature, and they have separate computer simulation models for each temperature. Further details on the physical experiment are available in Gray et al. (2007).

The physical experiment is accompanied by a collection of computer simulation models. These radiation-aware models are built on a Xyce implementation of the Tor Fjeldy photocurrent model for the bjt. Xyce is an electrical circuit simulator developed at Sandia National Laboratories (Keiter, 2004), and the Tor Fjeldy photocurrent model is described in detail in Fjeldy et al. (1997). The model input is a radiation pulse expressed as a dose rate over time. The corresponding output is a curve of current value over time which reflects the response of the electrical device. This curve is summarized by its maximum value. The model also involves 38 user-defined tuning parameters. It is these parameters that need to be calibrated using the physical data so that the simulator output is as close as possible to the results from the physical experiments. All bjts share a basic underlying model and the same computer simulator can be used to approximate their behavior, with only changes to the parameters required.

1.2 Treed Gaussian Processes

Treed Gaussian processes are a highly flexible and computationally efficient model for spatially correlated data, as well as for more general functions. They combine standard Gaussian processes with treed partition models, producing an effective semi-parametric non-stationary model. In this section, we start with a brief review of Gaussian processes, cover the basics of treed Gaussian processes, then discuss their use for adaptive sampling in computer experiments and for optimization. Additional details are available in the appendix.

Gaussian processes are the standard model for creating a statistical emulator of the output of a computer simulator (Sacks et al., 1989; Kennedy and O'Hagan, 2001; Santner et al., 2003). A Gaussian process (GP) specifies that the set of responses $z(x_1, \ldots, x_m)$ at any finite collection of locations x (which could be spatial locations or multivariate inputs to a computer model) has a multivariate Gaussian distribution. In general, we can write $z(x) = \mu(x) + w(x)$ where μ is a mean function and $w(x)$ is a zero mean random process with covariance $C(x_j, x_k)$. Typically we will assume stationarity for the correlation structure, and thus we have $C(x_j, x_k) = \sigma^2 K(x_j, x_k)$ where K is a correlation matrix whose entries depend only on the difference vectors $x_j - x_k$. Sometimes a further assumption of isotropy is made, and then the correlations depend only on the distance between x_j and x_k, typically specified with a simple parametric form (Abrahamsen, 1997). We take the mean function to be linear in the inputs, $\mu(x) = \beta' x$. We also use a small nugget to allow for noisy data, for smoothing, or for numerical stability. For more details on GPs, we refer the reader to Cressie (1993) or Stein (1999). Herein we use the separable power family for the correlation structure, with a separate range parameter d_i in each dimension ($i = 1, \ldots, m_X$), and power $p_0 = 2$ to give

smooth realizations:

$$K(x_j, x_k|d) = \exp\left\{-\sum_{i=1}^{m_X} \frac{|x_{ij} - x_{ik}|^{p_0}}{d_i}\right\}. \quad (1)$$

Standard GPs have three drawbacks that affect us here. First, they are computationally intensive to fit, with effort growing with the cube of the sample size due to the need to invert the covariance matrix. Second, GP models usually assume stationarity, in that the same covariance structure is used throughout the entire input space, which may be too strong of a modeling assumption, or else they are fully nonstationary and too computationally expensive to be used for more than a relatively small number of datapoints. Third, the estimated predictive error of a stationary model does not directly depend on the locally observed response values. Rather, the predictive error at a point depends only on the locations of the nearby observations and on a global measure of error that uses all of the discrepancies between observations and predictions without regard for their distance from the point of interest (because of the stationarity assumption). Thus there is no ready local measure of lack-of-fit or uncertainty, which will be needed for sequential experimental design (adaptive sampling).

All of these issues can be addressed by combining the stationary GP model with treed partitioning, resulting in what we call a treed Gaussian process (TGP). The input space is partitioned into disjoint regions, with an independent stationary GP fit in each region. This approach provides a computationally efficient way of creating a nonstationary model. It reduces the overall computational demands by fitting separate GP models to smaller data sets (the individual partitions). The partitioning approach is based on that of Chipman et al. (1998, 2002), who used it to develop the Bayesian Classification and Regression Trees.

Reversible jump Markov chain Monte Carlo (Green, 1995) with tree proposal operations (prune, grow, swap, change, and rotate) allows simultaneous fitting of the tree and the parameters of the individual GP models. In this way, all parts of the model can be learned automatically from the data, and Bayesian model averaging through reversible jump allows for explicit estimation of predictive uncertainty. It also provides a smooth mean fit when appropriate (as the partition locations are also averaged over, so the mean function does not exhibit discontinuities unless the data call for such a fit). We provide more details in the appendix, and also point the reader to Gramacy and Lee (2008a). Software is available in the `tgp` library for the statistical package R at `http://www.cran.r-project.org/web/packages/tgp/index.html` (Gramacy and Taddy, 2008; Gramacy, 2007).

1.2.1 Adaptive Sampling

When each datapoint is difficult or expensive to collect, either because a physical experiment is involved or because the computer simulator takes a long time to run, it is important to choose the sample points with great care. One needs to select the set of design points that will provide maximum information about the problem. Traditional methods for experimental design involve starting with a specific model then creating a design of all of the planned runs. If a simple enough model is chosen, or enough assumptions about parameter values are made, then optimal designs can be found in closed form. For more complex cases, optimal designs are found numerically. Various optimality criteria lead to designs such as $A-$optimal, $D-$optimal, maximin, maximum entropy, and Latin hypercube designs. A good review of the Bayesian literature on optimal designs is provided by Chaloner and Verdinelli (1995). However, such an approach is not readily amenable to learning about the process as the data

are collected. The concept of sequential experimental design is less well-developed, but work has progressed in some areas, such as computer experiments (Sacks et al., 1989; Santner et al., 2003). By updating the design after learning from each new observation (or each new batch), one can better deal with lack of knowledge about model parameters, or even about the model itself. Here we are concerned about learning both the tree structure and the GP parameters during the experiment, and so direct application of optimality criteria is not feasible. Some additional background on experimental design is in the appendix.

In selecting design points sequentially, we want those which will provide the most additional information, conditional on the data we have already collected. Because we are fitting a fully Bayesian model, we can consider the predictive variance as a measure of our uncertainty. Thus one way to define the amount of information we expect to learn from a new datapoint is to look at the expected reduction in squared error averaged over the input space. This approach was developed in the active learning literature in computer science by Cohn (1996). This expected reduction in global error for a proposed new sample $\tilde{x}$ under a GP model can be expressed as:

$$\Delta\hat{\sigma}^2(\tilde{x}) = \int \Delta\hat{\sigma}^2_{\tilde{x}}(y)dy \equiv \int \hat{\sigma}^2(y) - \hat{\sigma}^2_{\tilde{x}}(y)dy = \int \frac{\sigma^2\left[q'(y)C^{-1}q(x) - \kappa(x,y)\right]^2}{\kappa(x,x) - q'(x)C^{-1}q(x)}dy \qquad (2)$$

where y represents the input space, σ^2 is the overall variance, $\hat{\sigma}^2_{\tilde{x}}(y)$ is the estimated (posterior) predictive variance at y when $\tilde{x}$ is added into the design, and

$$\begin{aligned} C^{-1} &= (K + \tau^2 FWF')^{-1} \\ q(x) &= k(x) + \tau^2 FWf(x) \\ \kappa(x,y) &= K(x,y) + \tau^2 f'(x)Wf(y) \end{aligned}$$

with $f'(x) = (1, x')$, and $k(x)$ a $n-$vector with $k_{\nu,j}(x) = K(x, x_j)$, for all $x_j \in X$. Refer

to Appendix 4 for definitions of the other quantities, e.g., W and K. For a treed GP, this expression is evaluated within partitions each MCMC iteration and the result averaged across the whole MCMC run. For points in separate partitions, there is no change in predictive variance. More details on this derivation are given by Gramacy and Lee (2009). In practice the integral is evaluated as a sum over a grid of locations.

Searching for the optimal design point over a multi-dimensional continuous space is quite difficult and computationally expensive, so we restrict our attention to a smaller set of candidate points that are well-spaced out relative to each other, and then choose the best point from our candidate set. To obtain a well-spaced set, we return to standard optimal designs, such as maximum entropy designs, maximin designs, and/or Latin hypercubes. This approach is then easily extensible to choosing more than one point, either because a physical experiment is being done in batches or because an asynchronous parallel computing environment is being used for a computer experiment. Since the points are spread apart, a simple approach to choosing n_b design points for the next batch is to just select the n_b points in the candidate set with the highest $\Delta\hat{\sigma}^2(\tilde{x})$ as per Equation (2). A better approach, if computational resources allow, is to choose the first design point as the one with the highest $\Delta\hat{\sigma}^2(\tilde{x})$, then to add a pseudo-datapoint at this chosen location with value equal to its predictive mean value and to recompute Equation (2), choosing the second point as the one that maximizes this among the remaining points in the candidate set. This approximation has the effect of reducing uncertainty in the local region of the first point, so that the second point will typically be selected from a different part of the input space. This process is iterated until n_b design points are obtained.

1.2.2 Sensitivity Analysis

Global sensitivity analysis (SA; not to be confused with local derivative based analysis) is a resolving of the sources of output variability by apportioning elements of this variation to different sets of input variables (Saltelli et al., 2000). In large engineering problems there can be a huge number of input variables over which the objective is to be optimized, but only a small subset will be influential within the confines of their uncertainty distribution. Thus global SA is an important (but often overlooked) aspect of efficient optimization and it may be performed, at relatively little additional cost, on the basis of a statistical model fit to the initial sample. Variance-based SA methods decompose the variance of the objective function, with respect to an uncertainty distribution placed on the inputs, into variances of conditional expectations. These provide a natural measure of the output association with specific sets of variables and provide a basis upon which the importance of individual inputs may be judged.

We will concentrate on two influential sensitivity indices: the first order for the jth input variable, $S_j = \mathrm{var}\left(\mathrm{E}\left[f(\mathbf{x})|x_j\right]\right)/\mathrm{var}(f(\mathbf{x}))$, and the total sensitivity for input j, $T_j = \mathrm{E}\left[\mathrm{var}\left(f(\mathbf{x})|\mathbf{x}_{-j}\right)\right]/\mathrm{var}(f(\mathbf{x}))$. Here, f denotes the objective function, $\mathbf{x}_{-j}$ is the input vector excluding the jth input, and all expectations and variances are taken with respect to the uncertainty distribution placed on $\mathbf{x}$. The uncertainty distribution may be any probability function defined over the input space, but we will assume that it consists of independent uniform distributions over each (bounded) dimension of the input space. The first order indices measure the portion of variability that is due to variation in the main effects for each input variable, while the total effect indices measure the portion of variability that is

due to total variation in each input. Thus, the difference between T_j and S_j provides a measure of the variability in the objective function due to interaction between input j and the other input variables. A large difference may lead the investigator to consider other sensitivity indices to determine where this interaction is most influential, and this is often a key aspect of the dimension-reduction that SA provides for optimization problems. Refer to Sobol' (1993) and Homma and Saltelli (1996) for a complete discussion of the properties and derivation of variance-based SA indices.

The influential paper by Oakley and O'Hagan (2004) describes an empirical Bayes estimation procedure for the sensitivity indices; however, some variability in the indices is lost due to plug-in estimation of GP model parameters and, more worryingly, the variance ratios are only possible in the form of a ratio of expected values. Likelihood based approaches are proposed by Welch et al. (1992) and in Morris et al. (2008). The technique proposed here is, in contrast, fully Bayesian and provides a complete accounting of the uncertainty involved. Briefly, at each iteration of an MCMC sampler that is taking draws from the TGP posterior, output is predicted over a large (carefully chosen) set of input locations. Conditional on this predicted output, the sensitivity indices can be calculated via Monte Carlo integration. In particular, Saltelli (2002) describes an efficient LHS based scheme for estimation of both first order and total effect indices in such situations, and we follow this technique exactly. That is, the locations chosen for TGP prediction are precisely those prescribed in Saltelli's approach. At each MCMC iteration, after calculating Monte Carlo estimates of the integrals involved conditional on the TGP predicted response, we obtain a posterior realization of the variance indices. The resultant full posterior sample then incorporates variability from both the integral estimation and uncertainty about the function output.

Apart from the variance-related quantities, another common component of global SA is an accounting of the main effects for each input variable, $\mathrm{E}[f(\mathbf{x})|x_j]$ as a function of x_j. These can easily be obtained as a byproduct of the above variance analysis procedure, again through Monte Carlo integration conditional upon the TGP predicted response.

1.2.3 Optimization

In Section 1.2.1, TGP prediction was used to guide the intelligent collection of data, with the goal being to minimize the predictive variance. An alternative goal would be one of optimization – it is not the entire response surface which is of interest, but rather only the minimum (or maximum) response point on this surface. In this case, although TGP prediction still forms the backbone of our inference, we need a different objective function and a different sampling approach.

Statistical methods are useful in optimization problems, particularly where the function being optimized is best treated as an expensive black-box, i.e., each function evaluation is relatively costly to obtain (physically or in computing time), and no additional information about the function (such as parametric form or gradient evaluations) is available. Creation of a statistical surrogate model, or emulator, of the black-box then allows optimization on the cheaper statistical model, reducing the need for expensive full evaluations (Booker et al., 1999). The most direct application of this approach would have statistical prediction fully determine the search for an optimum input configuration, and the most prominent example of this strategy is the Expected Global Optimizer (EGO) algorithm of Jones et al. (1998). The method is designed to search the input space and converge towards the global minimum. At each iteration, a GP is fit to the current set of function evaluations and a new location

for data collection is chosen based upon the GP posterior expectation for the improvement statistics:

$$I(x) = \max\{(f_{best} - f(x)), 0\}, \tag{3}$$

where f_{best} is response corresponding to the current best point in the search. The location with maximum expected improvement is chosen for evaluation, and following evaluation the GP is then fit anew to the data set augmented by these results. Schonlau et al. (1998) provide an extensive discussion of improvement statistics. The key to success for this algorithm is that, in the expectation, candidate locations are rewarded both for a near–optimal mean predicted response as well as for high response uncertainty (indicating a poorly explored region of the input space). Hence, the posterior expectation for improvement provides an ideal statistic to inform intelligent optimization.

For this particular project, our collaborators were interested in not simple global point optimization, but rather robust local optimization. Cost and time constraints often make it infeasible to execute a search of the magnitude required to guarantee global convergence (such as EGO). On the other hand, purely local optimization algorithms will fail on highly multi-modal problems where one can easily get stuck in poor local optima. Furthermore, in engineering applications (such as that discussed herein) it is essential to avoid local solutions on a *knife's edge* portion of the response surface, where small changes in the input lead to large changes in the response.

Thus, the problem at hand may be characterized as robust local optimization – we need to find a solution with a response that is close to the global optimum, while using many fewer iterations than a truly global search would require. The proposed solution is to

combine existing local optimization methods, for quick convergence, with TGP statistical prediction, to give the algorithm a global scope. Briefly, a local optimization search pattern is periodically augmented with locations chosen to maximize the TGP predicted expected improvement.

The TGP generated search pattern consists of m locations that maximize (over a discrete candidate set chosen through some space-filling design) the expected multi-location improvement, $E\left[I(x_1, \ldots, x_m)\right]$, where

$$I(x_1, \ldots, x_m) = \max\{(f_{best} - f(x_1)), \ldots, (f_{best} - f(x_m)), 0\} \tag{4}$$

(Schonlau et al., 1998). Taking a fully Bayesian approach, the improvement $I(\tilde{x})$ is drawn for each $\tilde{x}$ in the candidate set at each iteration of MCMC sampling from the TGP posterior. This offers an improvement over competing algorithms that use only point estimates of the parameters governing the probability distribution around the response surface. Our Bayesian analysis results in a full posterior predictive distribution for the response (and, hence, the improvement) at any desired location in the input space. This full posterior predictive sample is essential to the maximization of the multivariate expected improvement in (4): locations x_1 through x_m maybe chosen iteratively, such that each x_i maximizes the expected i-location improvement conditional on x_1 through x_{i-1} already having been selected. Full posterior samples for the improvement statistics at each $\tilde{x}$ are required to re-calculate the expected improvement conditional on the iteratively expanding list of selected locations. This simplifies what would have itself been a complex optimization problem, and has the added benefit of defining an order to the list of m search locations.

Finally, although our general hybrid optimization approach will work with any local pat-

tern search algorithm, the local optimization scheme used here is the asynchronous parallel pattern search (APPS) (Kolda, 2005; Gray and Kolda, 2006) developed at Sandia National Laboratories. APPS is a derivative-free optimization method that works by evaluating points in a search pattern of decreasing radius around the present best point. Software is publicly available at `http://software.sandia.gov/appspack/`. The primary stopping criterion is based on step length, and APPS is locally convergent under mild conditions. In addition, APPS is an efficient method for finding a local optimum when already in the neighborhood of this optimum. By combining it with a TGP emulator, we can more quickly find the correct neighborhood. Thus we use more points chosen by TGP early in the optimization process, and more points chosen by APPS later in the process. More details of optimization through posterior expected improvement via TGP, as well as the hybrid optimization algorithm with APPS and a parallel implementation, are provided by Taddy et al. (2009).

2 Experimental Design

Calibration of the circuit devices under study consists of a minimization of a loss function for the distance between simulated current amplitude curves and experimental data. Before turning to the optimization problem in the following section, it is necessary to physically test the devices. The goal of this calibration is to obtain a single simulator parameterization for each device that will provide accurate current amplitude curve predictions under a variety of different physical situations. In particular, simulation will be required at different temperatures, for different radiation dosages, and at different DC voltage bias levels for the relevant complex system composed of the circuit devices. Thus the experiment design

should provide as full a picture as possible of the device performance over the space defined by these three physical variables. Although device performance is completely characterized as a current amplitude curve, the peak amplitude provided a univariate output which the experimentalists felt would be representative of the device behavior at different input variable values.

For each of 20 circuit devices, historical testing had yielded a bank (approximately 50 observations per device) of existing data for device current amplitude behavior at various levels of temperature, radiation dosage, and bias. The experimentalists requested a list of 150 additional variable location vectors for testing. Although, in each case, the underlying input variables are continuous (and unbounded over the region of interest), the limitations of physical experimentation reduced the possible input values to three temperature levels, six dosage levels, and five bias levels. Hence, the space of potential experiment input locations consisted of a $3 \times 6 \times 5 = 90$ point grid. These factor levels correspond only to the set-up configuration; actual temperature, dosage, and bias amounts for each experiment are measurable and will be in the neighborhood of the specified input configuration, but will not be exactly the same as designed.

The experiments are performed in batches, with five circuit devices grouped together on each of four different boards. Due to the difficulties inherent in finding a design that is optimal for each batch of five circuits, one device was chosen to be representative of each board and the experiment was designed around this single device. There is random noise in the results, and the data already include replication at individual input vectors. The list for additional testing locations should include replication where necessary to reduce the overall variance of output (peak amplitude) predictions.

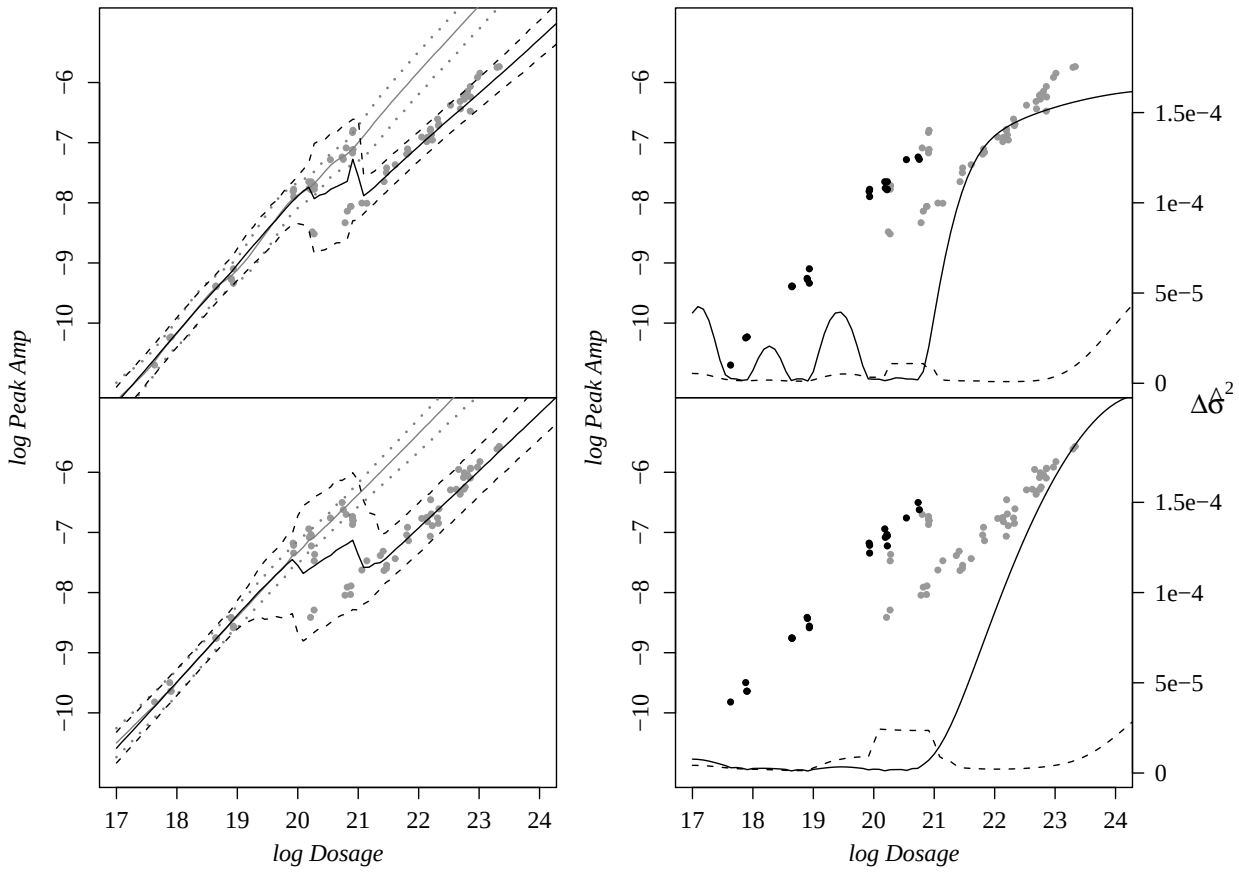

Figure 1: Experiment results for *bft92a* on the top and *bfs17a* on the bottom, given a temperature of 75 Celsius and zero bias. The left hand figures show mean (solid lines) and 90% interval predicted log peak amplitude; grey lines correspond to a TGP fit for the original observations, and black corresponds to a TGP fit to all observations. The right hand figure shows the original data in black and the data obtained through additional testing in grey. Plotted over the data we see the expected posterior reduction in error variance, $\Delta\hat{\sigma}^2$ for this temperature and bias, as a function of log dosage. The solid line refers to the posterior expectation after the original testing, and the dashed line corresponds to posterior expectation conditional on the completed data set.

Our approach to design is an iterative application of the adaptive sampling procedure outlined in Section 1.2.1. The expected reduction in global error shown in (2) is not easily extended to an analogous criterion for combinations of multiple locations. This issue is resolved through the implementation of a greedy algorithm which repeatedly chooses a single new location for testing to be added to an existing list. A first new input vector is chosen exactly as proposed above in Section 1.2.1, through adaptive sampling based on the TGP prior for peak amplitude conditional on temperature, dosage, and usage. A value for realized peak amplitude at this input location is then drawn from the conditional posterior predictive distribution, and this value is used as the imputed output corresponding to a future test at this location. Thus in searching for the second location, we treat the existing data as the combination of the observed data and the one new imputed point, and now look to maximize the expected reduction in global error of this updated dataset. The process is repeated, and at each iteration the treed GP model is fit to the existing data augmented by randomly imputed output values at all of the locations already chosen for future testing. All of the existing imputed output values are re-drawn from their conditional posterior predictive distributions at each iteration. This additional variability helps to account for the variability in the physical observations (in contrast to the typically deterministic behavior of a computer simulator). This iterative adaptive sampling algorithm is used to provide an ordered list of 150 locations (including repetition) for additional testing of each of the 20 devices. The prioritization implied by the ordering of the list is especially valuable in the motivating example, as the expense of individual experiments is unknown in advance and experimentation is terminated once the study has reached a predetermined budget constraint. At some point, a batch of experimental data becomes available, and all of the placeholder

values are replaced by the new real data, and the iterative process continues for the next round of physical experiments.

Results for two of the devices, *bft92a* and *bfs17a*, are shown in Figure 1. The predictions shown in the left hand figure expose a discontinuity in the log peak amplitude surface which occurs for log dosages between 20 and 21. Although the exhibited results are conditional on a temperature of 75 Celsius and a bias of zero, similar behavior was discovered at other parameter configurations. The posterior mean for error reduction, $\Delta\hat{\sigma}^2(x)$ as defined in (2), from additional testing at a single new point x, is plotted in the right hand panel. We see that $\Delta\hat{\sigma}^2(x)$ is substantially reduced following the additional testing, with significant room for variance reduction only for x in the zone of discontinuity and at the boundaries. Finally, we note that the complex surfaces shown in the left hand figure mean that significant modeling gains are available by using treed Gaussian processes instead of more traditional stationary models.

3 Calibrating the Computer Model

The second part of this project focused on calibration and validation of a computer simulation model for the circuits. Numerical simulation is increasingly used because of advances in computing capabilities and the rising costs associated with physical experiments. In this context the computer models are often treated as an objective function to be optimized, and this is how they were treated by our collaborators. The challenges inherent in implementing this optimization are characterized by an inability to calculate derivatives and by the expense of obtaining a realization from the objective function. Fast convergence of the optimizer is

needed because of the cost of simulation, and a search of the magnitude required to guarantee global convergence is not feasible. However, it is important to be aware that these large engineering problems typically have multi-modal objective functions, so we want to avoid converging to low quality solutions. Thus we combine the local optimization method APPS for quick convergence with TGP to help provide a more robust solution, as described in Section 1.2.3.

The goal here is to find tuning parameter values for the Xyce simulator that lead to predicted current output that is as close as possible to the results obtained in the physical experiments. Here "close" is defined through a squared-error objective function:

$$f(\mathbf{x}) = \sum_{i=1}^{N} \frac{1}{T_i} \sum_{t=1}^{T_i} \left[(S_i(t;\mathbf{x}) - E_i(t))^2 \right]. \tag{5}$$

where N is the number of physical experimental runs (each corresponding to a unique radiation pulse), T_i is the total number of time observations for experiment i, $E_i(t)$ is the amount of electrical current observed during experiment i at time t, and $S_i(t;\mathbf{x})$ is the amount of electrical current at time t as computed by the simulator with tuning parameters $\mathbf{x}$ and radiation pulse corresponding to experiment i. Since each physical experiment may result in a different number of usable time observations, the weights of the errors are standardized so that each experimental run is counted equivalently. We note that the traditional statistical approach would also include a term for model discrepancy. However, our collaborators did not want such a term, as the mathematical modelers want to know what the best fit is for their model, and then they intend to address any remaining discrepancies by updating the physics in their model.

Because of the need to do both calibration and validation, only six experimental runs

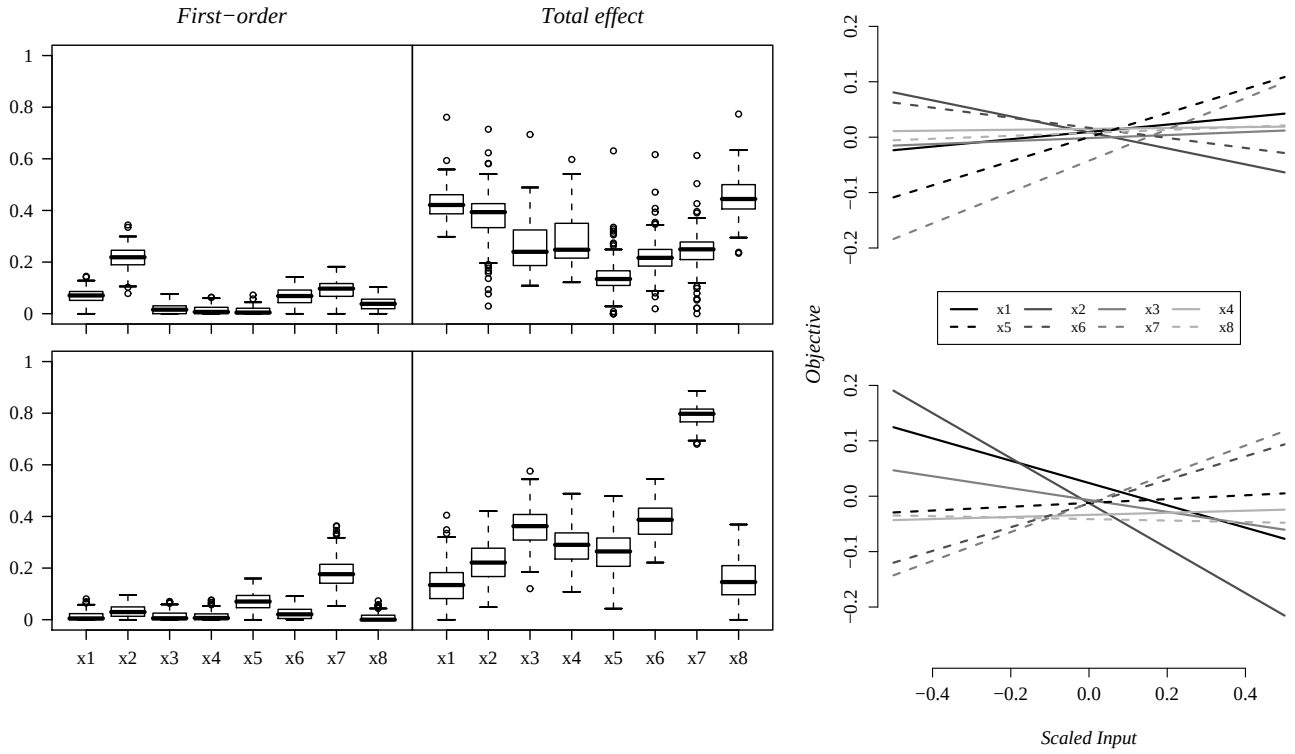

Figure 2: Sensitivity analysis for *bft92a* (top) and *bfs17a* (bottom) optimization objective functions, summarized by posterior distributions for the first order (left), total (middle) sensitivity indices, and posterior mean main effects (right).

were used for each calibration, with the remaining datapoints saved for the validation stage. For each circuit device and each temperature setting, the six points were chosen to be representative of the whole set of data collected, and were selected by fitting a six component mixture model. The selection details are not needed herein, we just treat the six points as the available data, but if the reader is interested, the details are given by Lee et al. (2009).

The simulator involves 38 user-defined tuning parameters for modeling current output as a function of radiation pulse input. Through discussion with experimentalists and researchers

familiar with the simulator, 30 of the tuning parameters were fixed in advance to values either well known in the semiconductor industry or determined through analysis of the device construction. The semiconductor engineers also provided informative bounds for the remaining eight parameters. It is these eight parameters which are the inputs for our objective function (5). These parameters include those that are believed to have both a large overall effect on the output of the model and a high level of uncertainty with respect to their ideal values. Figure 2 shows the results of an MCMC sensitivity analysis, as described in Section 1.2.2, based on a TGP model fit to an initial Latin Hypercube Sample (LHS) of 160 input locations and with respect to a uniform uncertainty distribution over the bounded parameter space. All of the eight parameters appear to have significant effect on the objective function variability, although the main effect and first-order plots indicate that some variables are only effective in interaction with the other inputs. These higher-order interactions create challenges for optimization. In addition, the posterior mean main effect plots alerted the researchers to the possibility of optimal solutions on the boundaries of the input space (especially for x_2 and x_7) and will provide valuable guidance for checking the validity of the calibrated simulator.

The objective function (5) was optimized using both APPS by itself and the hybrid algorithm TGP-APPS. In the case of the hybrid algorithm, a LHS of 160 points was used to provide and initial fit for the TGP. The wall clock time and the number of objective function evaluations corresponding to each device and each optimization algorithm are shown in Table 1. Figure 3 shows simulated current response curves corresponding to each solution and to the initial guess for tuning parameter values, as well as the data, for a single radiation pulse input to each device. Results for the other radiation pulse input values exhibit similar

properties.

In the case of *bft92a*, the solutions produced by the two optimization algorithms are practically indistinguishable (they appear on top of each other in the figure). However, the APPS solution required over seven times as may functional evaluations, leading to much additional computational expense and elapsed time. The gain of the hybrid algorithm here is in its ability to move the search pattern quickly into better areas of the input space. We note that even if we started with the hybrid algorithm without an initial LHS (and starting from the same initial parameter vector as for APPS), it only takes about two more hours (a total of 15.8 hours) to obtain an equivalent solution through TGP-APPS – still a huge gain over APPS alone.

For the *bfs17a* device, the difference in the resulting response curves is striking and illustrates the desirable robustness of our hybrid algorithm. The response curve created using the parameter values obtained by APPS alone differs significantly from the data in overall shape. In contrast, the curve resulting from the parameters found by TGP-APPS is a reasonable match to the experimental data. These results suggest that the APPS algorithm was unable to overcome a weak local minimum while the inclusion of TGP allowed for a more comprehensive search of the design space. Note that the time results support this, as they show that the APPS algorithm converged relatively quickly. The extra computational cost of TGP-APPS is well justified by the improvement in fit. Of course, it is still clear that the simulator is not completely matching the data, and at this point we suspect that there is an inherent bias in the simulator. A complete statistical calibration would thus require the modeling of a bias term, as in the work of Kennedy and O'Hagan (2001). However, for our purposes in this project on optimum control, both our collaborating modelers and

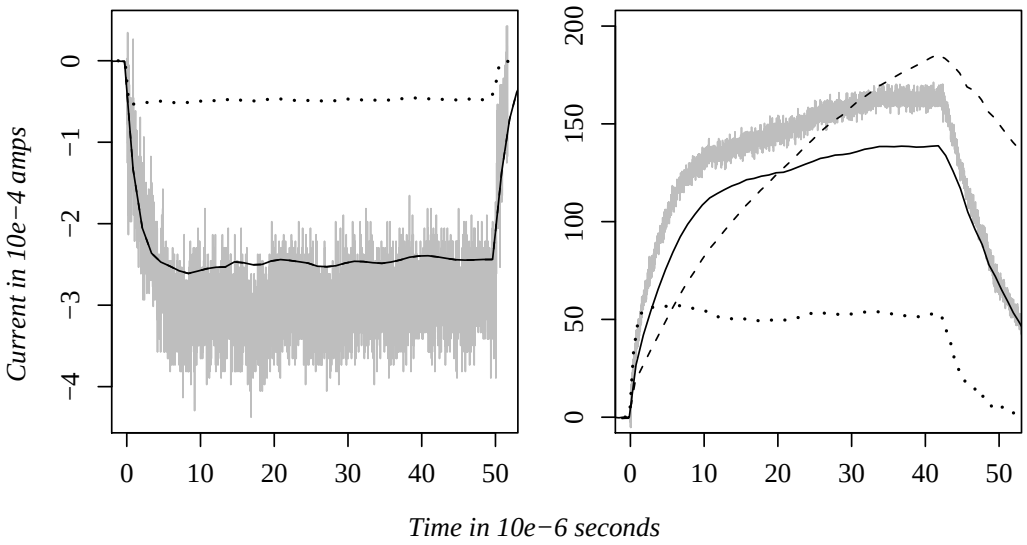

Figure 3: Simulated response current curves for the *bft92a* (left) and *bfs17a* (right) devices. The solid line shows the response for parameters found using TGP-APPS, the dashed line for parameters found through APPS alone, and the dotted line for the initial parameter vector guess. The experimental current response curves for the radiation impulse used in these simulations are shown in grey.

Device	Method	Evaluations	Time (hours)
bft92a	APPS	6823	94.8
bft92a	APPS-TGP	962	13.8
bfs17a	APPS	811	10.3
bfs17a	APPS-TGP	1389	18.1

Table 1: For each bjt device and each optimization algorithm, the number of objective function evaluations and total wall clock time required to find a solution.

experimentalists were quite happy with the robust solution with respect to minimization of the provided objective function that TGP-APPS was able to find.

4 Further Discussion

We have illustrated how the treed Gaussian Process (TGP) model can be useful for spatial data and semiparametric regression in the context of a computer experiment for designing a circuit device. We have seen how the model can be used towards sequential design of (computer) experiments (via Bayesian adaptive sampling), sequential robust local optimization (with the help of APPS), validation, calibration, and sensitivity analysis all by simply sampling from the posterior distribution. In both optimization and experiment design, full posterior sampling combined with recursive iteration allowed us to use univariate prediction to optimize multivariate criteria.

Some of the models and methods described herein have also been used to design a computational fluid dynamics computer experiment for a rocket booster at NASA (Gramacy and Lee, 2009), and have been validated as competitive regression and spatial models on numerous synthetic and real data sets (Gramacy and Lee, 2008a,b; Gramacy, 2007). TGP is indeed a flexible model with many potential applications. However, one limitation is that the current methodology only supports real-valued inputs and responses. An adaptation of these methods to support categorical inputs and outputs promises to be a fruitful future direction of research.

Allowing categorical inputs will widen the scope of regression and design applications that can be addressed by the model. While the GP part of the model can easily handle binary-

encoded categorical inputs on its own, it represents a sort of overkill. For example, a separable correlation function with width parameter d_i will tend to infinity if the output does not vary with binary input x_i, and will tend to zero if it does. Clearly, this functionality is more parsimoniously served by partitioning, e.g., using a tree. However, in a TGP implementation the tree will never partition on the binary inputs because doing so would cause the resulting design matrices at the leaves to be rank deficient. So without special care, any benefits of a divide-and-conquer approach (e.g., speed) to a non-parametric (TGP) regression with categorical inputs are lost. Once a careful implementation has been realized, one can imagine many further extensions. For example, including latent variable categorical inputs could enable the model to be used for clustering.

Extending the methodology to handle categorical responses will allow TGP to be applied to problems in classification. Separately, treed models and GP models have enjoyed great success in classification problems. Adapting treed models for classification (e.g., CART) is straightforward, whereas adapting GP models is a bit more complicated, requiring the introduction of $O(nk)$ latent variables where k is the number of classes. A combined modeling approach via TGP has the potential to be as fruitful for classification as it is for regression. It will be exciting to see how this extension develops, as well as accompanying methods for adaptive sampling, optimization, validation, calibration, and sensitivity that can be developed along side.

Appendices

A. Broader Context and Background

A1. Treed Gaussian Processes

Here we provide more background on treed Gaussian processes, with further details available in Gramacy and Lee (2008a). First, the structure of the tree. We partition the input space using a tree, along the lines of models such as CART Breiman et al. (1984). Such a tree is constructed with a series of binary recursive partitions. For example, in two dimensions, one might consider an input space on [0,1] X [0,1]. The first split could be at $X_1 = .4$ separating the space into two rectangles, [0,0.4] X [0,1] and [0.4,1] X [0,1]. The second split might be at $X_2 = .3$ in the first partition, thus creating a third partition. Note that this second split does not affect the [0.4,1] X [0,1] region. By allowing multiple splits on the same variable, any arbitrary axis-aligned partitioning structure can be achieved. The restriction of axis-alignment allows us to build models in a computationally efficient manner, without losing too much modeling flexibility. Arbitrary partitions would require significantly more computing resources. We denote the whole tree structure by $\mathcal{T}$ and the leaf nodes by $\eta \in \mathcal{T}$, each of which represents a region of the input space. A prior for the tree is defined through a growth process. We start with a null tree (no partitions, all of the data is together in a single leaf node). Each leaf node η splits with probability $a(1 + q_\eta)^{-b}$, where q_η is the depth of $\eta \in \mathcal{T}$ and a and b are parameters chosen to give an appropriate size and spread to the distribution of trees. Further details are available in Chipman et al. (1998, 2002). Here we use the default values of $a = 0.5$ and $b = 2$ from the R package (Gramacy and Taddy, 2008).

The tree recursively partitions the input space into into R non-overlapping regions $\{r_\nu\}_{\nu=1}^R$. Each region r_ν contains data $D_\nu = \{x_\nu, Z_\nu\}$, consisting of n_ν observations.

Let $m \equiv m_X + 1$ be number of covariates in the design (input) matrix X plus an intercept. For each region r_ν, the hierarchical generative GP model is

$$\begin{aligned} Z_\nu|\beta_\nu, \sigma_\nu^2, K_\nu &\sim N_{n_\nu}(f_\nu\beta_\nu, \sigma_\nu^2 K_\nu) & \beta_0 &\sim N_m(\mu, B) \\ \beta_\nu|\sigma_\nu^2, \tau_\nu^2, W, \beta_0 &\sim N_m(\beta_0, \sigma_\nu^2\tau_\nu^2 W) & \tau_\nu^2 &\sim IG(\alpha_\tau/2, q_\tau/2), \\ \sigma_\nu^2 &\sim IG(\alpha_\sigma/2, q_\sigma/2) & W^{-1} &\sim W((\rho V)^{-1}, \rho) \end{aligned}$$

with $F_\nu = (1, X_\nu)$, and W is an $m \times m$ matrix. The N, IG, and W are the (Multivariate) Normal, Inverse-Gamma, and Wishart distributions, respectively. Hyperparameters $\mu, B, V, \rho, \alpha_\sigma, q_\sigma, \alpha_\tau, q_\tau$ are treated as known, and we use the default values in the R package. This model specifies a multivariate normal likelihood with linear trend coefficients β_ν, which are also modeled hierarchically. Each region is fit independently (conditional on the hierarchical structure), which gives this approach some similarity to change-point models.

The GP correlation structure within each region is given by $K_\nu(x_j, x_k) = K_\nu^*(x_j, x_k|d) + g_\nu\delta_{j,k}$, where K_ν^* is the separable power family given in Equation (1) and g is the nugget term. Our choice of priors encodes a preference for a model with a nonstationary global covariance structure, giving 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:

$$p(d, g) = p(d) \times p(g) = p(g) \times \frac{1}{2}[Ga(d|\alpha = 1, \beta = 20) + Ga(d|\alpha = 10, \beta = 10)]. \quad (6)$$

We take the prior for g to be exponential with rate λ.

In some cases, a full GP may not be needed within a partition; instead a simple linear

model may suffice. Because of the linear mean function in our implementation of the GP, the standard linear model can be seen as a limiting case. The linear model is more parsimonious, as well as much more computationally efficient. We augment the parameter space with indicator variables $b = \{b\}_{i=1}^{m_X} \in \{0,1\}^{m_X}$. The boolean b_i selects either the GP ($b_i = 1$) or its limiting linear model for the i^{th} dimension. The prior for b_i specifies that smoother GPs (those larger range parameters d_i) are more likely to jump to the limiting linear model:

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

and we use the R package default values of $(\gamma, \theta_1, \theta_2) = (10, 0.2, 0.95)$. More details are available in Gramacy and Lee (2008b).

A2. Experimental Design

The basic ideas for experimental design in computer experiments follow those of standard experimental design, i.e., one wants a relatively small set of points that are expected to provide maximal information about parameters for a particular choice of model. For a GP, as with most models, one generally wants to spread out the points, as each observation gives a fair amount of local information because of the smoothness properties of the GP model. Approaches include maximin distance, Latin hypercube, D-optimal, maximum entropy, and orthogonal array designs (McKay et al., 1979; Santner et al., 2003). In general, no replications are planned when the computer simulator is deterministic.

With computer experiments, it is natural to move on to sequential collection of data, i.e., Sequential Design of Experiments (DOE) or Sequential Design and Analysis of Computer Experiments (SDACE) (Sacks et al., 1989; Currin et al., 1991; Welch et al., 1992). Depending

on whether the goal of the experiment is inference or prediction, the choice of utility function will lead to different algorithms for obtaining optimal designs (Shewry and Wynn, 1987; Santner et al., 2003).

In the machine learning literature, sequential design of experiments is often referred to as Active Learning. Two approaches applied to Gaussian processes are that of Cohn (1996) described in the main text (maximizing the expected reduction in average squared error) and that of MacKay (1992), which chooses the new point as that with the largest standard deviation in predicted output. While the Mackay approach is simpler, it is also more localized, and less useful in the presence of heteroskedasticity.

B. Computations

We fit these models using the `tgp` package in R (Gramacy and Taddy, 2008). A tutorial is provided by Gramacy (2007). The core of the package is based on C++ code that employs reversible jump Markov chain Monte Carlo to fit both the tree structure and the GPs in each of the partitions. By averaging across the Markov chain realizations, estimates of the posterior mean and of predictive intervals are obtained. This averaging includes the tree structures, and as a result, we typically obtain smooth posterior mean fits because of mixing over the location of partitions.

Conditional on a tree structure, most parameters can be updated via Gibbs sampling. The linear regression parameters β_ν and their prior mean β_0 all have multivariate normal full conditionals. The data variance parameter σ^2 and the linear variance parameter τ^2 are both conditionally inverse-gamma, and the linear model covariance matrix W is conditionally

inverse-Wishart. Correlation parameters d and g require Metropolis-Hastings updates. The tree structure itself is updated with reversible jump steps: grow, prune, change, swap, and rotate. The first two require care in accounting for the change of dimension, while the latter three are straightforward Metropolis-Hastings steps. More details on estimation and prediction are available in Gramacy and Lee (2008a). It can be helpful to standardize the data before running the R code, so that the default parameter values are reasonable.

References

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

Booker, A. J., J. E. Dennis, P. D. Frank, D. B. Serafini, V. Torczon, and M. W. Trosset (1999). A rigorous framework for optimization of expensive functions by surrogates. *Structural and Multidisciplinary Optimization 17*, 1–13.

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

Chaloner, K. and I. Verdinelli (1995). Bayesian experimental design, a review. *Statistical Science 10 No. 3*, 273–304.

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

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

Cohn, D. A. (1996). Neural network exploration using optimal experimental design. In

Advances in Neural Information Processing Systems, Volume 6(9), pp. 679–686. Morgan Kaufmann Publishers.

Cressie, N. A. C. (1993). *Statistics for Spatial Data, revised edition*. John Wiley & Sons.

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

Fjeldy, T. A., T. Ytterdal, and M. S. Shur (1997). *Introduction to device modeling and circuit simulation*. Wiley-Interscience.

Gramacy, R. B. (2007). `tgp`: An `R` package for Bayesian nonstationary, semiparametric nonlinear regression and design by treed gaussian process models. *Journal of Statistical Software 19*(9).

Gramacy, R. B. and H. K. H. Lee (2008a). Bayesian treed Gaussian process models with an application to computer modeling. *Journal of the American Statistical Association 103*, 1119–1130.

Gramacy, R. B. and H. K. H. Lee (2008b). Gaussian processes and limiting linear models. *Computational Statistics and Data Analysis 53*, 123–136.

Gramacy, R. B. and H. K. H. Lee (2009). Adaptive design and analysis of supercomputer experiments. *Technometrics*. to appear.

Gramacy, R. B. and M. A. Taddy (2008). *tgp: Bayesian treed Gaussian process models*. R package version 2.1-2.

Gray, G. A. and T. G. Kolda (2006). Algorithm 856: APPSPACK 4.0: Asynchronous parallel pattern search for derivative-free optimization. *ACM Transactions on Mathematical Software 32*(3), 485–507.

Gray, G. A., M. Martinez-Canales, C. Lam, B. E. Owens, C. Hembree, D. Beutler, and C. Coverdale (2007). Designing dedicated experiments to support validation and calibration activities for the qualification of weapons electronics. In *Proceedings of the 14th NECDC*. also available as Sandia National Labs Technical Report SAND2007-0553C.

Green, P. (1995). Reversible jump markov chain monte carlo computation and Bayesian model determination. *Biometrika 82*, 711–732.

Homma, T. and A. Saltelli (1996). Importance measures in global sensitivity analysis of nonlinear models. *Reliability engineering and system safety 52*, 1–17.

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

Keiter, E. R. (2004). Xyce parallel elctronic simulator design: mathematical formulation. Technical Report SAND2004-2283, Sandia National Labs, Albuquerque, NM.

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

Kolda, T. G. (2005, December). Revisiting asynchronous parallel pattern search for nonlinear optimization. *SIAM Journal of Optimization 16*(2), 563–586.

Lee, H. K. H., M. Taddy, and G. A. Gray (2009). Selection of a representative sample. *Journal of Classification*. to appear.

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

McKay, M. D., W. J. Conover, and R. J. Beckman (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.

Morris, R. D., A. Kottas, M. Taddy, R. Furfaro, and B. Ganapol (2008). A statistical framework for the sensitivity analysis of radiative transfer models used in remote sensed data product generation. *To appear in the IEEE Transactions on Geoscience and Remote Sensing*.

Oakley, J. and A. O'Hagan (2004). Probabilistic sensitivity analysis of complex models: a Bayesian approach. *Journal of the Royal Statistical Society, Series B 66*, 751–769.

Oberkampf, W. L., T. G. Trucano, and C. Hirsch (2003). Verification, validation, and predictive capability. Technical Report SAND2003-3769, Sandia National Labs, Albuquerque, NM.

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

Saltelli, A. (2002). Making best use of model evaluations to compute sensitivity indices. *Computer Physics Communications 145*, 280–297.

Saltelli, A., K. Chan, and E. Scott (Eds.) (2000). *Sensitivity Analysis*. John Wiley and Sons.

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

Schonlau, M., D. Jones, and W. Welch (1998). Global versus local search in constrained optimization of computer models. In *New Developments and applications in experimental design*, Number 34 in IMS Lecture Notes - Monograph Series, pp. 11–25.

Sedra, A. S. and K. C. Smith (1997). *Microelectronic Circuits* (Fourth ed.). Oxford University Press.

Shewry, M. and H. Wynn (1987). Maximum entropy sampling. *Journal of Applied Statistics 14*, 165–170.

Sobol', I. M. (1993). Sensitivity analysis for nonlinear mathematical models. *Mathematical Modeling and Computational Experiment 1*, 407–414.

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

Taddy, M., H. K. H. Lee, G. A. Gray, and J. D. Griffin (2009). Bayesian guided pattern search for robust local optimization. *Technometrics*. to appear.

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