Bayesian methodology for latent function modeling in applied physics and engineering

by

Michael Grosskopf

B.Sc., University of Michigan, 2005

Dissertation Submitted in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy

in the
Department of Statistics and Actuarial Science
Faculty of Science

SIMON FRASER UNIVERSITY
Fall 2017

Approval

Name: **Michael Grosskopf**

Degree: **Doctor of Philosophy (Statistics)**

Title: ***Bayesian methodology for latent function modeling in applied physics and engineering***

Examining Committee: **Chair:** Jinko Graham
Professor

Derek Bingham
Senior Supervisor
Professor

Dave Campbell
Supervisor
Associate Professor

Liangliang Wang
Supervisor
Assistant Professor

Richard Lockhart
Internal Examiner
Professor

Roshan Joseph Vengazhiyil
External Examiner
Professor
Industrial & Systems Engineering
Georgia Tech University

Date Defended: 20 December 2017

Abstract

Computer simulators play a key role in modern science and engineering as a tool for understanding and exploring physical systems. Calibration and validation are important parts of the use of simulators. Calibration is a necessary part of assessing the predictive capability of the model with fully quantified sources of uncertainty. Field observations for physical systems often have diverse types. New methodology for calibration with generalized measurement error structure is proposed and applied to the parallel deterministic transport model for the Center for Exascale Radiation Transport at Texas A&M University. Validation of computer models is critical for building trust in a simulator. We propose a new methodology for model validation using goodness-of-fit hypothesis tests in a Bayesian model assessment framework. Lastly, the use of a hidden Markov model with a particle filter is proposed for detection of anomalies in time series for the purpose of identifying intrusions in cyber-physical networks.

Keywords: Computer Model Calibration; Bayesian; Computer Experiments; Verification and Validation; Particle Filtering

Dedication

To my amazing wife, Donna Marion, whose support and care drives me constantly. It would take a thousand lifetimes to show just a fraction of the appreciation you deserve. To my parents Mike and Patti and my siblings Mickayla and Patrick. I always want to make you all proud.

Acknowledgements

I would like to thank my advisor Dr. Derek Bingham for sparking my initial interest in statistics, for taking a chance on the potential he saw in me, and for being an example of a researcher I'd strive to emulate.

Many thanks to the faculty in the Statistics and Actuarial Sciences Department at Simon Fraser. You foster an incredible learning environment and I've appreciated every chance I've had for discussion with you. I could not have imagined a better place to do my doctoral work and hope I live up to the excellent standards of the department.

I'd like to thank the entire Center for Exascale Radiation Transport team at Texas A&M University for providing an interesting application and data to work with, as well as for giving the opportunity to present my research at project reviews and to do an internship at Los Alamos National Lab. This material is supported by the Department of Energy, National Nuclear Security Administration, under Award Number(s) DE-NA0002376.

I would also like to thank Dr. R. Paul Drake at the University of Michigan for the opportunity to do research in laboratory astrophysics and the doors that working with him have opened for my career and life. He gave me the chance to learn and grow as a researcher and develop as a person, all the while being a fantastic mentor and friend. Most relevant to this thesis, I never would have been connected with Dr. Bingham and Simon Fraser University if not for the work with Dr. Drake on the CRASH project.

For fear of leaving off any other specific names, I would like to wish a broad thanks to everyone who has helped me on my journey to this point, whether academically or as a friend or family member who has brought joy to my life. I appreciate everything you all have done for me.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Many systems in physics and engineering involve understanding complex relationships between the inputs to the system and the observable outputs of interest. Explicit modeling of these relationships as indirectly observable, latent response functions is valuable for understanding the processes governing the system and for making predictions. For instance, in computer experiments, the response function of a physical system can be modeled as a combination of the computer model (or simulator) and systematic bias between the simulator and experiment. Understanding this latent discrepancy function is important to calibration and prediction (Kennedy and O'Hagan, 2001; Higdon *et al.*, 2004). This thesis covers the modeling of latent functions with two broad contexts. In chapters 3 and 4, the latent function modeling is for calibration and validation of computer experiments, while Chapter 5 is focused on the estimation of a latent function describing the behavior of periodic time series for purposes of intrusion detection.

The work in Chapters 3 and 4 is motivated by a collaboration with the Center for Exascale Radiation Transport (CERT) at Texas A&M University. The parallel deterministic transport (PDT) code is used at CERT for prediction of radiation and particle transport in complex geometries (Adams *et al.*, 2013). Experiments with an Americium-Beryllium neutron source and graphite bricks are modeled with PDT as part of the work with CERT. The goals of CERT include: (i) calibration of model parameters of PDT to improve the predictive model; (ii) calibration of the level of impurity in each graphite brick and the distribution of impurity level from brick-to-brick; and (iii) validation of PDT on a series of experiments of increasing complexity.

Background on computer model calibration with Gaussian process emulators is given in Chapter 2 in order to set the stage for the new developments in Chapters 3 and 4. The chapter presents the model proposed by Kennedy and O'Hagan (2001) for Bayesian model calibration with unknown model discrepancy. Background on Gaussian process regression is also outlined in Chapter 2.

In Chapter 3, a new approach to calibration of computer models is developed to handle the structure of count data from the CERT experiments. The proposed Bayesian framework generalizes the work by Kennedy and O'Hagan (2001) to non-Gaussian responses. To facilitate this method, a strategy for sampling the latent mean function is presented. The proposed approach is demonstrated using the CERT application and synthetic examples.

Chapter 4 moves from calibration to validation of computer models. The foundation for use of PDT in scientific applications is the trust that the model is able to accurately reproduce physical observations. Assessment of validity involves accounting for uncertainty due to stochastic variation inherent in the data and that due to lack of knowledge of aspects of the computer model (parameters, the response function of the simulator, etc). If the simulator is able to predict the true response of the physical system, we would expect that the distribution function of the simulator output predicted by propagating sources of stochasticity through the computer model would be the same as that of the observed data. We propose a new methodology for assessment of the validity of computer models combining methods for Bayesian model assessment and frequentist goodness-of-fit testing. The proposed procedure mitigates some issues with existing methodology in the literature and provides both qualitative and quantitative assessment of the model validity.

The motivating application changes from CERT to intrusion detection in engineered control systems in Chapter 5. Engineered systems (e.g. the power grid) are a critical part of the infrastructure in the United States and Canada. The safe operation of these facilities is coordinated with control systems involving networked computers coupled to the physical infrastructure. These computers can be vulnerable to intrusion and attack which threaten the stable operation of the system. Anomalous behavior in the system may be an indication of intrusion, so methods for quickly detecting these anomalies are critical. In Chapter 5, a method combining a hidden Markov Model for time series alignment, particle filtering, and on-line quality monitoring is proposed to identify anomalous behavior. The hidden Markov model is used to estimate the shared latent function describing the behavior of the time series, taking into account the varied timing of structure in the series. The method is demonstrated on synthetic examples, as well as data from the heating, ventilation, and air conditioning system of a large office building.

Chapter 2

Background on Model Calibration and Gaussian Processes

The use of large-scale computational models is playing a growing role in major science and engineering research. The simulators are critical tools for developing understanding of complex physical systems. They attempt to encode the entirety of our knowledge of physical processes mathematically and use numerical algorithms to solve these equations and approximate a physical system response. For experimenters, the simulators can be used to meet a variety of goals e.g. attempting to gain qualitative insight into the dynamics that arise in a system, estimating the risk of a quantity of interest reaching a desired (or undesired) threshold, or indicating how a system responds to changes in model inputs to assist in designing a planned experiment.

Typically, a physical system has some set of inputs that govern the behavior of the model and a quantity of interest (QoI) - the measurable outcome of the observed system or experiment. In order to be modeled, a physical system or experiment must be converted to a digital representation in the simulator and the numerical methods for running the simulator must be configured for this representation. The simulator requires setting different types of parameters: (i) physical constants and other values relevant to the underlying mathematical model (ii) values needed for evaluation of the model that may not be physical but may be needed in the software implementation of the model, and (iii) parameters required for the numerical methods to run the simulator. If the parameters are related to the quality of the numerical method to approximate the underlying mathematics of the system, (e.g. spatial or temporal discretization parameters), typically testing the quality of this approximation falls into the process of model verification which will be discussed in Chapter 3. For parameters of type (i) and (ii), model calibration is the process of estimating the values that make the model consistent with the observations. Loeppky *et al.* (2006) separated these into calibration and tuning parameter respectively, however for the remainder of this work we will not distinguish the two, instead treating both as calibration parameters.

In this thesis, the uses of the simulator range from computer model calibration to assessing the model's ability to capture the important features of the system's response (i.e., model validation). Many simulators can be computationally demanding, with only a limited number evaluations may be feasible. Consequently, for model calibration and validation, a statistical emulator of the simulator is required (Sacks *et al.*, 1989). Frequently Gaussian processes are used for this task. An introduction to Gaussian processes is covered in Section 2.1. Background for computer model calibration is then presented in Section 2.2.

2.1 Background on Gaussian Processes

A Gaussian process (GP) is a type of nonlinear regression model that has a wide range of application (Rasmussen and Williams, 2006). Denote the d-dimensional vector of model inputs as $\mathbf{x}$ and the response of the computer model at $\mathbf{x}$ as $Y(\mathbf{x})$. A GP is specified by a mean function, $m(\mathbf{x})$, and a covariance function, $\frac{1}{\kappa}r(\mathbf{x_i}, \mathbf{x_j})$, that defines how deviations from the mean function are correlated at locations in the input space. The covariance function is often specified as the product of the inverse of a precision parameter, κ, and a correlation function, $r(\mathbf{x_i}, \mathbf{x_j})$. The model specified by the GP is:

$$\begin{aligned} Y(\mathbf{x}) &= m(\mathbf{x}) + Z(\mathbf{x}), \\ E[Y(\mathbf{x})] &= m(\mathbf{x}), \\ Var[Y(\mathbf{x})] &= \frac{1}{\kappa}, \text{ and} \\ Cov[Y(\mathbf{x}_i), Y(\mathbf{x}_j)] &= \frac{1}{\kappa} r(\mathbf{x_i}, \mathbf{x_j}). \end{aligned} \tag{2.1}$$

The correlation function specifies the properties (e.g. smoothness, periodicity, stationarity) of members of the function space and determines the distribution on this space. A stationary covariance function is one where the covariance depends only on the distance between input locations (Rasmussen and Williams, 2006). In this work, we will focus on the squared-exponential correlation function, commonly used for emulating computer models (Sacks *et al.*, 1989; Higdon *et al.*, 2004; Rasmussen and Williams, 2006). That is:

$$r(\mathbf{x_i}, \mathbf{x_j}) = \prod_{\ell=1}^{d} e^{-\beta_\ell (x_{i,\ell} - x_{j,\ell})^2}, \tag{2.2}$$

where d is the number of input dimensions in the model and $\beta_\ell > 0$. A value of $\beta_\ell = 0$ indicates that the response is insensitive to variation in the ℓth covariate.

The predictive distribution at $\mathbf{x}^*$, conditional on observed data $\{\mathbf{x}_i, \mathbf{y_i}; i = 1, ..., n_{obs}\}$, is Gaussian with (Jones *et al.*, 1998):

$$E[Y(\mathbf{x}^*) \mid y_1, ..., y_n] = \frac{1}{\kappa} r(\mathbf{x}^*, \mathbf{x})^T \Sigma_{\mathbf{x}}^{-1} (\mathbf{y} - m(\mathbf{x})), \text{ and} \tag{2.3}$$

$$Var[Y(\mathbf{x}^*) \mid y_1, ..., y_n] = \Sigma_{\mathbf{x}^*} - \frac{1}{\kappa^2} r(\mathbf{x}^*, \mathbf{x})^T \Sigma_{\mathbf{x}}^{-1} r(\mathbf{x}^*, \mathbf{x}). \tag{2.4}$$

Here, $\Sigma_{\mathbf{x}}$ is the covariance matrix of the observed data, $\Sigma_{\mathbf{x}^*}$ is the covariance matrix of the new observations at new covariate values, κ is the precision parameter, and $r(\mathbf{x}^*, \mathbf{x})$ is the $n_{obs} \times n_{new}$ matrix of correlations between the observed outcomes and n_{new} new values. The predicted mean in Equation 2.3 can be viewed as a weighted mean of the observed data, where the weights depend on the correlations between observations at the predicted locations and the observed locations.

A common choice of mean function for the GP emulator is to center the observed data by subtracting the sample mean of the observations and use a constant function $m(\mathbf{x}) = 0$ (Higdon *et al.*, 2008; Goh *et al.*, 2013). We will use this approach to modeling the mean function throughout this work.

GPs have the beneficial property of interpolating at observed inputs (Jones *et al.*, 1998), however, for computational stability, a small constant, often called a "nugget", is added to the diagonal of the covariance matrix for the emulator (Gramacy and Lee, 2012). While the use of the "nugget" relaxes the ability of a GP to strictly interpolate, the nugget can be controlled to allow the emulator to approximately interpolate to as high precision as numerical stability will allow.

GPs are particularly well-suited for the task of emulating deterministic computer model output due to the flexibility in modeling complex response surfaces, the ability to interpolate observed values and to provide a foundation for assessment of uncertainty in the model response at unobserved inputs Sacks *et al.* (1989); Jones *et al.* (1998). Kennedy and O'Hagan (2000) propose a GP as a fast emulator for computationally-intensive computer models including joint prediction. They then propose to use this emulation in the process of model calibration in Kennedy and O'Hagan (2001). Calibration will be the focus of later discussion in 2.2 and beyond.

Gramacy and Lee (2008) developed a treed GP method for regression and emulation of simulators in order to accommodate non-stationarity. This worked by fitting independent GPs to different partitions of the input space. Ba *et al.* (2012) propose to use a composite of two Gaussian processes with a spatially-varying variance model for non-stationarity and heteroscedasticity. The composite Gaussian processes capture variation at different scales similar to methods proposed in Ferreira and Lee (2007). Gramacy and Apley (2015) propose a method for extending the use of GPs in computer emulation to large data environments by use of a local approximation. By performing a greedy search over points near the required prediction location, a subset of observations can be used that approximately minimize the mean-squared prediction error among possible subsets. Gramacy *et al.* (2015) apply this method to emulating a radiative shock model. Gramacy and Lee (2012) propose a GP for use with a single-index model. Here, the simulator response surface is modeled as a 1D function in a single direction defined by a linear combination of the inputs. Gramacy and Lian (2012) then model the 1D function with a GP. Treating a complex simulator by a low dimensional function is justified in Constantine *et al.* (2014). Gramacy and Polson (2011) proposed a sequential Monte Carlo approach for fitting GP models in regression and classification in the case of on-line acquisition of data. Bastos and O'Hagan (2009) propose a number of numerical and graphical diagnostics for assessing the quality of an emulator.

Gaussian processes are also used for emulation of computer models for the purpose of objective optimization. Jones *et al.* (1998) discuss the use of GPs to approximate "black-box" functions for optimization by an expected improvement algorithm. They utilize the estimates of uncertainty provided by GPs for balancing global and local search. This work plays a foundational role in Bayesian optimization (Brochu *et al.*, 2010). Lee *et al.* (2011) propose GPs for use in optimization with unknown constraints, where both the approximation to the response surface of the model and the domain to which the optimization is constrained must be learned from data. Lindberg and Lee (2015) later use GPs for and propose an asymmetric entropy measure to assist in optimizing near or on the boundary of this sort of unknown constraint region.

Savitsky *et al.* (2011) and Shang and Chan (2013) proposed using Gaussian process regression with non-Gaussian responses. They show the value in using a GP as a flexible regression tool for modeling complex relationships but do not relate their regression approach with computer model emulation or calibration. A similar method for using GPs to model count data in spatial statistics is used in the log-Gaussian Cox process model (Diggle, 1985). Diggle *et al.* (2007) gives an overview of methods for Gaussian processes in spatial statistics including MCMC approaches and show examples using an `R` package `geoRglm` (Christensen and Ribeiro Jr, 2002). Their focus is on 2D Gaussian processes for spatial models and do not extend to higher dimensional problems nor to learning parameters related to the inputs to the Gaussian process as in computer model calibration.

In the machine learning community, Titsias and Lawrence (2010) propose a latent variable GP model for unsupervised learning of relationships between observed variables using variational Bayesian inference. Gal *et al.* (2015) proposed the use of this latent variable model with categorical observations.

2.2 Background on Computer Model Calibration

In some applications, model calibration is the stated goal. That is, using a limited number of field observations and evaluations from the computer model, the aim is to (i) estimate calibration parameters that govern the simulator response (Kennedy and O'Hagan, 2001);

and (ii) build a predictive model for reality that is "better" than using either the simulator or the experimental observations alone.

The standard approach for calibration and prediction for computer experiments was put forth in Kennedy and O'Hagan (2001). They proposed to view field data as noisy versions of a discrepancy adjusted computer model. An additional problem is that some of the inputs (calibration parameters, $\boldsymbol{\theta}$) to the computer model are neither adjustable, nor measurable, in the physical system and have to be estimated. Often computing costs dictate that the computational model must be replaced with a GP emulator of the response surface of the computer model. The Kennedy-O'Hagan (KOH) model can be summarized below:

$$\begin{aligned} Y_o &= \eta(\mathbf{x}, \boldsymbol{\theta}) + \delta(\mathbf{x}) + \epsilon, \\ Y_c &= \eta(\mathbf{x}, \mathbf{t}), \\ \eta(\mathbf{x}, \mathbf{t}) &\sim GP(0, \Sigma(\mathbf{x}, \mathbf{t}, \boldsymbol{\omega}_c)), \\ \delta(\mathbf{x}) &\sim GP(0, \Sigma(\mathbf{x}, \boldsymbol{\omega}_\delta)), \text{ and} \\ \epsilon &\sim N(0, \sigma^2), \end{aligned} \tag{2.5}$$

where Y_o and Y_c are the field and computer model observations respectively. The response surface of the simulator is $\eta(\mathbf{x}, \mathbf{t})$, $\delta(\mathbf{x})$ is the discrepancy between the response surface for the simulator and the true response surface, and ϵ is the random error. Additionally, we follow the convention of Kennedy and O'Hagan (2001), where $\mathbf{x}$ are observable or adjustable physical inputs, $\mathbf{t}$ are the inputs to the computer model to be calibrated from data, and $\boldsymbol{\theta}$ are the values for $\mathbf{t}$ which best model the true response. Lastly, $\boldsymbol{\omega}$ represents hyper-parameters, with a subscript indicating their associated distribution in the hierarchical model, with subscript c for the emulator of the computer model, δ for the discrepancy, and o for the field observations. The above model specification leads to a multivariate normal likelihood function of the form outlined in Higdon *et al.* (2004).

As illustrated in **?**, **?**, **?**, and **?**, there is an identifiability issue between the calibration parameters and the discrepancy function. By varying the discrepancy, the quality of fit to the data can be the same for a wide set of calibration parameters, meaning the posterior for the calibration parameters will be dependent on the choice of prior for the discrepancy. **?** proposed a form of discrepancy model attempting to correct this identifiability issue.

This model has been widely used and extended in many scientific applications. Higdon *et al.* (2004) model calibration to a charged particle accelerator and to a spot welding process model. Reese *et al.* (2004) propose a method to integrate expert judgment into the Kennedy and O'Hagan model for calibration. For handling multivariate observed outcomes, Bayarri *et al.* (2007b) use a wavelet decomposition and perform Gaussian process emulation on the wavelet coefficients. Higdon *et al.* (2008) similarly handle multivariate outcomes using principal components decomposition of the outcome variable. Liu *et al.* (2009) discuss the benefit of "modularization" in calibration with the Kennedy-O'Hagan framework. They discuss benefits, beyond simply reduced computational resources, that separating the estimation of emulator parameters from the estimation of calibration parameters and model discrepancy. Jacob *et al.* (2017) have recently discussed these benefits in a wider statistical model context. Qian *et al.* (2006) proposes a model for improving the quality of an emulator of a low-quality simulator by using limited observations of a high-quality simulator. Later, Qian and Wu (2008) and Goh *et al.* (2013) propose methods of combining the output of multiple experiments with a experimental observations for calibration beyond that discussed in Kennedy and O'Hagan (2001).

Heitmann *et al.* (2006) use the Kennedy-O'Hagan model to calibrate cosmological parameters using limited runs of an expensive simulator. Bayarri *et al.* (2009) use GP emulation to perform risk assessment with a volcanic flow model. They emulated the computer model for assessing the probability of flow depth exceeding dangerous levels, however their code did not require calibration at the stage that it was published. Henderson *et al.* (2009) and Henderson *et al.* (2010) do Bayesian calibration as part of a hierarchical model using a non-normal stochastic simulator with Binomial outcomes and experimental data normal observational errors. They model the probability of neuron survival in their simulator using a GP; however, they do not consider non-normal experimental data, model discrepancy and the flexibility that the generalized model allows in incorporating it, nor do they consider implementing the calibration framework more generally for other forms of non-normal outcomes. Oakley and Youngman (2017) propose calibration where the GP is used to estimate the expensive likelihood function, rather than emulating the computer model directly.

Some methods of Bayesian model calibration have been proposed with emulators besides GPs. Higdon *et al.* (2013) propose the ensemble Kalman filter as a computationally cheap alternative to doing full GP emulation for calibration. Pratola and Higdon (2016) propose emulating the response surface of the computer model with Bayesian additive regression trees to deal with high-dimensional input spaces and complex high-dimensional response surfaces. Sargsyan *et al.* (2015) propose an approach to calibration using polynomial chaos expansions (PCE) use approximate Bayesian computation (ABC) for fitting the model. Chakraborty *et al.* (2017) propose an alternative polynomial method with similar control over interpolation and assessment of uncertainty between observations and investigating this method may be a valuable path of future research. Joseph and Kang (2011) proposed an innovation on inverse distance weighting as an interpolation technique and showed strong performance of the model compared with the Gaussian process. They also proposed an approach for building confidence intervals for estimating uncertainty in emulator response for their regression-based inverse distance weighting model.

Chapter 3

Generalized Calibration of Computer Models with Gaussian Processes

The new methodology proposed in this chapter is motivated by effort to calibrate a radiation transport model used in neutron detection experiments by the Center for Exascale Radiation Transport (CERT) at Texas A&M University. Radiation transport models play a key role in simulating high-energy-density physical systems and are often a key computational bottleneck. The simulator that has been developed uses the parallel deterministic transport code (PDT) for predicting radiation and particle transport in complex geometries with high-fidelity and for scaling the calculations for large parallel computing architectures (Adams *et al.*, 2013). Calibration and quantification of prediction uncertainty with PDT is an important part of CERT.

The experimental data for this application comes in the form of neutron counts, where the discrete nature of the response is inconsistent with the support and structure of the usual Gaussian likelihood used for model calibration. In this chapter, we develop new methodology for Bayesian model calibration for count data. Our aim is to perform model calibration with uncertainty assessments that are faithful to the nature of the data. We utilize an elliptical slice sampling approach to fitting this model proposed by Murray *et al.* (2010) for efficient sampling of latent Gaussian processes for fitting the model. The proposed modeling framework is quite general and can also be used with other non-continuous observation error for model calibration such as categorical or survival data. Our proposed approach also allows the Kennedy-O'Hagan calibration framework to be embedded into different Bayesian hierarchical models which reflect the structure of the observation error and scientific understanding of the generating processes.

In this chapter, we discuss the proposed approach to model calibration for non-continuous outcomes, including model discrepancy of the form described in Kennedy and O'Hagan

(2001). The proposed hierarchical model is outlined in Section 3.3, followed by application of the model on simulated data in Section 3.4 and in calibration of a radiation transport model in Section 3.5, and concluding with discussion in Section 3.7.

3.1 Calibration with Gaussian Observations

The standard approach for calibration and prediction for computer experiments put forth in Kennedy and O'Hagan (2001) was described in Section 2.2. The KOH model is restated below, in a slightly different form as a precursor to the new methodology:

$$\begin{aligned} Y_o(\mathbf{x}) &= \eta(\mathbf{x}, \boldsymbol{\theta}) + \delta(\mathbf{x}) + \epsilon, \\ Y_c(\mathbf{x}, \mathbf{t}) &= \eta(\mathbf{x}, \mathbf{t}), \\ \eta(\mathbf{x}, \mathbf{t}) &\sim GP(0, \Sigma(\mathbf{x}, \mathbf{t}, \boldsymbol{\omega}_c)), \\ \delta(\mathbf{x}) &\sim GP(0, \Sigma(\mathbf{x}, \boldsymbol{\omega}_\delta)), \text{ and} \\ \epsilon &\sim N(0, \sigma^2 \mathbf{I}), \end{aligned} \tag{3.1}$$

with a slight change in notation for the field observations to note the dependence of the field and simulator observations on $\mathbf{x}$ and $\mathbf{t}$.

This model specification can be expressed to explicitly include the mean response surface $\mu(\mathbf{x}) = \eta(\mathbf{x}, \boldsymbol{\theta}) + \delta(\mathbf{x})$. The field data are noisy observations of this function at a location $\mathbf{x}$. The computer model data, with discrepancy and at the best value of the calibration parameters $\boldsymbol{\theta}$, are observations of this surface with some systematic discrepancy:

$$\begin{aligned} Y_o(\mathbf{x}) &= \mu(\mathbf{x}) + \epsilon \text{ and} \\ Y_c(\mathbf{x}, \boldsymbol{\theta}) &= \mu(\mathbf{x}) - \delta(\mathbf{x}). \end{aligned}$$

This form is identical to the typical Kennedy-O'Hagan approach, but the latent space is explicitly expressed instead of being marginalized implicitly. The minus sign for the discrepancy is only present to illustrate the equivalence with (2.5) but does not limit the model form for the discrepancy.

3.2 Generalized Regression Models

Generalized linear models are ubiquitous in statistics as an extension of linear regression models (McCullagh *et al.*, 1989). These models account for the support of the data (e.g., integer count data) and corresponding variability. In the context of uncertainty quantification in computer experiments, the likelihood is important for properly capturing the distribution of the outcome.

For generalized linear models, the mean of an outcome variable, Y, is specified as a function of a linear combination of the predictors via a link function related to the assumed response distribution (Agresti, 2013). More specifically, this can be viewed as modeling the latent mean function $\mu(\mathbf{x})$:

$$\begin{aligned} Y &\sim f(\mathbf{y}; \mu = \mu(\mathbf{x}), \boldsymbol{\omega}_o) \text{ and} \\ \mu(\mathbf{x}) &= g^{-1}(\mathbf{x}\beta), \end{aligned}$$

where $f(\cdot)$ is a density function parameterized to have a mean $\mu(\mathbf{x})$, parameter vector $\boldsymbol{\omega}_o$ to specify the remainder of the density function, $\mathbf{x}$ is an length p vector of covariates, n is the number of observations, β is a $p \times 1$ vector of linear coefficients, and $g^{-1}(\cdot)$ is the inverse link function connecting the linear combination of covariates to the latent mean. For a standard linear model, the density, $f(\cdot)$ is normal, the link function is the identity, and $\boldsymbol{\phi}$ is the residual variance. For count data, on the other hand, the distribution could be Poisson, with the log-link function used to connect the mean response and the covariates (see, for example, Agresti (2013) or Gelman *et al.* (2013)).

As discussed in Section 2.1, Savitsky *et al.* (2011) and Shang and Chan (2013) propose using GP regression models for non-Gaussian responses. Instead of modeling the mean function as a linear combination of the covariates, the mean is modeled as a random function $z(\mathbf{x})$, with a Gaussian process defining a distribution on the class of random functions. This leads to the following model specification (Savitsky *et al.*, 2011):

$$\begin{aligned} Y &\sim f(y; \mu(\mathbf{x})), \\ \mu(\mathbf{x}) &= g^{-1}(z(\mathbf{x})), \text{ and} \\ z(\mathbf{x}) &\sim GP(0, \Sigma), \end{aligned}$$

where $\mu(\mathbf{x})$ is a latent mean of the parametric distribution $f(\cdot)$ as a function of the covariate information.

3.3 Calibration with Non-Gaussian Observations

For the radiation transport application that motivated this work, the response variable comes in the form of neutron counts. For this data, as well as other non-continuous responses, we propose to replace the Gaussian observation likelihood with a distribution from a parametric family with density $f(x)$. We denote the expected response for the physical system as $\mu(\mathbf{x})$ and propose the following specification to combine field observations and simulation outputs for model calibration:

$$
\begin{aligned}
Y_o(\mathbf{x}) &\sim f(\mu = \mu(\mathbf{x}), \boldsymbol{\omega}_o), \\
\mu(\mathbf{x}) &= g^{-1}(\eta(\mathbf{x}, \boldsymbol{\theta}) + \delta(\mathbf{x})), \\
Y_c(\mathbf{x}, \mathbf{t}) &= g^{-1}(\eta(\mathbf{x}, \mathbf{t})), \\
\eta(\mathbf{x}, \mathbf{t}) &\sim GP(0, \Sigma(\mathbf{x}, \mathbf{t}, \boldsymbol{\omega}_c)), \text{ and} \\
\delta(\mathbf{x}) &\sim GP(0, \Sigma(\mathbf{x}; \boldsymbol{\omega}_\delta)).
\end{aligned}
\tag{3.2}
$$

The simulator at some value of its calibration parameters is treated as an observation of the underlying mean function of the process generating the field data with a systematic discrepancy.

For generalized linear models, the link function serves two main purposes: it connects the linear combination of regression coefficients to the mean of the observations, and it allows the support of the linear predictor to match that of the mean for the likelihood. Because the Gaussian process is capable of capturing complex, non-linear response surfaces, the second purpose, controlling the support, is most important here. The link function is likely to act on the simulator output as well, to place its values on the same support as the Gaussian process. For continuous Gaussian outcomes the link function is the identity, as in linear models, and $\boldsymbol{\omega}_o$ is the variance, σ^2. In the proposed setup, the log-link function is used to constrain the Poisson mean to the positive real numbers.

In the previous section, when the distribution for the mean and observations were both normal, the latent mean at each input setting need not be directly estimated as it can be analytically marginalized. For non-normal likelihood models, this is no longer possible. Explicitly marginalizing the latent space via integration would be an expensive calculation in this context, however, as we will discuss in Section 3.3.1, this can be addressed when fitting a Bayesian model by augmenting the parameter space to include the values of mean function, $\mu(\mathbf{x}_i)$, at each unique location $\mathbf{x}_i$ for the experimental data in the input space. This adds a potentially large number of correlated parameters required for inference, which can be difficult for computation.

In this chapter, the focus is on count data. We will take the Poisson regression approach, performing calibration on the intensity function using the model in Equation (3.2) and the log-link function. Potential over-dispersion or under-dispersion can be handled in the same manner as in classical generalized linear models(Agresti, 2013; Savitsky *et al.*, 2011).

3.3.1 Bayesian Model Inference

Fully accounting for uncertainty in estimation of model parameters and in prediction is important for computer simulation. In Bayesian inference, the uncertainty in the calibration parameters, $\boldsymbol{\theta}$, and in predictions for new observations, $Y_o(\mathbf{x}^*)$, can be summarized by their posterior distributions. This includes uncertainty due to marginalizing over the hyperparameters of the model, $\{\boldsymbol{\omega}_c, \boldsymbol{\omega}_\delta, \boldsymbol{\omega}_o\}$, and the unobserved expected values for the field data $\{\mu(\mathbf{x}_1), ..., \mu(\mathbf{x}_{n_u})\}$ where n_u is the number of unique input vectors. Field observations at the same location, $\mathbf{x}$, share a single latent mean parameter $\mu(\mathbf{x})$.

For example, the posterior distribution for the calibration parameters, upon observation of the simulator and field data is:

$$\begin{aligned}
&\pi(\boldsymbol{\theta} \mid Y_c, Y_o) \\
&\quad = \int \pi(\boldsymbol{\theta}, \boldsymbol{\omega}_c, \boldsymbol{\omega}_\delta, \boldsymbol{\omega}_o, \mu(\mathbf{x}) \mid Y_c, Y_o) d\boldsymbol{\omega}_c d\boldsymbol{\omega}_\delta d\boldsymbol{\omega}_o d\mu(\mathbf{x}) \\
&\quad \propto \int \pi(\boldsymbol{\theta}, \boldsymbol{\omega}_c, \boldsymbol{\omega}_\delta, \boldsymbol{\omega}_o) \pi(\mu(\mathbf{x}) \mid \boldsymbol{\theta}, \boldsymbol{\omega}_c, \boldsymbol{\omega}_\delta) \pi(Y_o \mid \mu(\mathbf{x}), \boldsymbol{\omega}_o) \pi(Y_c \mid \mu(\mathbf{x}), \boldsymbol{\omega}_c) d\boldsymbol{\omega}_c d\boldsymbol{\omega}_\delta d\boldsymbol{\omega}_o d\mu(\mathbf{x}),
\end{aligned}$$

where $\pi(\cdot)$ denote the prior distributions for the model parameters and where the probability distribution of the observed data may include parameters $\boldsymbol{\omega}_o$ or $\boldsymbol{\omega}_c$ for the experimental and simulator observations respectively. The integral is over the full support of the parameters. A common method to approximate this integral is to use Markov Chain Monte Carlo (MCMC) (Robert and Casella, 2005), which we will discuss later in this section.

The prior distributions for the model parameters must be specified. We follow Higdon *et al.* (2004) and use independent prior distributions for the calibration parameters ($\boldsymbol{\theta}$), the correlation hyper-parameters for the emulator and discrepancy ($\boldsymbol{\beta}, \boldsymbol{\gamma}$), the precision parameters for the emulator and the discrepancy (κ_c, κ_δ), and any parameters governing the likelihood $\boldsymbol{\omega}_o$. The Gaussian process prior specification for the latent mean function, conditional on the other model parameters, has already been discussed in Section 3.2.

The prior distributions that we use are summarized in (3.3). A common prior distribution for calibration parameters is a uniform distribution over the plausible range elicited from experts working with the simulator (Higdon *et al.*, 2004). The range should be made wide enough to ensure that the best value for the calibration parameter does not fall outside the range. By scaling the inputs to the unit hypercube, independent Uniform(0,1), or equivalently Beta(1,1), distributions are chosen for the prior specification.

We utilize independent, Exponential($\frac{1}{4}$) prior distributions for each β_ℓ. The motivation for this prior distribution comes from an equivalent parameterization of the squared-exponential covariance used in Linkletter *et al.* (2006) and Higdon *et al.* (2008):

$$r(\mathbf{x_i}, \mathbf{x_j}) = \prod_{\ell=1}^{d} \rho_\ell^{4(\mathbf{x}_{\mathbf{i}\ell} - \mathbf{x}_{\mathbf{j}\ell})^2}.$$

The parameter ρ_k can be interpreted as the correlation between two observations at a distance of half the input space in the kth dimension, assuming the input space has been scaled to the unit cube, $[0, 1]^p$, in p dimensions (Linkletter *et al.*, 2006). An uninformative, Uniform(0,1) prior distribution on ρ_k is equivalent to the Exponential($\frac{1}{4}$) prior distribution for β_k after transformation. We use the exponential parameterization to avoid sampling with a hard upper bound on the parameter.

For the discrepancy correlation parameter $\boldsymbol{\gamma}$, a prior distribution with more probability mass near 0 can be used to express *a priori* expectation that the discrepancy function is smoother than the emulator, while still only weakly imposing this prior preference. We use independent Exponential($\frac{1}{2}$) prior distributions for each γ_k, which is equivalent to independent Beta(2,1) prior distributions on ρ_δ.

It is not straightforward to assign prior distributions to the precision parameters. One option is to assign a weak prior to the precision for both the emulator, κ_c, and for the discrepancy, κ_δ. This ignores the prior expectation that the computer model should account for a larger fraction of the variability in the underlying true response function than the discrepancy - that the computer model is at least a reasonable approximation to the true system. Instead, we can quantify this prior information by choosing a prior distribution on the discrepancy parameter such that the prior expected variability of the discrepancy is a small percentage of the combined variability of the simulator and the discrepancy. Choosing this percentage to be 10% was reasonable in the examples:

$$\frac{1/\mathrm{E}[\kappa_\delta]}{1/\mathrm{E}[\kappa_\delta] + 1/\mathrm{E}[\kappa_c]} \approx 0.10.$$

The scale of the prior distribution for both precision parameters remains to be set. Modeling the standardizing the output of the computer model as a GP and setting the total expected prior variance to 1 is a common approach to choosing a prior on the precision (Higdon *et al.*, 2008; Linkletter *et al.*, 2006). Because of the link function, the Gaussian process describing the mean function may not be operating on the same scale as the observed data, so the outcome data cannot be standardized in the usual way. We instead standardize the link-transformed simulator output and latent mean values by the mean and standard deviation of the transformed simulator values. The transformation must be accounted for in the log posterior evaluations for sampling as well. Doing this allows us to set the prior distribution for the emulator precision parameter to correspond to a prior mean on the emulator variance of 0.9. After the standardization, we follow the approach of Higdon *et al.* (2008) and use an informative Gamma prior distribution for the precision parameters.

$$\begin{aligned}
\pi(\theta) &\sim \mathrm{Beta}(1,1),\\
\pi(\beta_k) &\sim \mathrm{Exponential}(1/4),\\
\pi(\gamma) &\sim \mathrm{Exponential}(1/2), \qquad (3.3)\\
\pi(\kappa_c) &\sim \mathrm{Gamma}(5,3.6), \text{ and}\\
\pi(\kappa_\delta) &\sim \mathrm{Gamma}(5,0.4).
\end{aligned}$$

Conditional on the hyperparameters, the GP prior distribution is specified for the transformed latent mean function, $\pi(g(\mu(\mathbf{x})) \mid \boldsymbol{\theta}, \boldsymbol{\omega}_c, \boldsymbol{\omega}_\delta)$. To translate this to a distribution on the latent mean, $\pi(\mu(\mathbf{x}) \mid \boldsymbol{\theta}, \boldsymbol{\omega}_c, \boldsymbol{\omega}_\delta)$, the Jacobian of the transformation is required. Because the link acts on each mean parameter separately, the Jacobian is the determinant of a diagonal matrix. For instance, with a log-transformation, the Jacobian is the product of the inverses of each latent mean value, $1/\mu(\mathbf{x})$.

Bayesian inference can be carried out using MCMC samples from the posterior distribution of the model parameters conditional on the observed field and simulator data. Unlike in previous work in computer model calibration, where the latent mean function can be marginalized trivially, the latent function values are treated as parameters. What this implies in practice is that not only do the model parameters have to be sampled, but so do the values of the mean function $\mu(x)$ for the field data. The latent mean values are highly correlated, which can make sampling difficult.

A straightforward MCMC approach would be to use single Metropolis-Hastings (MH) updates to each parameter individually. While this method works, it is inefficient for sampling the latent function values because of the aforementioned strong correlations between values. Neal (1998) and Savitsky *et al.* (2011) use a staged, random-walk MH sampler to alternate between updating the latent function values as a group with the other parameters fixed, then updating the remaining parameters. They do so multiple times per full iteration of the MCMC to improve the mixing. Elliptical slice sampling, as discussed in Murray *et al.* (2010), has been proposed as a more efficient alternative to sampling from Gaussian processes. We use elliptical slice sampling in this work based on empirical evidence of efficiency in test problems and the ability to avoid choosing the fixed Markov transition scale parameter in Neal (1998). The sampling approach used is summarized in Algorithm 1 and discussed in more detail in Appendix 1.

Algorithm 1 MCMC Sampling for Generalized Kennedy-O'Hagan Model

```
for $i = 1 \to n_{samples}$ do
    Single-site updates to model parameters
    for $j = 1 \to n_{\boldsymbol{\omega}_c}$ do
        Sample hyperparameter $\boldsymbol{\omega}_{c,j}$ with Metropolis-Hastings (MH)
    for $j = 1 \to n_{\boldsymbol{\omega}_\delta}$ do
        Sample hyperparameter $\boldsymbol{\omega}_{\delta,j}$ with MH
    for $j = 1 \to n_{\boldsymbol{\omega}_o}$ do
        Sample hyperparameter $\boldsymbol{\omega}_{o,j}$ with MH
    for $j = 1 \to n_{\boldsymbol{\theta}}$ do
        Sample calibration parameter $\boldsymbol{\theta}_j$ with MH
    Elliptical slice sampling for $g(\mu(\mathbf{x}^*))$
    Iterate
```

Often, one is interested in obtaining predictions and uncertainty estimates of the latent mean function at unobserved values of the inputs or of new observations. Using the output from the Markov chain above, obtain samples of the link-transformed predicted mean using the normal conditional distribution defined in Equations (2.3) and (2.4). The mth sample of the predicted, link-transformed mean is drawn according to:

$$\pi(g(\mu(\mathbf{x}^*)) \mid \mu(\mathbf{x})^{(m)}, \boldsymbol{\theta}^{(m)}, \boldsymbol{\omega}_c^{(m)}, \boldsymbol{\omega}_\delta^{(m)}) \sim N\bigg(\frac{1}{\kappa^{(m)}} r(\mathbf{x}, x^*)^T (\Sigma_{\mathbf{x}}^{(m)})^{-1} (g(\mu(\mathbf{x})^{(m)}) - 0), \\ \frac{1}{\kappa^{(m)}} - \frac{1}{(\kappa^{(m)})^2} r(\mathbf{x}, \mathbf{x}^*)^T (\Sigma_{\mathbf{x}}^{(m)})^{-1} r(\mathbf{x}, \mathbf{x}^*)\bigg). \tag{3.4}$$

Applying the inverse link function gives samples from the posterior distribution for the latent mean at the new location. Samples for future field observations can then be obtained by drawing from the distribution proportional to the likelihood:

$$Y_o(\mathbf{x}^*) \sim \pi(\mu = \mu(\mathbf{x}^*), \boldsymbol{\omega}_o) \propto f(\mu = \mu(\mathbf{x}^*), \boldsymbol{\omega}_o).$$

3.4 Implementation and Performance

Here we show the performance of the proposed methodology on a series of examples. First, we consider the case of a simulator that, conditional on the correct specification of the calibration parameters, has no discrepancy. Next, we explore a setting where there is a systematic discrepancy (i.e., the mean of the field data does not match the simulator for any values of the calibration parameters).

3.4.1 Simple Model Without Discrepancy

The first example has a simulator with two calibration parameters and one physical input variable. The computer model in this example is well-specified - the simulator is the mean function at the true values of the input parameters. Let:

$$Y_c(x, t_1, t_2) = t_2 \frac{3^{t_1}}{\Gamma(t_1)} (x + 2.2)^{t_1 - 1} e^{-3(x+2.2)}, \tag{3.5}$$

where t_1 and t_2 are the calibration parameters and x is an input. For the field data, the process mean is specified as in (3.5) with calibration parameters $\theta_1 = 1.4$ and $\theta_2 = 80$. This function was chosen to represent a rise-decay response on a scale such that the observed counts with this Poisson rate function are low, emphasizing the discrete, bounded support for the observations.

Two scenarios for the field data are considered - one with replicates and one without. The field data consist of $n_o = 8$ values at unique values for x in the interval $[-2, 3.8]$. Additionally, $n_o = 40$ values were generated with 5 at each of 8 unique x values to show the value of replicate observations. We will also evaluate the simulator at $n_c = 30$ locations selected using a maximin Latin hypercube design generated using the `lhs` package in `R` (Carnell, 2012). The ranges for t_1 and t_2 are $[1.0, 2.1]$ and $[40, 100]$ respectively; however, for the analysis all inputs are rescaled to $\in [0, 1]$ as discussed in Section 3.3.1. The ranges were selected to cover θ_1 and θ_2 while allowing for an amount of variability typical of a computer experiment application. We used the prior distributions from (3.3) for the calibration and hyper-parameters.

The joint posterior distribution of the statistical model and calibration parameters was sampled using the MCMC scheme in Section 2.3. We ran two Markov chains and obtained 45,000 samples in each from the posterior distribution as described in Section 3.3.1, after discarding 5,000 samples per chain used in the MCMC burn-in phase. The Gelman-Rubin diagnostic, implemented in the R package `coda` (Plummer *et al.*, 2006), was equal to 1.0 for all parameters. The maximum 95% upper confidence limit was 1.03 across all parameters in both cases, which indicates convergence of the sampler (Gelman and Rubin, 1992; Plummer *et al.*, 2006). We chose a conservatively large number of samples to assure convergence, however examination of the traceplots indicated the sampler was well-mixed and converged within one thousand samples.

Figure 3.1 shows a comparison between the true response from Equation (3.5) and the posterior predictive mean from the proposed model in Equation (3.2) for the unreplicated and replicated scenarios (Figures 3.1(a) and 3.1(b) respectively). The dashed lines in Figure 3.1 reveal close agreement between the estimated 95% prediction intervals from the model in red and the 95% prediction intervals based on Poisson errors with the true mean function in black. This agreement is much closer in Figure 3.1(b) with the replicate observations. The blue shaded region is a 95% credible interval for the mean. For both cases, the true mean is contained within the credible band, with the band for the larger sample size being tighter to the true mean.

To test the empirical coverage of the 95% credible interval when $n_o = 8$, 4,000 test samples were generated at 40 evenly spaced locations in X. The data generation and model fitting procedure above was repeated 200 times and the empirical coverage on the test set was 97.3%.

Figure 3.2 shows histograms of the draws from the posterior distribution for the calibration parameters θ_1 and θ_2 in the two cases outlined above. The posterior distribution for both θ_1 and θ_2 are peaked near the true value. In Figure 3.2(a) neither posterior distribution is strongly concentrated, as the limited number of observations and high measurement variance only discount extreme values. With the addition of the replicates, the posterior distribution is more concentrated near the true value in Figure 3.2(b).

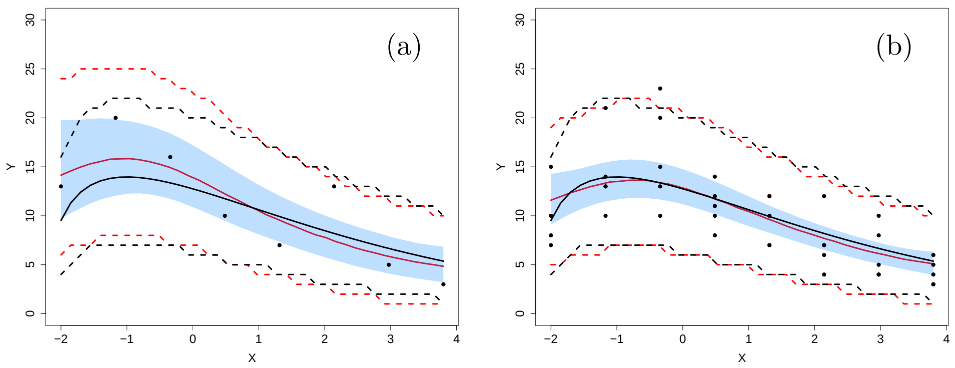
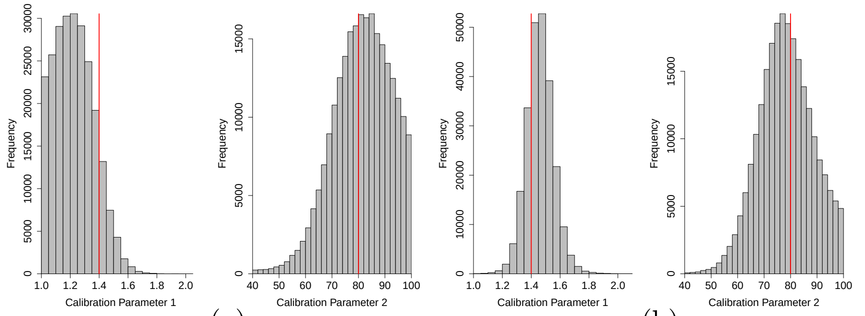


Figure 3.1: Comparison of the true mean function and predictive intervals (black) with the estimated mean and intervals (red) for $n_o = 8$ in (a) and $n_o = 40$ evenly spread across 8 unique input values in (b). A 95% pointwise credible interval for the mean is shown in blue and observations shown as black points for both cases.



(a) (b)

Figure 3.2: Histograms of the posterior samples of marginal distribution of the calibration parameters using the extension of the Kennedy-O'Hagan work outlined in Equation (3.2). The value for the calibration parameters used to generate the data are shown as a vertical red line.

3.4.2 Simple Model with Discrepancy

Consider the case where the simulator cannot adequately capture the true response function due to model form error. In this setting, a discrepancy function is required to account for portion of the signal in the field observations missing from the simulator. For this example, the previous mean function is modified to:

$$\mu(x) = 80\frac{3^{1.4}}{\Gamma(1.4)}(x+2.2)^{1.4-1}e^{-3(x+2.2)}(1+\frac{1}{2}\sin(\pi x)). \tag{3.6}$$

The computer model in this example in the same as specified in (3.5). The additional component in the system mean function is a periodic variation about the same curve with the amplitude varying with the level of the underlying signal. Figure 3.3(a) and 3.3(b) show this added periodic variation that is not included in the simulator.

Proceeding as before, we consider a setting with $n_o = 8$ field observations and $n_c = 30$ simulator runs chosen as a maximin Latin hypercube design. For data analysis, we use the prior distributions and sampling procedure outlined in Section 3.3.1. The same analysis is repeated with 5 replicates at each of the 8 locations of the field observations, leading to a total of $n_o = 40$ points.

Figure 3.3(a) shows one example of the full model fit with the discrepancy. As in Figure 3.1, the true mean function in black, and the predicted mean function with estimated 95% prediction intervals in red. The central 95% predictive intervals based on the generating mean function and Poisson error are plotted in black. The interval in black is not what we would expect to recover with a sample of data, but the target in the limiting case with infinite data. We can see that the model, with $n = 8$ is unable to reconstruct the underlying discrepancy between the simulator and the field observations. Despite the amplitude of the discrepancy being large, it is on the same order as the error variance due to the Poisson nature of the data. In this case, the amount and precision of data is unable to overcome the *a priori* assumption of a small discrepancy. In Figure 3.3(a), the predictive interval for the limiting case (black) noticeably extend beyond the estimated prediction intervals (red), leading to an empirical predictive coverage probability of 90.6%. This example illustrates a potential difficulty in estimating a discrepancy with limited, high-variance observations.

Obtaining five replicate observations at each of the previously sampled points provides a better estimate of the mean function at different locations in the parameter space. The addition of replicate observations allows the model to estimate the discrepancy as shown in Figure 3.3(b). We again test the empirical coverage of the 95% credible interval when $n_o = 8$, by generating 100 test samples at each of 40 evenly spaced locations in X, then repeating 200 times. The estimated prediction intervals in Figure 3.3(b) closely match the prediction intervals based on Poisson errors with the true mean function, leading to an empirical predictive coverage of the model with the replicates was 93.5%.

Because of the large model discrepancy, the posterior distribution for the calibration parameters is not tightly constrained near the values used to generate the data. The histogram of from the posterior distribution of the calibration parameters in Figure 3.3(c) shows the posterior distribution for θ_1 is concentrated near the lower end of the prior range and the posterior distribution for θ_2 still allows for a number of possible values, though it is peaked away from the correct value. In general, calibration in the presence of discrepancy is difficult to impossible (Loeppky *et al.*, 2006). Instead, in these cases, one is tuning to find good parameters for prediction.

3.4.3 Exploring the Simple Models

We continue the above examples, exploring the coverage of prediction intervals while varying the total number of observations, the number of replication observations at a particular value of x, and the scale of the observations. The reason for varying the scale of the model is

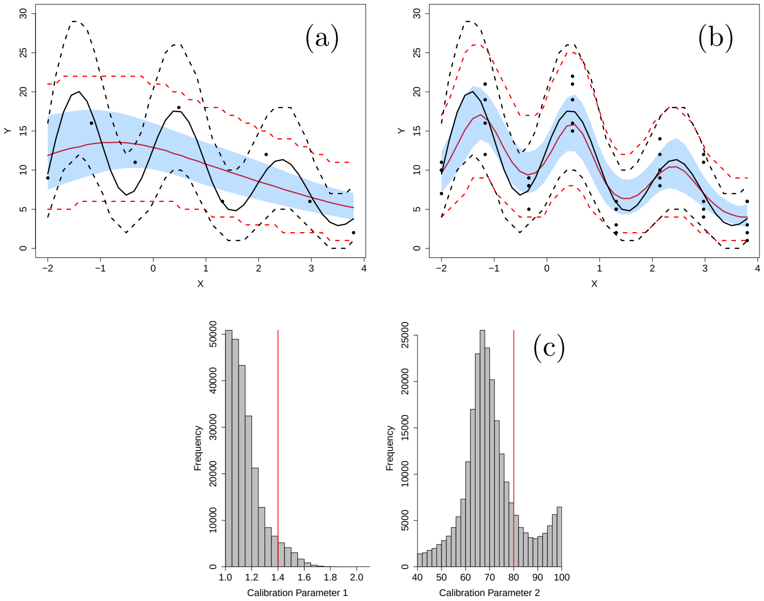


Figure 3.3: Comparison between the true response from Equation (3.6) in black and the posterior predictive mean from the proposed method in red. The dashed lines in red indicate the estimated 95% prediction intervals from the model. In (a) $n_o = 8$, while in (b) $n_o = 40$ spread across 8 unique values in x. We see that with little data, the discrepancy is unable to be reconstructed due to the high variance. The replicate values in (b) allow the discrepancy term in the calibration model to account for systematic bias in the simulator and better reconstruct the true function. In (c) the samples from the posterior distribution for the calibration parameters are not concentrated near the true values because the simulator cannot accurately describe the data.

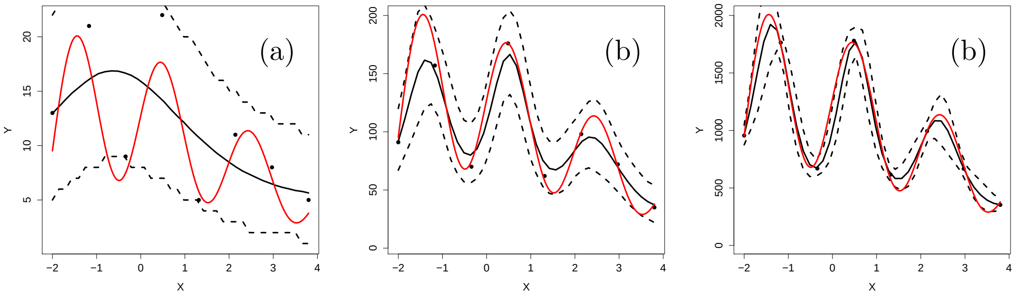


Figure 3.4: Comparing $n = 8$ with no replicates for output data on three orders of magnitude. The qualitative ability of the model to capture a periodic discrepancy for different magnitudes of outcome is clearly visible.

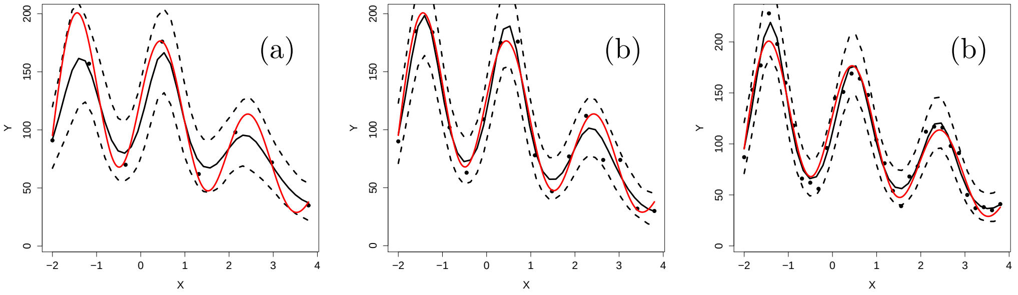


Figure 3.5: Comparing the ability to recover the model discrepancy for data with $n = 8$, $n = 16$, and $n = 32$.

that the Poisson variance is equal to its mean. As the mean of the observed data increases in magnitude, the standard deviation of the Poisson error shrinks in a relative sense.

In Figure 3.4, we compare the ability of the model to estimate the model discrepancy at different magnitudes of outcomes. For the lowest magnitude, the signal-to-noise ratio, $\frac{\mu(y)}{\sigma(y)}$, is approximately 3.5. The model is unable to capture even a large discrepancy given the level of noise. As the signal-to-noise increases to approximately 11, the model is able to roughly capture the shape of the discrepancy and by a signal to noise of 35, the discrepancy is easily captured.

In Figure 3.5, we compare the qualitative ability to recover the periodic model discrepancy for data with $n = 8$, $n = 16$, and $n = 32$. Figure 3.5a is the same example as Figure 3.4b.

By looking at cases with 4 replicates at each of two unique locations in the input space, we can see the value of using the simulator for prediction within the domain. In Figure 3.6, the shape of the estimated curves is entirely informed by the output of the simulator in this region. Because there are no field observations in the central portion of the input space, the model discrepancy is not identifiable. Coverage of the prediction intervals depends strongly

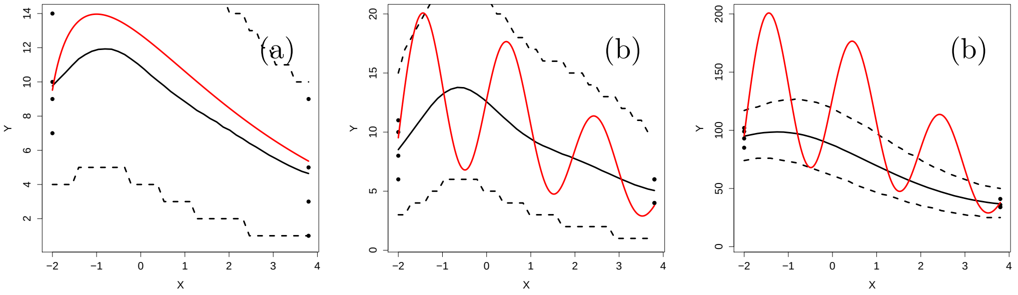


Figure 3.6: Comparing $n = 8$ with 4 replicates at each observations for the case with and without discrepancy. The simulator provides information about the shape of the function between the observed locations, however the model discrepancy is not identifiable without observations at more unique locations.

on the prior variance for the Gaussian process model of discrepancy. In Figure 3.6c, the prior variance is not large enough to account for the added periodic signal. The empirical prediction interval coverage for this case in repeat simulations was 59.4%.

3.4.4 High Dimensional Model Without Discrepancy

The same methodology can be applied to a higher dimensional input space. Instead of specifying a particular functional form for the true function to generate synthetic data, realizations from a 7-D Gaussian process are drawn, treating the first two dimensions as physical inputs and the remaining five as calibration parameters. This approach is taken to obtain empirical coverage probabilities of the method for a class of random functions. First, hyper-parameters are drawn from the prior distributions in Equation (3.3) and independent Latin hypercube designs for 20 field, 70 simulator, and 100 hold-out validation design points. To replicate the typical choice to run the simulator on a wider range of physical input values than is used for the field measurement, the simulator design points are generated on the range $[0, 1]$, while the field and validation design points are generated on the range $[0.2, 0.8]$. The simulator observations and the latent mean for the field and validation measurements are obtained as a draw from the resulting Gaussian process. To make the scale and magnitude of the data in this example comparable to the previous, the draw from the Gaussian process is multiplied by 0.4, added to 1.45, and exponentiated. Finally, field and validation measurements are obtained as Poisson draws with the generated latent mean.

This data generating procedure was repeated 50 times and used to obtain 100,000 samples from the posterior distribution of the calibration model using the sampling described in Section 3.3.1, as well as empirical coverage probabilities for the 50×100 validation points. The empirical coverage probability for the 95% predictive interval on the hold-out set was

97.54% coverage. This is consistent with the conservative coverage of the interval on the lower dimensional model as well.

3.5 Calibration of Radiation Transport Model

CERT is focused on high performance computing and simulation of radiation transport in high-energy-density conditions. Adequately modeling the flow of energy mediated by radiation in these conditions is crucial for capturing the overall complex behavior of physical systems. The CERT team develops and uses PDT for high-fidelity modeling of radiation and particle transport (Adams *et al.*, 2013). Calibration and uncertainty quantification of PDT are important for understanding the model's predictive capability and the ability to validate the model to new experimental conditions.

One focus of this work is to utilize PDT to quantitatively predict the measured neutron-detector count rate in experiments as a function of many variables, including those whose uncertainties are judged to play important roles. CERT experiments extensively use graphite, which is an important neutron "moderator" used in nuclear reactors and other applications involving neutron scattering. Impurities in the graphite can significantly change the neutron-transport properties of graphite, but the impurity concentration in CERT's graphite is unknown. The purpose of the work presented was to calibrate a model that accounts for the effects of impurities on the neutron transport in CERT's graphite. Experiments have been carried out measuring the neutron-detector count rate given neutrons emitted from an Americium-Beryllium source and transported through an experimental apparatus in which a graphite brick can be placed.

The experiment first measures the count rate in air only, which forms a baseline. Later CERT experiments will employ many graphite bricks, so interest is not only in characterizing the impurity in a single brick, but understanding the brick-to-brick variation as well. Each graphite brick, i, is exposed to the neutron source for time τ_i. The source is also exposed for time τ_{air} without a graphite brick to characterize the baseline emission rate.

PDT can inform the relationship between the concentration of impurities and attenuation of neutrons by modeling the count rate at the detector with and without the graphite between the source and the detector. The simulator uses two scalar model parameters: impurity concentration (in parts per million) and the neutron source axial position. Both are calibration parameters and there are no physically measured inputs in this application. The source axial position is a calibration parameter shared among graphite bricks, while each brick has its own impurity concentration. Simulator data from PDT was obtained from a size $n_c = 32$ maximin Latin hypercube design for the two inputs.

Understanding the distribution of impurities between bricks is of primary interest for making predictions of future experiments. In order to characterize the uncertainty in impurity for a single brick as well as the brick-to-brick variation, this calibration parameter is treated as a random effect. That is, since each brick can have different impurity concentrations, the interest lies in estimating parameters that govern the calibration distribution instead of a constant value. For the population distribution, we will use a truncated normal distribution on the $[0, 1]$ interval, with hyperparameters $\mu_{imp}, \sigma^2_{imp}$.

In Section 3.5.1, we will look at time-resolved data for each of the individual bricks. In Section 3.5.2, we will apply the proposed methodology to an expanded model including both the experimental observations through air and with graphite.

3.5.1 Poisson Calibration of Radiation Transport Model

Using only the observations on graphite bricks, the calibration is performed using the proposed methodology. The motivation for this example is two-fold: (1) to assess whether the observations for each brick are time-homogeneous and therefore can be treated as a single observation with a long exposure and (2) to illustrate how to use the methodology in cases where the data are inhomogeneous in time.

The simulator informs us of how the count rate through graphite varies as a function of the calibration parameters. In the experiments, the data are collected over time, and we are interested in assessing whether there is evidence for a time-varying discrepancy. We model the observations as having a constant, “best” value of the calibration parameters for each brick and allow the discrepancy to handle any time-dependence. This leads to the following specification:

$$\begin{aligned}
Y_{o,i} &\sim Poisson(\lambda(\boldsymbol{\theta}_i, \mathbf{x})),\\
log(\lambda(\boldsymbol{\theta}_i, \mathbf{x})) &= \eta(\boldsymbol{\theta}_i) + \delta(\mathbf{x}),\\
log(Y_c(\mathbf{t_i})) &= \eta(\mathbf{t_i}),\\
\eta(\mathbf{t}) &\sim GP(0, \Sigma(\mathbf{t}, \boldsymbol{\omega}_c)),\\
\theta_{i,imp} &\sim N_{[0,1]}(\mu_{imp}, \sigma^2_{imp}),\\
\mu_{imp} &\sim Unif(0, 1), \text{ and}\\
\sigma^2_{imp} &\sim Unif(0, 0.25).
\end{aligned} \tag{3.7}$$

In (3.7), $\mathbf{x}$ denotes time (we avoid using $\mathbf{t}$ to avoid confusion with the calibration inputs to the model). The Gaussian process parameter prior distributions are as defined in Section 3.3.1.

We have 6 bricks with 100 observations taken at 3.6 second intervals for each. The sampling approach discussed in Section 3.3.1. 5,000 samples from the posterior distribution for the statistical model parameters, calibration parameters, and model discrepancy are obtained via MCMC. Figure 3.7 shows the posterior predictions of the mean function through time, with 95% credible intervals for the mean. Because any deviation from horizonal in

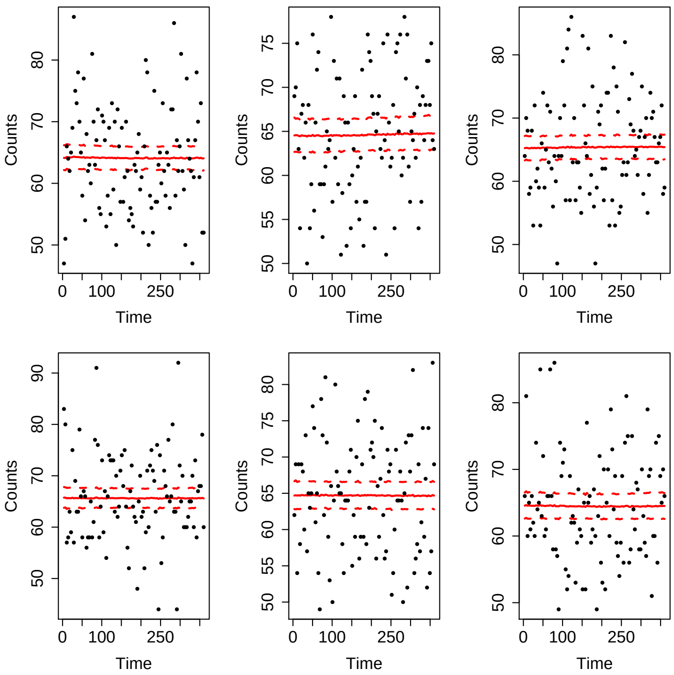


Figure 3.7: Observations of time-resolved detector count data for neutrons through each of 6 graphite bricks are shown as black points. The estimated time-varying mean function is shown in red with corresponding 95% credible intervals. There is no evidence of time-varying discrepancy in the observed count data for any of the experimental bricks.

the mean function is well within the credible interval for all bricks, there is no evidence of a time-varying discrepancy.

3.5.2 Expanded Calibration of Radiation Transport Model

We can more fully utilize the data for this system by embedding the simulator in a larger Bayesian model. Instead of simply modeling the count rate through graphite, the simulator can be used to model the attenuation of neutrons by the graphite as a function of the brick-specific model parameters, $\alpha(\mathbf{t})$. This demonstrates the flexibility in modeling that the approach allows. Because there is no evidence for time-varying signal, the observations on a brick can be treated as *iid* Poisson observations. Using the approach described in this chapter, the detector counts for each observation are modeled to have a Poisson likelihood, leading to the following model without accounting for model discrepancy:

$$
\begin{aligned}
Y_{air} &\sim Poisson(\lambda \tau_{air}), \\
Y_i &\sim Poisson(\alpha(\boldsymbol{\theta}_i)\lambda \tau_i), \\
logit(\alpha(\boldsymbol{\theta}_i)) &= \eta(\boldsymbol{\theta}_i), \\
logit(Y_c(\mathbf{t_i})) &= \eta(\mathbf{t_i}), \\
\eta(\mathbf{t}) &\sim GP(0, \Sigma(\mathbf{t}, \boldsymbol{\omega}_c)), \\
\theta_{i,imp} &\sim N_{[0,1]}(\mu_{imp}, \sigma^2_{imp}), \\
\mu_{imp} &\sim Unif(0,1), \text{ and} \\
\sigma^2_{imp} &\sim Unif(0, 0.25).
\end{aligned}
\tag{3.8}
$$

The observations through air simply depend on the baseline count rate of the source, λ, while the observations through graphite additionally depend on the attenuation - the fraction of incident neutrons blocked by the graphite. The dependence on time, $\mathbf{x}$ in Section 3.5.1, is dropped because each set of time-homogeneous Poisson observations can be combined to single observation with a known time exposure. The simulator provides information about the attenuation for a given brick geometry. As before, PDT must be emulated, in this case with a Gaussian process.

Further experiments provided $n_{bricks} = 10$ bricks, each with a total exposure of 3,600 s, while the baseline emission rate was obtained from an exposure of τ_{air} =10,800 s. Figure 3.8 shows the observed simulator data from PDT plotted versus each input parameter used in the computer experiment. The solid blue and red lines show the effect of each input, averaging the emulator over the other two inputs. The horizontal gray lines indicate the observed experimental data. The calibration process can be thought of as finding the values of the inputs that make the simulator compatible with the observed data (i.e., the gray lines). From this plot, it is clear that the observed data are only consistent with very low values of the impurity concentration and high values of the axial position. We also see that the brick-to-brick variation is small compared to the emulator response surface variation within the explored range of the model parameters.

We fit the model in (3.8), using prior distributions specified in Section 3.3.1 for the GP hyper-parameters, as a full Bayesian model using the sampling methods discussed in Section 3.3.1. We obtained 100,000 samples from the posterior distribution after discarding 5,000 samples used in the burn-in phase.

Figure 3.9 shows histograms of the MCMC samples from the posterior marginal distribution for the calibration parameters for each individual brick. The posterior distribution is consistent with our intuition from Figure 3.8. The posterior distribution of impurity concentration for each brick is tightly constrained against the lower boundary of the parameter range, which is consistent with the intuition gleaned from Figure 3.8. The posterior

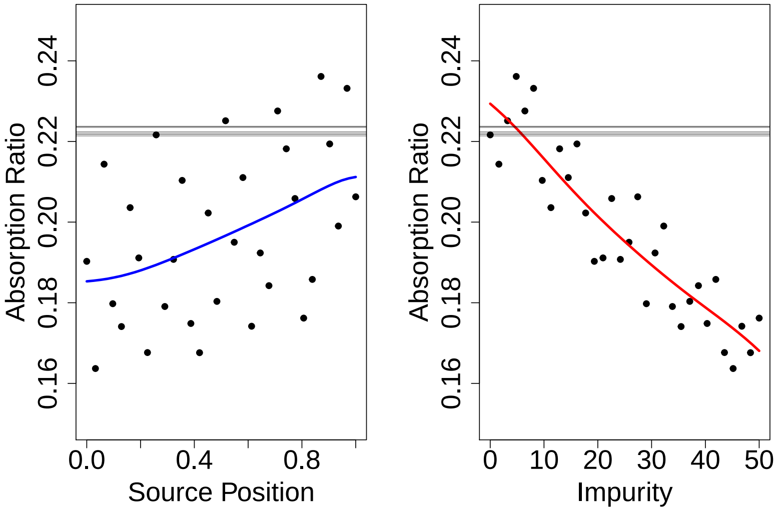


Figure 3.8: Projections of the simulator data to each calibration parameter dimension. The absorption ratio is clearly more sensitive to the impurity concentration than to the source position. Experimental measurements on ten bricks are indicated by the ten horizontal gray lines. They are not points because they do not have a measured value for any calibration parameter. We can see clearly that the experimental data is consistant with axial position values near the top of its input range and near the low end of the impurity concentration.

uncertainty for the impurity concentration in each brick is smaller than their brick-to-brick variability.

Because the impurity concentration is modeled hierarchically from a population of bricks, we can also look at the posterior distribution for the "between brick" distribution. It is important to account for this source of variability for making predictions of future bricks that may not have been previously used in an experiment. A histogram of samples from the posterior for the calibration distribution for impurity concentration is shown in Figure 3.10(a) and the histogram of samples from the posterior distribution of the shared calibration parameter is shown in Figure 3.10(b). While the observations of ten bricks clearly inform the marginal distribution of the impurity concentration, the data only weakly constrain the source position in the upper half of its input range.

3.6 Additive Gaussian Error with Constrained Model-Form Error

For some computer models, the observed model response may have constraints that are inconsistent with those of the observed noise in the experimental systems.

In the nuclear cross-section modeled discussed in Helgesson *et al.* (2017), the response of a perfect model is restricted to the positive real domain. The potentially imperfect simulator

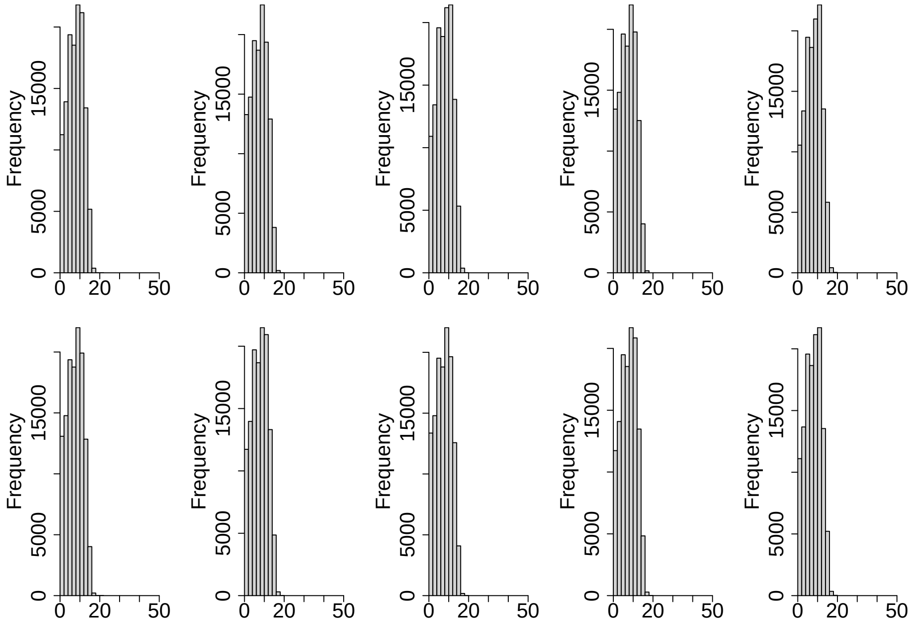


Figure 3.9: Posterior marginal histograms for the impurity concentration of each of the ten bricks. The impurity concentration is tightly contrained for each brick, even when fully accounting for uncertainties in the problem.

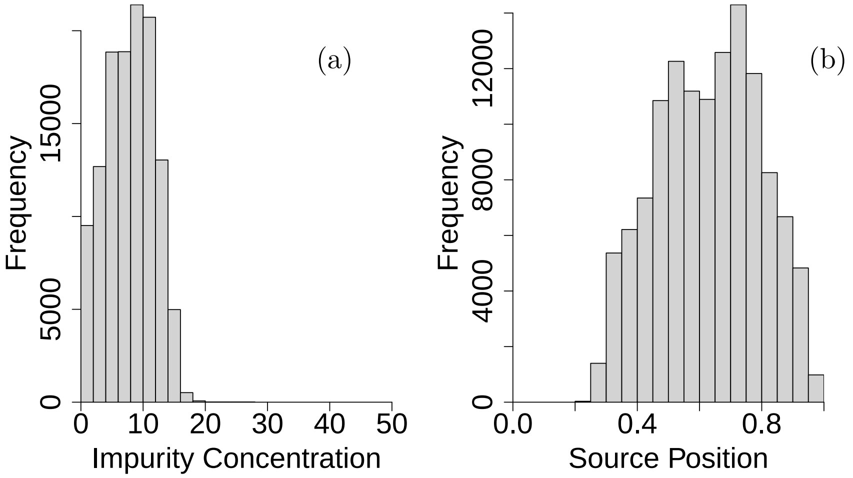


Figure 3.10: Posterior histogram for the population distribution of the impurity concentration, as well as the posterior distribution for the shared calibration parameters. The marginal posterior distribution for the impurity concentration is more tightly constrained by the ten observations than that of the axial position.

also satisfies this constraint, so modeling with any form of model discrepancy should as well. The experimental observations are not considered subject to the same constraint, as the observations are treated as having multivariate normal error.

Helgesson *et al.* (2017), used synthetic examples related to fission cross section models to assess the improved prediction with model-form uncertainty by using an additive Gaussian process model for the discrepancy. This model did not adhere to the aforementioned constraints, but provided a reasonable accounting of uncertainty for prediction. For the synthetic examples, the true experimental response function was:

$$Y_o(x) = (\sqrt{x} + 0.5x)e^{-x/2} \tag{3.9}$$

and the simulator with model-form error was:

$$Y_c(x, \mathbf{t}) = (\theta_1\sqrt{x} + \theta_2/\sqrt{(x)}) * e^{-x/\theta_3}.$$

The statistical model fit is similar to (2.5), but with a known, non-diagonal experimental covariance matrix Σ_ϵ and not requiring the emulation of the simulator:

$$\begin{aligned} Y_o(x) &= \eta(x, \boldsymbol{\theta}) + \delta(x) + \epsilon, \\ Y_c(x, \mathbf{t}) &= \eta(x, \mathbf{t}), \\ \delta(x) &\sim GP(0, \Sigma(x, \boldsymbol{\omega}_\delta)), \text{ and} \\ \epsilon &\sim N(0, \Sigma_\epsilon). \end{aligned} \tag{3.10}$$

This model formulation does not respect the *a priori* knowledge that the true mean function is known to have support only for positive values - $\mu(x) = \eta(x, \boldsymbol{\theta}) + \delta(x)$ has support on the real line for each value of x. A natural way to handle discrepancy for outcomes restricted to positive real numbers is to estimate the discrepancy of the log-transformed data. However, the measurement error is Gaussian on the untransformed scale of the experimental data, so fitting the full model with the log-transformation is not feasible.

The method proposed in Section 3.3 can be used to model the simulator with discrepancy on the log-transformed space, while still evaluating the field observation Gaussian likelihood on the original space. The model is the same as the above, but with:

$$\begin{aligned} Y_o(\mathbf{x}) &= \mu(\mathbf{x}, \boldsymbol{\theta}) + \epsilon \\ \log(Y_c(\mathbf{x}, \boldsymbol{\theta})) &= \log(\mu(\mathbf{x}, \boldsymbol{\theta})) - \delta(\mathbf{x}). \end{aligned} \tag{3.11}$$

This model can be fit using the sampling discussed in 3.3.

The statistical models in Equation 3.10 and Equation 3.11 are compared on two sets of six synthetic experimental observations. The first case is six observations from Equation 3.9

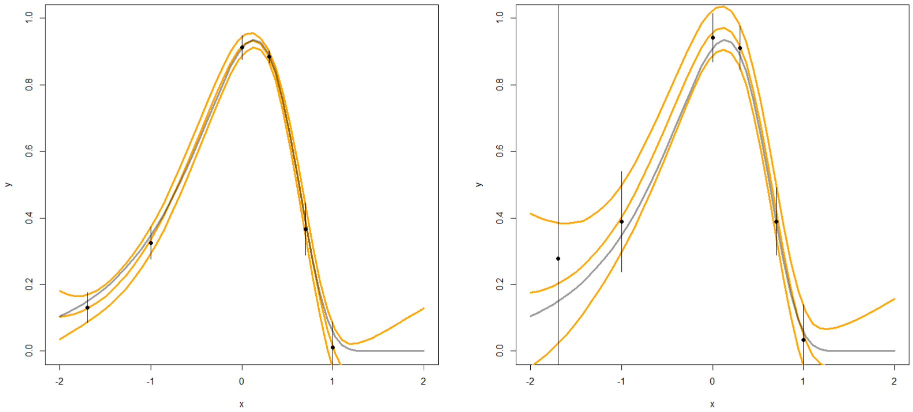


Figure 3.11: Estimation of true response curve for synthetic example relating to fission cross section models. The image on the left shows the case where the experimental measurements have uncorrelated Gaussian error but the model has a systematic bias. The image on the right includes an experimentally measured error covariance on the field data.

with a diagonal experimental covariance matrix; the second case is generated with correlated errors. The synthetic observations with the diagonal covariance matrix are shown in the left panel of Figure 3.11 and the correlated errors in the right panel.

Both models are estimated using Markov Chain Monte Carlo. Figure 3.11 shows the result of estimating the true experimental response function with 95% credible intervals in orange and the true response function in green. It is clear that the credible interval captures the true signal well, even with correlated experimental errors. However, the credible intervals extend far below zero, especially for large values of x, even though it is known the true function does not drop below zero.

Figure 3.12 show the estimated response function from fitting the model in Equation 3.11. The prediction bands still capture the true curve, while respecting the constraints on the model form. Additionally, the uncertainty near small values of the response surface is much smaller, reflecting the knowledge that the true response is likely to be on a similar order of magnitude to the estimated response.

3.7 Summary of Generalized Calibration

We have proposed a method for computer model calibration with Poisson outcomes using Gaussian process emulators in a similar vein to the Kennedy-O'Hagan framework. The performance of the methodology was shown on simulated examples and was used to perform calibration of a radiation transport model. The proposed approach requires sampling the latent mean function at each unique observed location in the input space. An elliptical

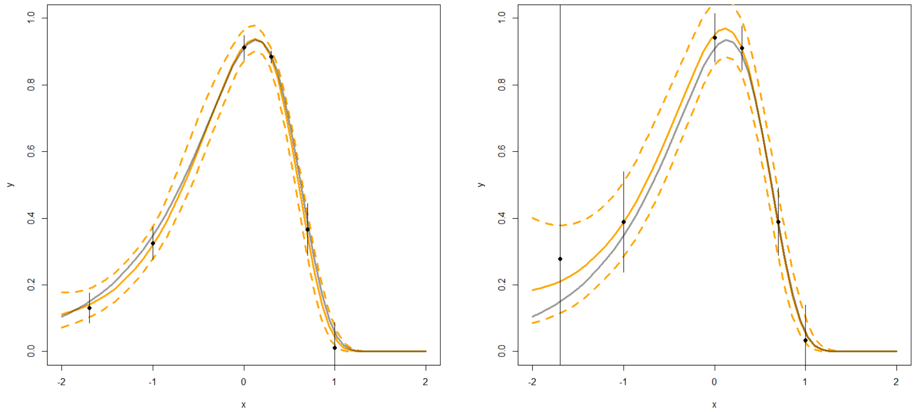


Figure 3.12: Estimation of true response curve for synthetic example relating to fission cross section models. The image on the left shows the case where the experimental measurements have uncorrelated Gaussian error but the model has a systematic bias. The image on the right includes an experimentally measured error covariance on the field data.

slice sampler was proposed in Section 3.3.1 and provides an efficient means for sampling the latent mean function parameters.

Because the sampler must be augmented with the mean function, the method can be slow as the number of observations becomes large. Additionally, storing these samples adds to the computational memory requirements of sampling from the posterior distribution. One way to address the scaling limitations of the memory is to only keep the current state of the Markov chain for the latent mean function parameters. An alternative may be to use inducing point methods described in Quiñonero-Candela and Rasmussen (2005).

For count data, when the number of counts is large, the normal approximation to the Poisson distribution is reasonable, and the Kennedy-O'Hagan framework can be used. In cases such as these, results using the proposed method from this work will likely agree with results using the classical Kennedy-O'Hagan framework. The benefit of the proposed framework in those cases is to incorporate likelihood structure other than the mean and variance into the model and to build a generative model which connects more closely to the actual process, with downside of the extra computational cost for sampling the latent mean values.

The proposed framework can easily be extended to other forms of observational data. Consider the case that observed data is binary instead of Poisson and the simulator provides information about the probability that the outcome $Y_o = 1$ as a function of some physical and calibration inputs. An instance would be labeling the quality assessment of an industrial part as {Pass,Fail}. The above framework can be applied with the following model:

$$\begin{aligned}
Y_{o,i} &\sim \text{Bernoulli}(p(\boldsymbol{\theta}_i, \mathbf{x}_i)) \\
\text{logit}(p(\boldsymbol{\theta}_i, \mathbf{x}_i)) &= \eta(\boldsymbol{\theta}_i, \mathbf{x}_i) + \delta(\mathbf{x}_i) \\
\text{logit}(Y_c(\mathbf{t_i}, \mathbf{x})) &= \eta(\mathbf{t_i}, \mathbf{x}) \\
\eta(\mathbf{t}, \mathbf{x}) &\sim GP(0, \Sigma(\mathbf{t}, \mathbf{x}, \boldsymbol{\omega}_c)).
\end{aligned} \tag{3.12}$$

where the transformation $\text{logit}(p) = \log(\frac{p}{1-p})$ is used to keep the transformed emulator bounded between (0,1). If the outcome data has $c > 1$ categories, the approach can be used, but requires $c - 1$ latent mean functions directly analogous to those in multinomial or probit regression in generalized linear models.

Lastly, for computational convenience, the emulation of the computer model and the calibration can be modularized as described in Bayarri *et al.* (2007a). In this empirical Bayes approach, point estimates of the hyperparameters of the emulator, commonly the posterior mean or maximum value, are passed forward to the calibration stage. This approach usually ignores propagation of uncertainty in the hyperparameters into the final prediction estimates and posterior distributions of calibration parameters. Their work has indicated that this has minimal impact (Bayarri *et al.*, 2007a).

Chapter 4

Validating Predictive Distributions from Complex Simulators

4.1 Background for Model Validation

Large-scale computational models are critical for developing our understanding of complex physical systems and for use in decision-making related to policy and risk assessment. Developing methods for building trust in the quality of the model, and understanding the limitations of conclusions drawn from the model, play an important role in their use (Post and Votta, 2005; Oberkampf and Roy, 2010). The processes of verification and validation for scientific models provide the basis of building evidence for this trust in a simulator.

Verification is the task of ensuring the simulator is adequately solving the specified mathematics of the physical theory for the system. The simulator is designed to solve these equations, however numerical approximations are often necessary, (e.g. discretization of the computational domain) (Oberkampf and Roy, 2010). Additionally, simulators may contain bugs or improperly applied algorithms (Post and Votta, 2005). These sources of error can contribute to deviations between the simulator output and that of the mathematical model. A verified model is one in which these issues are considered negligible (Oberkampf and Roy, 2010). Throughout this chapter, the simulators are assumed to be verified.

For use in decision making and risk assessment, it is not enough for a model to solve the underlying mathematical model properly, it must also adequately reflect the behavior of the system of interest. Validation is the comparison of the computer model output to observations from the system and the assessment of the ability of the model to predict aspects of real-world systems (Oberkampf and Roy, 2010).

Validation is performed in the presence of various sorts of uncertainty. In this field, it is conventional to classify the sources of uncertainty into two broad categories: (i) epistemic uncertainty due to insufficient knowledge about aspects of the system (e.g. the value of parameters that govern the model), and (ii) aleatoric uncertainty due to random variability inherent in the physical systems. We will describe these types of uncertainty in more detail in Section 4.1.2. Several approaches to statistical model validation have been proposed and several of these will be briefly discussed shortly (Hills and Trucano, 1999, 2002; Mahadevan and Rebba, 2005; Rebba *et al.*, 2006; Bayarri *et al.*, 2007a; Ferson *et al.*, 2008; Roy and Oberkampf, 2011; Sankararaman and Mahadevan, 2011; Wang *et al.*, 2012).

In this chapter, we propose a new method for validating a computer model by comparing the cumulative distribution of the outcome generated by uncertainty propagation through a simulator to observed outcomes. The proposed method uses a form of Bayesian posterior predictive model checking to investigate the ability of the simulator to generate predicted distribution functions for the output of interest similar to the observed distribution in field data, while also accounting for epistemic uncertainty about model parameters. The procedure gives both qualitative and quantitative comparisons of the distribution of observations generated by the simulator to the distribution of the field data. We additionally propose a hypothesis test as a decision rule for evidence against model validity that has a known asymptotic bounds on the probability of falsely declaring a model invalid.

The chapter is organized as follows: First, epistemic and aleatory uncertainty are introduced in Section 4.1.1. In Section 4.1.2, a brief re-review of computer model calibration; this time in the context of model validation that requires the introduction of new notation for this setting. Section 4.1.3 introduces the idea of uncertainty propagation and 4.1.4 establishes the background and key concepts for model validation. Section 4.1.5 illustrates one common method for validation and highlights some issues the proposed methodology attempts to overcome. In Section 4.2 we propose a new procedure for Bayesian model validation accounting for and separating aleatoric and epistemic uncertainties. In Section 4.3 performance of the proposed validation methodology is explored using a set of synthetic numerical examples. The approach is applied to the radiation transport application that motivated this work in Section 4.4. Final thoughts are presented in Section 4.5.

4.1.1 Epistemic and Aleatory Uncertainty

Different forms of uncertainty are commonly present during the process of calibration and validation. It is common to classify uncertainty from inputs to the experimental or computer experiments into two categories: aleatory uncertainties and epistemic uncertainties (Oberkampf and Roy, 2010). Aleatory uncertainties arise from aspects of the system that have uncontrolled variation from observation to observation, where the variability can be characterized by a sampling distribution. For example, additive Gaussian noise typically assumed in many models is one form of aleatory uncertainty. Another example is that a graphite brick in a neutron transport experiment may have a specified level of impurity, but because of uncontrolled variation in the process of making and storing the bricks, the realized impurity level in a population of bricks may vary according to a probability distribution centered near the specified level. The uncertainty in the QoI in a neutron transport experiment attributable to the variation in brick impurity cannot be reduced for prediction or reliability assessment of future experiments. Aleatory uncertainties are sometimes referred to as “irreducible” error for this reason (Oberkampf and Roy, 2010).

Epistemic uncertainties arise from a lack of knowledge about components of the system. One example is uncertainty due to the lack of knowledge of the best value for calibration parameters. Unlike aleatory uncertainties, epistemic uncertainties do not have any inherent variability. There are several approaches to incorporating epistemic uncertainty including interval-based assessment of uncertainty (Ferson *et al.*, 2003) or with the probabilistic models using Bayesian methods (Gelman *et al.*, 2013). Whether uncertainty in a variable is epistemic or aleatoric can depend on the context in a model. While the aleatory uncertainty in brick impurity cannot be reduced when sampling a brick for each experiment from the population of bricks, if one specific brick is used with a fixed amount of impurity, knowledge about the impurity level for that brick can be improved through observation and as such is an epistemic uncertainty.

The Bayesian framework is commonly use for calibration, where aleatory and epistemic uncertainties are not separated. Instead, as described in the previous chapter, all uncertainties in the statistical model are quantified using probability distributions. The prior distribution for a parameter represents a quantification of what values are thought to be plausible before data are collected, as specified by the support of the prior distribution, and how prior information supports some values relative to others in the assumed probabilistic model.

There is a parallel between epistemic/aleatory uncertainty and fixed/random effects in statistical models. The variability between units in a random effect model can be characterized by a probability distribution, but cannot be eliminated when sampling observations from new units. This variability is aleatoric. For a fixed effect, the uncertainty is only in the lack of knowledge of the magnitude of the effect. It has no intrinsic variability and so this uncertainty is epistemic.

4.1.2 Calibration in the Context of Model Validation

As discussed in the previous chapter, the calibration of computer models is concerned with using physical observations to constrain the uncertainty in model parameters and give evidence for which values of these parameters are consistent with the observed data. Calibration improves the ability to find valid parameter combinations, while also restricting the space to only parameter values consistent with calibration observations in a way that gives desirable statistical properties. The process of calibrating a model and using validation to ensure that the model gives outputs that are consistent with observations is the core to building trust with a model. At a coarse level, the calibration-validation path is as follows:

1. Calibrate the simulator to constrain the plausible parameter values using observed data.

2. Test the calibrated model against the observed data to ensure baseline validity, potentially by cross-validation.

3. Conduct an independent experiment on the system for the purposes of validation.

4. Run the computer model on the design of the validation experiment.

5. Assess the validity of the model by comparing simulator outputs to the observations from the validation experiment.

6. Improve model if there is evidence the model is invalid in Step 5 or design further validation tests to further probe the validity of the simulator.

In this chapter we shall depart from the notation of the previous chapter somewhat to differentiate between aspects of the statistical model for calibration and for validation. The model has physical inputs, $\mathbf{x}_{c,s}$, and calibration inputs required to evaluate the model, $\mathbf{t}_{c,s}$. Evaluation of the simulator at those inputs produces as an output, $y_{c,s}$. The first subscript denotes that the variable is for calibration, c, while the second subscript indicates the variable is used with the simulator, s. The proposed methodology focuses on deterministic simulators that output scalar QoIs, but can be adapted to stochastic or multivariate outputs (or both). Furthermore, the physical inputs, $\mathbf{x}_{c,s}$, may be considered known to high precision, or could be stochastic and characterized by a probability distribution. To distinguish between these two settings, we will refer to inputs that are known with a lowercase $\mathbf{x}_{c,s}$ and inputs that have aleatory uncertainty with an uppercase $\mathbf{X}_{c,s}$.

An important part of the process of computer model calibration is finding the value or potentially set of values for the calibration parameters $\boldsymbol{\theta}_{c,s}$ that make model predictions most consistent with observed data, $\mathbf{x}_{c,f}, y_{c,f}$, in order to reduce epistemic uncertainty. Here the second subscript, f, indicates the variable is for field observations.

For a computer model to be valid, there should be no evidence of model discrepancy at some value of the calibration parameter. In the model calibration context, this would mean the KOH model would have no discrepancy term. That is:

$$\begin{aligned} Y_{c,s} &= \eta_c(\mathbf{x}_{c,s}, \mathbf{X}_{c,s}, \mathbf{t}), \\ Y_{c,f} &= \eta_c(\mathbf{x}_{c,f}, \mathbf{X}^*_{c,f}, \boldsymbol{\theta}) + \epsilon \\ \theta &\sim \pi(\theta), \text{ and} \\ \mathbf{X}^*_{c,f} &\sim \pi(\mathbf{X}), \end{aligned} \tag{4.1}$$

where ϵ represents measurement error and is often modeled as a mean zero Gaussian random variable, though this error need not be Gaussian nor strictly additive error as discussed in the previous chapter. $\mathbf{X}_{c,s}^{*}$ is the unobserved value of the random physical input in the field data and $\eta_c(\mathbf{x}_{c,s}, \mathbf{X}_{c,s}, \mathbf{t})$ denotes the computer model configured to simulate the calibration experiment. We will continue to focus here on Gaussian process emulators, therefore $\eta_c(\mathbf{x}_{c,s}, \mathbf{X}_{c,s}, \mathbf{t})$ in Equation 4.1 is modeled as:

$$\eta_c(\mathbf{x}_{c,s}, \mathbf{X}_{c,s}, \mathbf{t}) \sim GP(\mathbf{0}, \mathbf{\Sigma}).$$

The hyperparameters of the Gaussian process have been suppressed for brevity. When attempting to validate the model, we take the view that a model is not considered valid if it requires a discrepancy model. As proposed by Bayarri *et al.* (2007a) and Wang *et al.* (2012), a non-zero discrepancy is an indication that the model is not valid.

4.1.3 Uncertainty Propagation

In some investigations, interest lies understanding how uncertainties in experiments propagate to prediction uncertainty (Smith, 2013). One may be interested in propagating both epistemic uncertainty in parameter values and aleatoric uncertainty from measurement error or stochastic inputs through a model for building prediction intervals incorporating all sources of uncertainty. Alternatively, one may be interested in predicting the probability distribution induced in the QoI by propagating sources of aleatory uncertainty through the model and assessing how that distribution changes due to epistemic uncertainty. This is done for reliability assessment (Karanki *et al.*, 2009), to explore properties of the distribution of the QoI Ghosh and Mujumdar (2009), and for validation of a simulator Gel *et al.* (2013); Lee *et al.* (2016); Roy and Balch (2012).

Let $F_{c,f}(y)$ be the cumulative distribution function (CDF) for the field observations, $Y_{c,f}$, defined as $F_{c,f}(y) = P(Y_{c,f} \leq y)$. Here $Y_{c,f}$ is the random variable denoting the field observations. Given a sample of size $n_{c,f}$ of field observations, the empirical cumulative distribution function (ECDF), $\hat{F}_{c,f}(y)$, is the fraction of observed values less than or equal to y. The CDF for the data can be predicted by propagating the known sources of variability across the simulator.

The statistical properties of the inputs and other sources of variability are often well characterized by a probability distribution on the inputs $\pi(\mathbf{X})$ and another distribution for measurement error. In light of the known sources of uncertainty, a valid simulator can be used to attempt to better understand properties of the distribution of $Y_{c,f}$. For a given simulator and known sources of variability (i.e. in input distributions and noise), denote the predicted cumulative distribution function of the outcomes be $\tilde{F}_{c,f}(y \mid \mathbf{t})$. Improved information about the value of the calibration parameters allows for better understanding of the predicted distribution of outcomes.

One simple method for uncertainty propagation is to use Monte Carlo sampling from the distributions of the sources of variability and propagating these through the computational model. This is, of course, infeasible if the simulator is computationally expensive. As with model calibration, the simulator can be emulated for purposes of uncertainty propagation.

A common class of surrogate models for uncertainty propagation are generalized polynomial chaos (gPC) methods (Xiu and Karniadakis, 2002). These methods give efficient estimation of the first two moments of the outcome variable. Monte Carlo can be performed on the gPC surrogate for estimation of other aspects of the distribution. We instead use the Gaussian process surrogate, because epistemic uncertainty in the response surface at un-evaluated locations results in uncertainty in the potential distribution of the outcome.

4.1.4 Validation of Computational Models

The goal of validating a computer model is to assess the ability of the model to reflect the behavior of the physical data, taking into account sources of variation and noise in the physical system. The process should account for all sources of uncertainty in the computer model and field observations.

Common practice is to apply a validation test to a set of data independent of the set used to calibrate the model. However, a first check to ensure that the simulator is, at minimum, able to replicate the calibration data can be valuable. Identifying that the model is unable to replicate the calibration data can save the resources to simulate and carry out the collection of validation data before the model deficiencies are addressed.

Depending on the setting, interest could lie in validating predictions of specific cases or validation predictive distributions. If all physical inputs to the model are measurable and can be considered known, then validation of specific model predictions, $\hat{y}(\mathbf{x}, \boldsymbol{\theta})$ involves looking for evidence that the mean predictive error, $E\left[\hat{y}(\mathbf{x}, \boldsymbol{\theta}) - y_{v,f}\right]$, is not zero. For systems in which inputs are variable and the predictive distribution is of interest, the goal is to probe for evidence against the hypothesis that the observed field data is distributed according to the distribution function generated by the simulator.

Zhang and Mahadevan (2003) apply Bayesian hypothesis testing to the problem of computer model reliability and validation. They use conventional Bayesian hypothesis testing with Bayes factors and proposed a method for marginalizing over uncertainties in computer model parameters. Mahadevan and Rebba (2005) consider validation using Bayesian hypothesis testing for the problem of calibration of computer models using Bayes networks. They propose building Bayes networks for representing the relationships in the joint distribution between parameters for a computer model or hierarchy of computer models and sharing information about parameters from different designed experiments. Rebba and Mahadevan (2008) proposed a Bayesian method for validation of specific predictions by assessing the posterior probability that $|y_{true} - \hat{y}| \leq \Delta_{crit}$ for some model prediction $\hat{y}$ and acceptable error threshold Δ_{crit}. This approach has been expanded by Jiang and Mahadevan (2009), Sankararaman and Mahadevan (2011) and Mullins *et al.* (2016).

Wang *et al.* (2012) proposed an alternative approach to Bayesian model validation, fitting a Bayesian model with a flexible discrepancy term $\delta(\mathbf{x})$ and validating the model based on the posterior probability that the discrepancy is "small". Specifically, they declared a model as valid if the absolute value of the upper and lower credible values for the discrepancy, $|U(\delta(x))|, |L(\delta(x))|$ were both less than some critical value, Δ_{crit}. Yuan and Ng (2015) use the Wang *et al.* (2012) approach to model validation, but in an integrated loop with calibration and sequential design. They discuss model validation and using maximum entropy or integrated prediction MSE-based sequential designs for obtaining new experiment or simulator results in the process of updating and validating a computer model. Bayarri *et al.* (2007a) similarly discuss validation in the context of estimation of model discrepancy, or bias, and assessing validity of the model by estimating the probability of the model with bias to predict within a given tolerance with high probability.

A frequentist approach to validation of specific model predictions using confidence intervals for the model discrepancy was proposed by Hills and Trucano (2002) and advocated in Oberkampf and Roy (2010). They propose to build a confidence interval for $Y_{true} - \hat{Y}$ that has a specified asymptotic coverage probability. A model is considered statistically valid if the confidence interval contains 0.

For systems in which inputs are variable and the predictive distribution is of interest, the goal is to probe for evidence against $\tilde{F}(y_{v,f}) \overset{d}{=} F(y_{v,f})$ where $\tilde{F}(y_{v,f})$ is the predicted CDF from the simulator and $F(y_{v,f})$ is the distribution function for the field data. For a valid simulator, one would expect those distributions to be equal. Ferson *et al.* (2008) proposed an area metric for measuring the distance between a predicted and empirically estimated distribution function as a means for validating with distributions. This area metric uses the area between the predicted distribution function and the observed ECDF as a measure of discrepancy. Oberkampf and Roy (2010) briefly discuss the use of other goodness-of-fit tests such as the Kolmogorov-Smirnov statistic for validating predictive distributions, but focus more on the validation test in the next section.

4.1.5 The P-Box for Model Validation

Oberkampf and Ferson (2007) proposed a method for validating a simulator and making reliability assessments on quantities of interest using an approach they called the "p-box method". The p-box uses the cumulative distribution function of the QoI and accommodates epistemic uncertainty in knowledge of parameters using an interval that bounds their plausible values.

Suppose we are interested in observations of a QoI, y, whose distribution is a function of parameters $\mathbf{t}$ that have fixed but unknown values, $\boldsymbol{\theta}$. For a given value of input parameters $\mathbf{t}$, there is a unique CDF, $F(y; \mathbf{t})$. The p-box is designed to be used if the value of $\boldsymbol{\theta}$ is only known to be in some interval, $[\mathbf{t}_l, \mathbf{t}_u]$, but no other information about the uncertainty in $\boldsymbol{\theta}$ can be claimed. For each value of $\mathbf{t}$ within the interval, a CDF can be constructed. From the set of all CDFs generated by $\mathbf{t}$ in the interval, the p-box is defined as the region interior to the inside the convex hull of the CDFs.

For an example, suppose the hypothesized distribution is Gaussian with the mean only known to be located in the range $\mu \in [-1.5, 1.5]$ and a known standard deviation of 0.25. That is,

$$Y = \mu + \epsilon, \text{ where } \epsilon \sim N(0, 0.25^2).$$

For each μ in the interval, there is a corresponding CDF. Figure 4.1 shows an example of a p-box, shaded in grey, formed by the interior of the convex hull of the Gaussian CDFs.

The convex hull is known in closed form for only a small set of parametric distributions. When working with the output of a complex simulator, the p-box validation testing procedure first builds the convex hull by evaluating $F(y, \mathbf{t})$ for some collection of values of $t = \{t^{(1)}, \ldots, t^{(M)}\}$, sampled on the interval (for example by simple random sampling or using a Latin hypercube design) (Roy and Oberkampf, 2011). The bounds of the p-box are then formed by finding the minimum and maximum value of the set of CDFs for each y.

For model validation, a metric of discrepancy between the empirical distribution function of the data and the p-box is used to assess whether the model is an adequate replication of the experimental measurements. The idea is that when the ECDF for the observations is far from the p-box, there is evidence that the model is not valid. Roy and Oberkampf (2011) recommend using the area metric (Ferson *et al.*, 2008), defined as the area between the ECDF and the boundaries of the p-box:

$$A = \int_{-\infty}^{\infty} \max\left(\inf \mathcal{F}(y) - \hat{F}(y), \hat{F}(y) - \sup \mathcal{F}(y), 0\right) dy, \tag{4.2}$$

where $\inf \mathcal{F}(y) - \hat{F}(y)$ is the distance between the lower boundary of the p-box at y and the ECDF, $\hat{F}(y) - \sup \mathcal{F}(y)$ is the distance between the ECDF and upper boundary of the p-box at y. The inclusion of zero ensures that there is no contribution to the area metric when the ECDF is in the interior of the p-box at y.

The second panel of Figure 4.1 shows an example of the area metric discrepancy between the p-box and an ECDF in red. For the empirical distribution in the illustration in Figure 4.1, a sample of $n = 25$ was drawn from a mixture of two Gaussians, one with mean of 0.5 and one with mean of 2.5, each with standard deviation of 0.25 and both mixture components having equal probability. The area of this region is the value of the area metric, in this case the value is 0.443.

The p-box validation framework has two main drawbacks. The first relates to deciding when the area test statistic is large enough to justify rejection of the model. The area metric has the same physical units as the QoI, which is part of the motivation for choosing

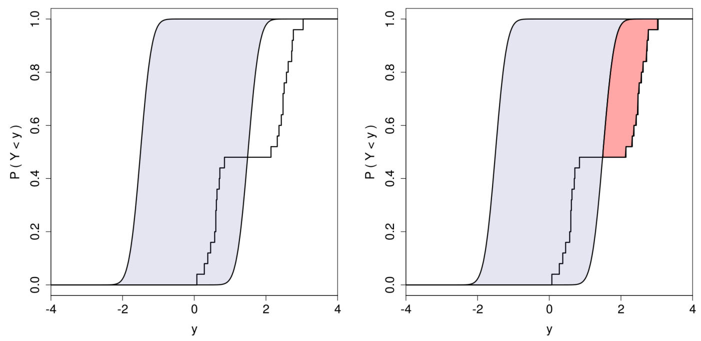


Figure 4.1: **Left:** Example of the p-box with an empirical CDF . **Right:** The p-box with the area of discrepancy between the ECDF and the p-box shaded in red. The area metric is the calculated area of the red region.

the area metric over the Kolmogorov-Smirnov or Cramer-von Mises test statistic. While the statistic is in the same units as the QOI, as stated in Ferson *et al.* (2008), “This area is thus a function of the shapes of the distributions, but is not readily interpretable as a function of the underlying random variables”. Presumably because the interpretation of the magnitude of the statistic is unclear, Ferson *et al.* (2008) and Roy and Oberkampf (2011) give no guidance for choosing a threshold for determining when the discrepancy is large enough to consider the model invalid. Oberkampf and Roy (2010) use the relative size of the metric as a way to compare the quality of fit of different models, rather than testing the validity of a particular model and suggest that if one is interested in statistical evidence against a model, that standard statistical tests, such as the Kolmogorov-Smirnov test, be used.

A second drawback relates to using the convex hull of the p-box as the basis for model validation. The issue is that the ECDF of field observations may be contained within the p-box, without being similar to any individual CDF. As a result, the simulator, and associated uncertainty, may not be capable of generating the output CDF but would pass the validation test regardless.

For instance, let us look at a simple example without a simulator to illustrate this issue. Consider the case that a hypothesized distribution is Gaussian with the mean only known to be located in the range $\mu \in [-1.5, 1.5]$ and a known standard deviation of 0.25. Figure 4.2 shows the p-box for this case, with the interior of the convex hull in gray. Suppose that the true generating distribution of the data was a from a mixture of two Gaussians, one with mean -1 and one with mean 1, each with a standard deviation of 0.25 and having mixture probability of 0.5. The plot on the left of Figure 4.2 shows the ECDF of a sample of size $n = 25$ from that mixture distribution and the right shows with sample size $n = 1000$, which is a close approximation for the true CDF for the mixture of two Gaussian distributions. It is clear that the both empirical distributions fall well within the p-box. By the p-box method, this model would be viewed as valid. However, the ECDF for n=25 and the true CDF resembles none of the individual CDF from the hypothesized distributions. This example will be discussed further in Section 4.3.

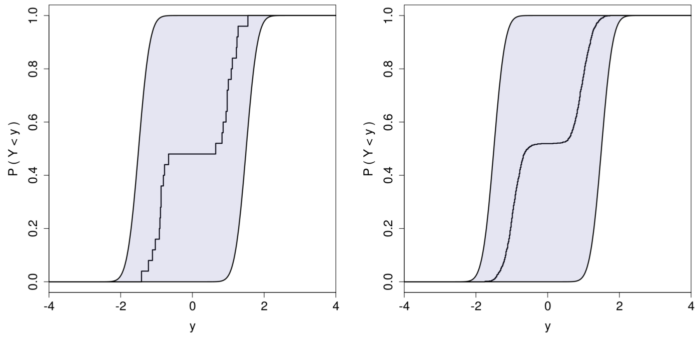


Figure 4.2: P-Box with interior shaded for case of normally distributioned observations with unknown mean constrained to an interval. The ECDF for a mixture of two normal distributions is shown for the $n = 25$ case on the left and $n = 1000$ on the right. In both cases the p-box indicates no discrepancy between the empirical and hypothesized distribution, despite the generating distribution not being in the hypothesized class.

As an aside, in addition to model validation, the p-box can be used for reliability assessment without reference to the observed ECDF. The bounds of the convex hull can be used to bound the probability that y is over or under a given reliability threshold given the epistemic uncertainty related to $\mathbf{t}$. For instance, for the p-box in Figure 4.1, the upper bound on $P(y \leq -1.5)$ is 0.5, which can be found by identifying the boundary of the p-box at $y = -1.5$. Similarly, the upper bound on the value of y_{thresh} such that $P(y \leq y_{thresh}) = 0.5$ is 1.5, which can be found by finding the rightmost edge of the p-box where the y-axis is equal to 0.5. The discussed issues exist with use of the p-box for model validation arise due to the fit between the p-box and the observed ECDF. The p-box for reliability uses only statements about the bounds of the p-box and so avoids limitations concerning the comparison with the ECDF.

4.2 Bayesian Goodness-of-Fit Decomposing Epistemic and Aleatoric Uncertainty

In order to avoid the drawbacks described in Section 4.1.5, a method that uses statistical goodness-of-fit tests with Bayesian model checking to compare the predicted distribution of the QoI with observations is proposed to assess the epistemic uncertainty in the goodness-of-fit. In Section 4.2.1 background on posterior p-values for a Bayesian model is presented and a goodness-of-fit procedure to assess the ability of the simulator and statistical model to generate data consistent with field observations is proposed. A formal test is proposed in Section 4.2.2 to allow for a validation assessment with conservative asymptotic control over the probability of a false rejection when the model is valid.

4.2.1 Bayesian Model Validation With Predictive P-values

Bayesian predictive p-values have a long history of association with model assessment for Bayesian computation (Gelman and Rubin, 1992; Gelman *et al.*, 1996, 2013). The idea is that (i) a Bayesian model specifies a probabilistic generating process for the observable data and, (ii) if the model is valid, data generated from the model should be quantitatively similar to field data. The steps of posterior predictive model checking are as follows (Gelman *et al.*, 2013):

1. Obtain samples from the posterior distribution for model parameters.

2. Generate synthetic data from probabilistic model.

3. Compute a measure of discrepancy between the data generated from the probabilistic model and observed data, either

    - using a classical hypothesis test for a measure of discrepancy, or
    - calculating the fraction of statistics from the synthetic data that are "more extreme" than the observed - the posterior predictive p-value.

The measure of discrepancy should be sensitive to some aspect of the model of scientific interest.

For this setting, using the posterior distribution for the model parameters, the posterior predictive model for the calibration data can be written:

$$\begin{aligned}
\theta &\sim \pi(\theta \mid y_{c,f}),\\
\mathbf{X}_c &\sim \pi(\mathbf{X}_c),\\
Y_{c,f} &= \eta_c(\mathbf{X}_{c,f}, \boldsymbol{\theta}) + \epsilon, \text{ and}\\
\eta_c(\mathbf{X}_{c,s}, \mathbf{t}) &\sim GP(\mathbf{0}, \boldsymbol{\Sigma}).
\end{aligned} \tag{4.3}$$

For a given set of calibration parameters and a realization from the GP emulator, the distribution for the inputs $\mathbf{X}$ can be propagated through the emulator to give an estimate of the CDF of the calibration data: $\tilde{F}_{c,f}(y \mid \boldsymbol{\theta})$. Here, we considered the case with only stochastically varying physical inputs $\mathbf{X}_{c,f}$ and no known inputs $\mathbf{x}_{c,f}$. We will discuss the case with both $\mathbf{X}_{c,f}$ and $\mathbf{x}_{c,f}$ in the discussion in Section 4.5.

Given a set of M samples from the posterior distribution for the calibration parameters and response surface, there is a collection of estimated distributions generated from the model: $\{\tilde{F}^{(1)}_{c,f}(y), ..., \tilde{F}^{(M)}_{c,f}(y)\}$. The collection of estimated CDFs can be compared to the ECDF of the observed calibration data using one of many classical tests for goodness-of-fit (see D'Agostino and Stephens (1986) for a coverage of classical goodness-of-fit tests). For any of these tests, a test statistic is compared to its known sampling distribution if the data were generated from a process with a hypothesized distribution function, $F(y)$.

Unlikely values of the test statistic given the sampling distribution indicate evidence that the predictive distribution from the model is not consistent with the observed data. For the examples in Section 4.3 and 4.4, we will use the Kolmogorov-Smirnov (K-S) test. The motivation for that choice is discussed in Section 4.2.3.

Applying the K-S test to the set of CDFs generated from the model gives a set of p-values such that $p_j = P\left(D > d(\tilde{F}^{(j)}_{c,f}(y), \hat{F}_{c,f}(y))\right)$ for $j = 1, ..., M$, where $d(\tilde{F}^{(j)}_{c,f}(y), \hat{F}_{c,f}(y))$ is the observed K-S test statistic and D is a random variable distributed according to the distribution of the K-S test statistic under the null hypothesis that observed ECDF comes from the hypothesized distribution (or that the distributions are equal in the case of a two-sample test).

Let $\{p_1, \ldots, p_M\}$ be the set of p-values obtained from application of the K-S test with the null hypothesis:

H_0: The field data and sample from $\tilde{F}^{(j)}_{c,f}(y)$ have the same distribution function for each of the $j = 1, \ldots, M$. The resulting set of p-values can be summarized graphically using histograms or plotting the p-value against each corresponding values of θ.

The posterior probability, conditioned on the calibration data, of rejecting the model can be estimated by the fraction of p-values which would reject the model. Put another way, this posterior probability is the probability that the model would be rejected, given the epistemic uncertainty in the parameters and emulation in the model. While these summaries give exploratory views of the quality of fit of the model, a formal test will be discussed in Section 4.2.2.

Using the same data to estimate parameters and perform hypothesis testing can make the test conservative (Durbin, 1973; Bayarri and Berger, 2000), therefore a set of validation data independent from the calibration data is valuable to statistically assess the model. Given the posterior distribution for the calibration parameters, the hierarchical model for the data in the validation experiment is:

$$\begin{aligned}
\theta &\sim \pi(\theta \mid y_{c,f}), \\
\mathbf{X}_v &\sim \pi(\mathbf{X}_v), \\
y_{v,f} &= \eta_v(\mathbf{X}_{v,s}, \boldsymbol{\theta}) + \epsilon, \text{ and} \\
\eta_v(\mathbf{X}_{v,s}, \mathbf{t}) &\sim GP(\mathbf{0}, \boldsymbol{\Sigma}),
\end{aligned} \tag{4.4}$$

where $\eta_v(\mathbf{x}, \mathbf{X}, \mathbf{t})$ is the simulator in the configuration for the validation system and $y_{v,f}$ is the validation field observations. Here ϵ has the same distribution as in the calibration set. This would be a strong assumption to make and the discussion will return to handling validation set specific parameters in Section 4.2.4. We can again generate a set of predicted distribution functions by sampling through the model,$\{\tilde{F}^{(1)}_{v,f}(y), ..., \tilde{F}^{(M)}_{c,f}(y)\}$. For sample j, the null hypothesis that $y_{v,f}$ have distribution $F^{(j)}_{v,f}(y)$ can be tested using p-values from the K-S test: $p_j = P\left(D > d(\tilde{F}^{(j)}_{c,f}(y), \hat{F}_{c,f}(y))\right)$ for $j = 1, ..., M$. Each of these p-values is from a valid classical hypothesis test for the specific set of parameters. The collection can again be summarized graphically or by statistics of their distribution.

The histograms of the posterior distribution of p-values and the estimated posterior probability of rejecting the null hypothesis can be used as summaries to indicate the validity of the model. The posterior probability of rejecting the null hypothesis indicates the probability, as a measure of epistemic uncertainty, that prior information and calibration data have given to model parameters that are inconsistent with the observations in the validation test. A high posterior probability of rejecting the model gives evidence against the model validity. However, these posterior summaries may be deceptive in judging the model's validity when the parameter and emulator uncertainty are still large. Figure 4.5 shows an example of these p-values plotted against the unknown model parameter for the synthetic example. For the plots in the upper row of Figure 4.5, although the model generating the observed data is in the set of hypothesized models in the example, nearly all the p-values shown are small and indicate the model should be rejected. Because the uncertainty in the value of the unknown parameter is large compared to the range of parameters consistent with the validation data, posterior probability of rejecting the model can be large despite the model being valid.

A reasonable interpretation of the collection of p-values in Figure 4.5 is that the remaining uncertainty in the calibrated model is large and incorporating additional data into the calibration may greatly improve the model. Depending on the application, it may be prudent to reject the validity of the model in this case, as predictions from the model with the large uncertainty may be poor reflections of future observed data. However, this is an application specific decision. In the next section, a formal test is introduced for evidence that the simulator is unable to generate the observed output distribution.

4.2.2 Formal Validation Test

The goal of the validation test is to assess any evidence that the model is inconsistent with observed data. Typically a decision must be made regarding whether to trust the model given that evidence. Statistical tests provide procedures with known properties to justify the those decisions.

To test the null hypothesis that the observed field data is generated from the predictive distribution from the model for some set of calibration parameters, in the presence of epistemic uncertainty in the values of the calibration parameters and emulation of the simulator, we propose the following procedure:

1. Sample CDFs as in Section 4.2.1.

2. Perform goodness of fit test (K-S) to compare validation observations with each generated CDF.

3. Identify p_γ, the maximum p-value over a $100(1-\gamma)\%$ credible interval for the calibration parameters and emulator response.
4. Reject the model and conclude evidence against the model validity if $p_\gamma + \gamma \leq \alpha$.

If a model is not rejected, one concludes is that there is not evidence that predictive distribution is inconsistent with the observed data. The parameter γ specifies the coverage to the credible interval. From Step 4, it is clear that if $\gamma > \alpha$, the test will never reject.

In classical testing of composite hypotheses with a nuisance parameter θ, the p-value of a test will depend on the value of the nuisance parameter. Berger and Boos (1994) show that building a $100 * (1-\gamma)\%$ confidence interval for θ and then rejecting the hypothesis test if the maximum p-value over that interval is less than $\alpha + \gamma$ is a valid level α hypothesis test. Because of the emulation and typically small sample sizes of experimental data, valid confidence intervals are rarely practical (Easterling and Berger, 2002). Instead the $100(1-\gamma)\%$ Bayesian credible interval, calculated using the posterior samples, is used. To do so we use the $100(1-\gamma)\%$ highest posterior probability samples to estimate this interval and maximize the p-value of the goodness-of-fit test over the interval.

With some assumptions on the prior distributions, most notably that the true value for the parameters are in the support, the Bayesian credible interval will asymptotically have the specified coverage (Gelman *et al.*, 2013). Using this, it is shown that this test is a valid level α test using the same strategy for a proof given in Berger and Boos (1994):

Proposition 4.2.1. *Assume that there is a value θ_0 for which the null hypothesis that the model CDF is equal to the CDF of the field data is true. Let C_γ be a $100(1-\gamma)\%$ credible interval for θ and $p_\gamma = \sup_{\theta \in C_\gamma} p(\theta) + \gamma$ where $p(\theta)$ is the p-value of the chosen level α hypothesis test as a function of the calibration parameter θ. Then a test to reject the null hypothesis when $p_\gamma < \alpha$ will falsely reject the null with probability $\leq \alpha$.*

Proof.

$$\begin{aligned}
P(p_\gamma \leq \alpha) &= P(p_\gamma \leq \alpha, \theta_0 \in C_\gamma) + P(p_\gamma \leq \alpha, \theta_0 \notin C_\gamma) \\
&\leq P(p_\gamma \leq \alpha, \theta_0 \in C_\gamma) + P(\theta_0 \notin C_\gamma) \\
&= P\left(\sup_{\theta \in C_\gamma} p(\theta) + \gamma \leq \alpha, \theta_0 \in C_\gamma\right) + \gamma \\
&\leq P(p(\theta_0) + \gamma \leq \alpha) + \gamma \\
&= P(p(\theta_0) \leq \alpha - \gamma) + \gamma \\
&= \alpha - \gamma + \gamma \\
&= \alpha.
\end{aligned}$$

□

In the finite data case, the coverage will depend on the exact specification of the statistical model including the prior distributions. In this case, the typical arguments for Bayesian quantification of uncertainty motivate use of the credible intervals. When the distribution for a parameter is uniform, such that there is no reasonable preference for any $100 * (1 - \gamma)$ interval in the support of the distribution, we recommend simply sampling over the entire uniform distribution. The distribution for a parameter is typically only uniform before performing any model calibration, as the calibration process will typically be informative in some manner for a parameter.

The test from Berger and Boos (1994) is known to be conservative (Bayarri and Berger, 2000), however utilizing this test allows for asymptotic control of the probability of falsely rejecting the model in the decision. This control supplements the use of Bayesian predictive summaries discussed in the Section 4.2.1. Algorithm 2 displays the discussed Monte Carlo approach.

Algorithm 2 Algorithm for validation of Bayesian predictive distributions from a model with epistemic and aleatoric uncertainty.

```
procedure BAYESIAN DISTRIBUTION VALIDATION($\alpha, \gamma : \alpha > \gamma$)
    Obtain $M$ posterior samples of parameters $\theta$ with epistemic uncertainty
    Calculate ECDF of the data:
        $\hat{F}(y) = \frac{1}{n} \sum_{j=1}^{m} \mathbf{I}(y_j \leq y)$
    for i=1 to $M$ do
        Draw $m$ samples of $X$ from their aleatoric probability distribution: $f(x \mid \theta_i)$
        Generate $m$ simulated values $y_1^{(s)}, \ldots, y_m^{(s)}$ from the statistical model $f(y \mid \theta, x)$ including emulator uncertainty
        Approximate the model CDF using $m$ samples:
            $\hat{F}_i(y) = \frac{1}{m} \sum_{j=1}^{m} \mathbf{I}(y_j^{(s)} \leq y)$
        Calculate test statistic, $GOF(\cdot)$ between experimental and simulated CDFs:
            $T_i = GOF(\hat{F}_i(y), \hat{F}(y))$
        Calculate p-value, $p_i$
    Retain $100 * (1 - \gamma)\%$ highest posterior probability samples
    $M_\gamma \leftarrow$ number of posterior samples retained
    Calculate fraction of samples for which GOF rejects the model
        $r = \frac{1}{M_\gamma} \sum_{j=1}^{M_\gamma} \mathbf{I}(p_j + \gamma < \alpha)$
    if $r == 0$ then
        Reject model
    else
        Report $r, \{p_1, \ldots, p_{M_\gamma}\}$
```

4.2.3 Choice of Goodness-of-Fit Test

As mentioned in Section 4.2.1, there are many goodness-of-fit tests that could be used in the proposed procedure. The one-sample Kolmogorov-Smirnov test uses the maximum distance between the ECDF and the hypothesized CDF as a test statistic (Smirnov, 1948):

$$d = \sqrt{n} \sup \left| \hat{F}(y) - \tilde{F}(y) \right| .$$

If $\tilde{F}(y)$ is also approximated by a sample, the two-sample K-S test can be used:

$$d = \sqrt{\frac{n_1 n_2}{n_1 + n_2}} \sup \left| \hat{F}(y) - \tilde{F}(y) \right| ,$$

where n_1 and n_2 are the number of samples used to estimate $\hat{F}(y)$ and $\tilde{F}(y)$ respectively (Smirnov, 1948).

The distribution of the test statistic is independent of the hypothesized distribution function in the one-sample case (D'Agostino and Stephens, 1986), and is a general test of discrepancy between an estimated and hypothesized distribution. Because the empirical distribution is tightly constrained near zero or one in the tails, the K-S test is weakly sensitive to variation in these locations.

The Anderson-Darling (AD) test improves the power of the test to detect discrepancy in the tails of the distribution by using the integrated, squared difference between the CDFs and adding a weight to the integral that depends on y:

$$d = n \int_{-\infty}^{\infty} \left| \hat{F}(y) - \tilde{F}(y) \right|^2 [\tilde{F}(y)(1 - \tilde{F}(y))]^{-1} d\tilde{F}(y).$$

The weight allows the test to be sensitive in the tail variation, but relies on having a closed form for the model CDF. Because the model CDF must commonly be estimated using Monte Carlo for simulators, see Section 4.1.3, we did not feel this test would be generally feasible in our applications.

The χ^2 goodness-of-fit test is another alternative to the K-S test. The domain of y is partitioned into K bins. For bin k with boundaries ℓ_{b_k} and u_{b_k}, the expected fraction of observations that one would expect to fall into the bin is $\tilde{p}_k = \tilde{F}(u_{b_k}) - \tilde{F}(\ell_{b_k})$. Given a sample of size n, the χ^2 test statistic is:

$$\chi^2 = \sum_{k=1}^{K} \frac{(o_k - np_k)^2}{np_k}$$

where o_k is the observed number of observations in bin k. Given a true, simple null, this statistic has a χ^2_{K-1} sampling distribution. There are multiple strategies for how to bin the domain, which are discussed in D'Agostino and Stephens (1986). To optimize the power of the test against general alternatives, D'Agostino and Stephens (1986) recommend equal probability bins with the number of bins to be chosen based on the sample size. The bins can also be chosen to increase power against deviations in specific regions.

For composite null hypotheses with parameters with epistemic uncertainty, choosing the parameters to minimize the χ^2 test statistic or equivalently, to maximize the multinomial likelihood leads to a χ^2 test with $K - p - 1$ degrees of freedom instead of $K - 1$, where p is the number of parameters (D'Agostino and Stephens, 1986). There is a subtle problem with using this for our application, which arises because of the epistemic uncertainty due to emulating the simulator. This is uncertainty in the response surface at any location in the domain that the simulator has not been evaluated, effectively adding infinite parameters to optimize the χ^2 test statistic. Additionally, for a GP covariance function like the squared-exponential, if there is any smooth, bounded function for which the expected bin probabilities equal the observed and the aleatoric input uncertainty is not a point mass at an observed value of the simulator, the support of the Gaussian process emulator is large enough to allow response surfaces that can make the χ^2 test statistic almost arbitrarily small. The instances of the GP that reach that minimum may have extremely low probability, but would still be an optimal solution if optimizing over the entire support.

4.2.4 Handling Validation Specific Parameters

It may be the case that the statistical model for the validation experiment contains parameters that are not able to be informed by the calibration experiment. For example, consider:

$$\begin{aligned}
X \sim& N(\mu, \sigma^2), \\
\epsilon_c \sim& N(0, \sigma^2_{\epsilon_c}), \\
\epsilon_v \sim& N(0, \sigma^2_{\epsilon_v}), \\
Y_c =& f_c(X) + \epsilon_c, \text{ and} \\
Y_v =& f_v(X) + \epsilon_v.
\end{aligned}$$

In this case, the distribution of Y_v depends on ϵ_v, which is not informed by observations of Y_c. This can be addressed in a few ways:

- The parameters that govern ϵ_v can be drawn from their prior distribution. This may give a high probability to poor values for ϵ_v if the prior distribution is wide.
- Design a pilot experiment to obtain a posterior distribution for ϵ_v given the pilot data.
- Estimate the posterior distribution for ϵ_v using a statistic ancillary to the calibrated model. For instance, ϵ_v in the example above could be estimated using replicates with a center point design in the validation experiment, whose sample variance only depends on $\sigma^2_{\epsilon_v}$ and not the rest of the model.

4.2.5 Validation and Coherent Updating

Given that a model has been shown to generate a predictive distribution consistent with the validation data for some set of sampled parameter values, the natural question that arises is whether future predictions should use only those parameters, rather than the full calibration distribution.

The proposed approach separates the tasks of calibration and validating the model. Calibration uses Bayesian updating to account for epistemic uncertainty in a coherent manner within the framing of the statistical model. The task of validation is meant to use data to probe whether the statistical model is reasonable and as such, in a sense stands outside of the model.

If the model should be updated to take advantage of information in the validation data, the coherent approach is to recalibrate the model using Bayesian updating to obtain new posterior distributions. Prediction accounting for the epistemic uncertainty in the model is consistently done using the Bayesian model.

4.3 Validation Examples

4.3.1 Single Parameter Gaussian Example

Assume that the experimental inputs, $\{X_1, ..., X_n\}$, are hypothesized to be generated *iid* from a normal distribution with a mean of 0 and standard deviation $\sigma = 0.25$. Two synthetic experiments will be conducted, a calibration experiment and a validation experiment. For this example, a very simple simulator will be used with a difference in functional form between the calibration and validation system, shown in Equation 4.5. *A priori* plausible values for t are known to lie within the interval $[-1.5, 1.5]$, but beyond that no other information is known. Both the calibration and validation experiment are collected with sample sizes $n = 25$ and $n = 1000$, respectively. Three examples will be considered - one where the simulators given in Equation 4.5 and assumed normal model for the input are valid, one where the simulator is invalid, and one where the hypothesized input distribution is not valid.

$$\begin{aligned} \eta_c(X,t) &= (X+t) \text{ and} \\ \eta_v(X,t) &= (X+t)^3. \end{aligned} \tag{4.5}$$

For both, first we perform a validation check using the *a priori* interval for θ with the proposed method and compare to the p-box formulation to check validity of the assumed Normal model. We then update the information about θ to the Bayesian posterior distribution. After updating, the validity of the model will be tested both with the same calibration data and using the sample from the validation experiment. In all applications of the proposed validation test, we use a $\gamma = 0.01$.

4.3.2 Case: Valid Model

Let the $y_{c,1}, \ldots, y_{c,n}$ be generated *iid* from the hypothesized model and let $\theta = 1$ where θ is the value of t used to generate the field data. We first construct the p-box as described in Section 4.1.5. The output of the calibration distribution is Normal in this example with t controlling the location, so the p-box can be computed analytically. Figure 4.3 shows the p-box using the prior interval. The ECDF for both the $n = 25$ and $n = 1000$ samples both fall entirely inside the p-box, leading to an area metric value of 0, giving no evidence for invalidity of the model.

We then perform the proposed Bayesian predictive test procedure. We first sample from the Uniform(-1.5,1.5) prior distribution for θ. For each sample $\theta^{(m)}$, we sample from the hypothesized normal distribution for $\mathbf{X}$ and use this sample and the simulator $\eta_c(X, t)$ from Equation 4.5 to estimate the hypothesized CDF of Y_c. The top row of Figure 4.4 and Figure 4.5 show summaries from this *a priori* validation check for the $n = 25$ and $n = 1000$ examples. The top row of Figure 4.4 shows $M = 200$ sampled model CDFs and the ECDF of the synthetic data. It is clear visually that a set of model CDFs in blue closely resemble the ECDF of the calibration data in red.

Because the prior distribution is uniform, as recommended we do not limit the testing procedure to only a $100(1-\gamma)\%$ interval. The top row of Figure 4.5 show the p-values resulting from the Kolmogorov-Smirnov test as a function of the sampled model parameter θ. Because only predicted CDFs generated by parameters in a small range around $\theta = 1$ were consistent with the observations, 92% and 97% of p-values would be rejected in a level $\alpha = 0.05$ test. Despite this, the maximum p-value from the test is well over 0.05, so we, correctly, do not find evidence against the validity of the model.

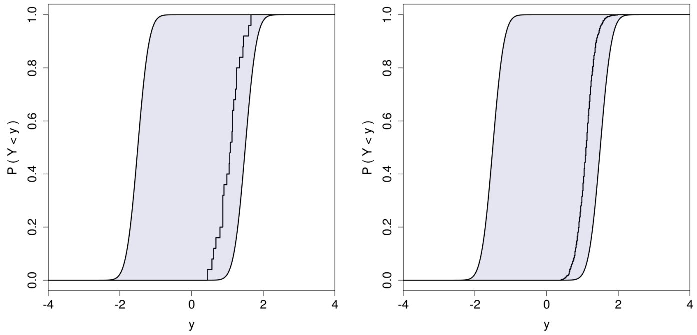


Figure 4.3: P-Box showing the ECDF entirely contained within the convex hull of the hypothesized CDFs. The sample size is $n = 25$ in the left panel and $n = 1000$ in the right panel.

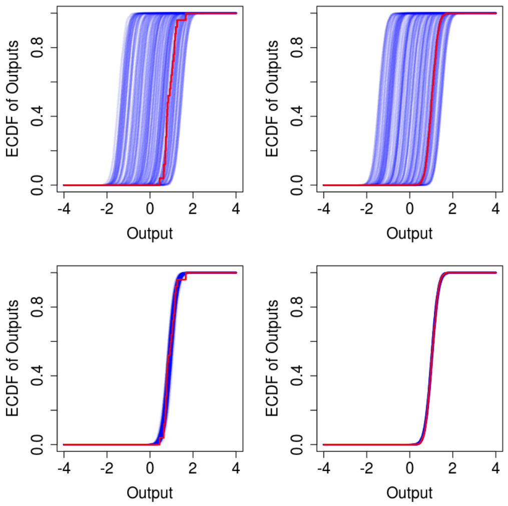


Figure 4.4: Collection of CDFs from null. The top row contains sampled CDFs from the prior distribution for θ and the ECDF of the synthetic data for the $n = 25$ and $n = 1000$ cases. The bottom row uses the same two sample sizes, but using samples from the posterior distribution for θ after calibration.

The calibration data provides information on the model parameter θ, so through Bayesian updating, we can obtain a posterior distribution: $\pi(\theta \mid y_c)$. Sampling from this posterior distribution, we generated a set of predicted CDFs for the calibration data, shown on the bottom row of Figure 4.4. The spread of the predicted CDFs has greatly decreased due to the additional information. We then retain only the 99% highest posterior probability samples for the proposed testing procedure, which corresponds to $\gamma = 0.01$. The bottom row of Figure 4.5 shows the generated p-values from Algorithm 2 as a function of θ. The majority of the resulting p-values fail to reject at a level $\alpha = 0.05$ and it's clear there is little evidence that the model is not valid.

We can then move on to testing the model against a set of data using the validation experiment. In this case, we will propagate the uncertainty in X through the function $\eta_v(X)$ for different values of θ from the posterior distribution $\pi(\theta \mid y_c)$. The bottom row of Figure 4.6 shows the generated CDFs in blue compared to the ECDF of the validation data in red. The predicted CDFs appear visually consistent with the ECDF. The top row of Figure 4.6 shows a histogram of the p-values from the test, with a large fraction being greater than $\alpha = 0.05$. This validation test again indicates, correctly, that there is no evidence that the model is invalid.

4.3.3 Case: Invalid Simulator

In this synthetic example, the true relationship between X and the observed variable is shown in Equation 4.6 below, which is distinct from Equation 4.5 for any value of t. Because only the outcome y_c is observed, and not the inputs X, we cannot directly compare

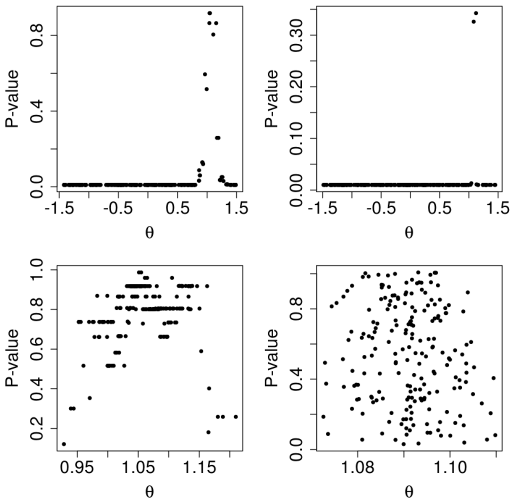


Figure 4.5: P-value of the K-S test for each sampled CDF as a function of the sampled parameter θ. The top row uses the prior distribution for θ with $n = 25$ and $n = 1000$ observations uses to test. The bottom row uses the same two sample sizes, but using samples from the posterior distribution for θ.

predictions from the functions in Equations 4.5 and 4.6. Again, the validation test will be performed *a priori*, followed by updating the information about θ using the calibration data. We then test the validity of the model using both the calibration and validation data. As in the previous example, we also compare the ECDF of the calibration data to the p-box defined using the prior interval on θ.

$$\begin{aligned} y_c = \eta_c(X,t) &= 0.7\sin(\frac{2\pi}{0.2}(X+0.8))/(X+t-0.2) \text{ and} \\ y_v = \eta_v(X,t) &= 0.343\sin^3(\frac{2\pi}{0.2}(X+0.8))/(X+0.8)^3. \end{aligned} \tag{4.6}$$

In Figure 4.7, the ECDFs of the mixture model are nearly contained within the p-box for both samples. For $n = 25$ there is a small area in which the ECDF protrudes outside the p-box, with an area metric value of 0.028. This appears to be minimal evidence against the model. In the $n = 1000$ case, there is the p-box procedure found no evidence at all for the model to be invalid.

The top row of Figure 4.8 shows the same summaries as the last example from the Bayesian predictive validation procedure. The two images on the top left show $M = 200$ sampled model CDFs with the parameters drawn from the prior distribution for θ. It is clear visually that the ECDF in red does not share a common form with any of the predicted CDFs.

The top right plots again show the p-values resulting from the proposed validation test as a function of the sampled model parameter θ. In this case, all generated p-values reject the

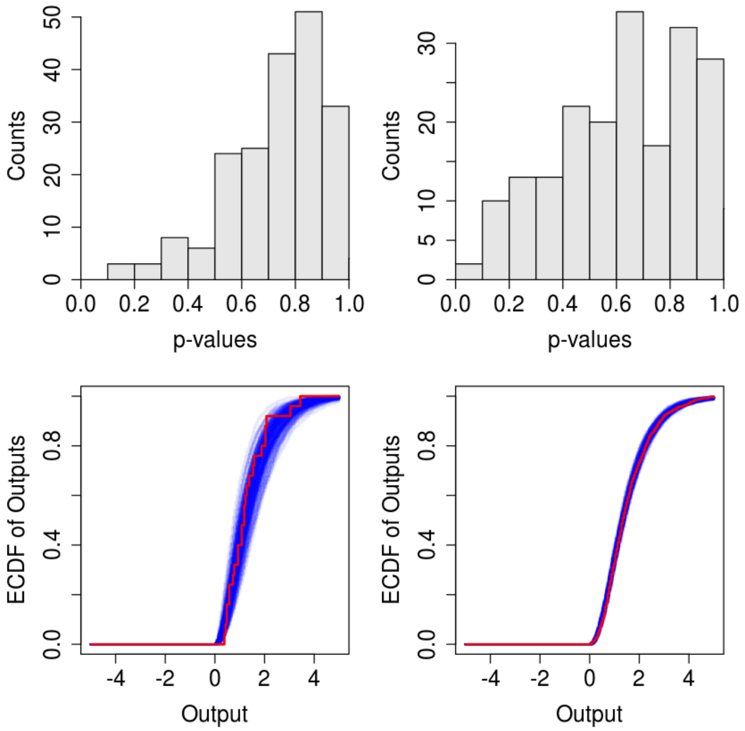


Figure 4.6: **Top:** Histogram of p-values from validation experiment for small data (left) and large data (right) case. **Right:** Empirical CDF of validation data along with the collection of model CDFs in small data (left) and large data (right) cases.

validity of the hypothesized model at a level $\alpha = 0.05$. For the $n = 1000$ case, the p-values are all precisely equal to γ indicated that the p-value from each individual Kolmogorov-Smirnov test was equal to 0. This was not the case for $n = 25$, however the p-values peak well short of the critical threshold, indicating evidence that the model is not valid.

By constraining the range of sampled values of θ through calibration, values of θ may be found that are able to generate distributions consistent with the data. Sampling from the posterior distribution given the observed y_c, we generated a set of predicted CDFs for the calibration data, shown on the bottom left of Figure 4.8. The generated CDFs show little overlap with the empirical distribution. The bottom right plots show the generated p-values from the proposed method as a function of θ. The p-values still fall well short of the critical $\alpha = 0.05$, so there is again clear evidence that the model is not valid. At this point, the model is clearly rejected, however, we will continue to the validation step for pedagogical purposes.

As in the previous example, we will propagate the uncertainty in X through the function $\eta_v(X)$ for different values of θ from the posterior distribution $\pi(\theta \mid y_c)$. The bottom row of Figure 4.9 shows the generated CDFs in blue compared to the ECDF of the validation data in red. Again the distribution functions are entirely inconsistent with one another visually. Similarly, corresponding p-values for the goodness-of-fit test for each sample CDF were

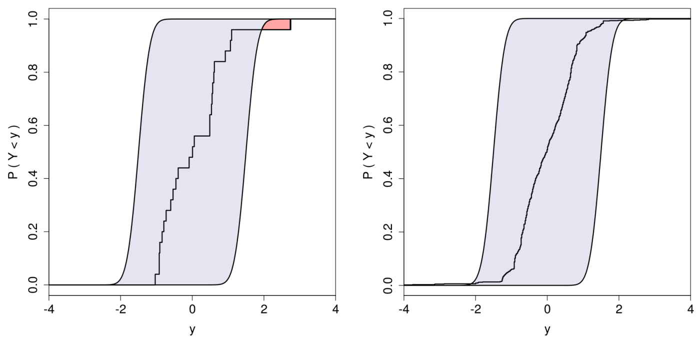


Figure 4.7: P-Box showing the ECDF entirely contained within the convex hull of the hypothesized CDFs

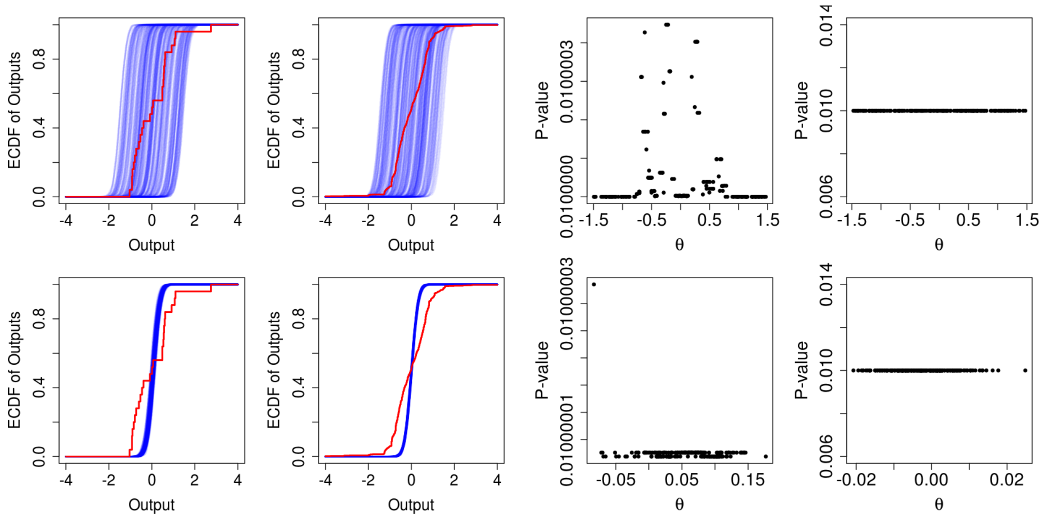


Figure 4.8: **Left:** Collection of CDFs **Right:** Plot of p-values vs. value of the parameter

computed. Histograms of the p-values from the test are shown in the top row of Figure 4.9. For both the $n = 25$ and $n = 1000$ case, the p-values are much smaller than the $\alpha = 0.05$ threshold, showing clear evidence that the model is invalid.

4.3.4 Case: Invalid Input Distribution

When testing the predictive distribution against observed data, the validation test is probing the entire model, not just the simulator. If the input distribution is incorrect, the propagated uncertainty through the simulator may lead to poor predictions of the output distribution. In this example, the simulator will be correct, however, rather than a single normal distribution, the input data, X, is instead drawn *iid* from a mixture of two Gaussians, one with a mean of 0 and the other with a mean of -2 and both components of the mixture having equal probability. We follow the same procedures as the previous two examples.

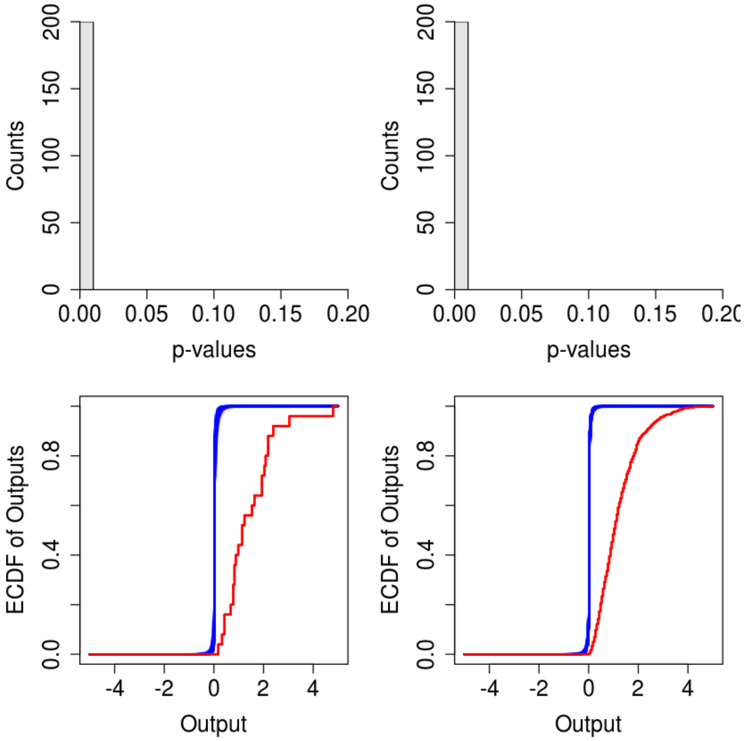


Figure 4.9: **Top:** Histogram of p-values from validation experiment for small data (left) and large data (right) case. **Right:** Empirical CDF of validation data along with the collection of model CDFs in small data (left) and large data (right) cases.

In Figure 4.2, the ECDFs of the mixture model are contained entirely within the p-box for both samples. The p-box procedure again finds no evidence at all for the model to be invalid, even though it is clear to see that the ECDF does not appear to be normal.

The top row of Figure 4.10 shows same summaries as the last examples from the Bayesian predictive validation procedure. The two images on the top left show $M = 200$ sampled model CDFs with the parameters drawn from the prior distribution for θ. It is clear visually that the ECDF in red displays a long plateau that is not seen in the sampled model CDFs.

The top right plots again show the p-values resulting from the proposed validation test as a function of the sampled model parameter θ. In this case, all generated p-values reject the validity of the hypothesized model at a level $\alpha = 0.05$.

We can again update the information about θ using the calibration data. Using the posterior distribution, we obtain the CDFs shown on the bottom left of Figure 4.10. The generated CDFs are now visibly barely overlapping with the empirical distribution. The bottom right plots show the generated p-values from the proposed method as a function of θ. The p-values still fall well short of the critical $\alpha = 0.05$, so there is again clear evidence that the model is not valid. As in the previous example, the model is clearly rejected.

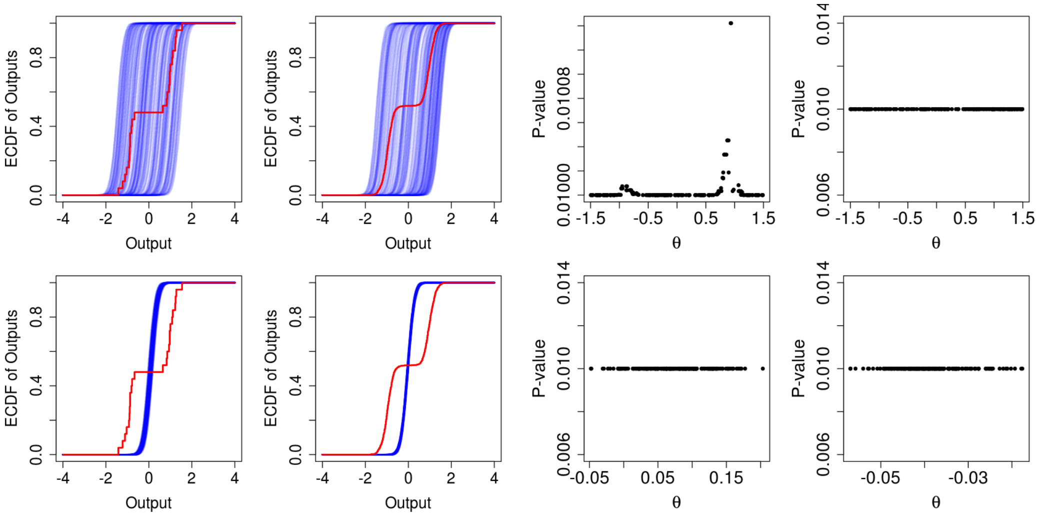


Figure 4.10: **Left:** Collection of CDFs **Right:** Plot of p-values vs. value of the parameter

4.3.5 Emulated Simulator for Validation Experiment Example

As a continuation of the previous examples, the validation simulator, instead of being simply the cube of the input X, is a slightly more complex function:

$$y_v = \eta_v(X) = \sin(\pi * (X - 1)) * e^{-1.5(X-1)}.$$

Instead of being able to arbitrarily evaluate the function, we instead emulate the simulator using a Gaussian process and $m = 10$ realizations of $\eta_v(X)$. Figure 4.11 shows 100 realizations from the Gaussian process emulator in black and the actual curve for $\eta_v(X)$ in red. The spread in the black curves reflects the lack of knowledge of the output of $\eta_v(X)$ at those locations.

Using the calibration data from the valid model example, we then generate predictive CDFs from the model for $M = 1000$ samples from the posterior distribution for θ, as done in the validation tests above. The larger size of Monte Carlo samples is beneficial to accommodate the extra variation due to emulator uncertainty. The ECDF of the validation data looks qualitatively similar to the CDFs generated propagating X through the emulator in on the left of Figure 4.12. The plots on the right of Figure 4.12 show the p-values obtained applying the proposed validation testing procedure. For both the large and small sample size validation data sets, the p-values indicated there is not evidence that the model is invalid. For the large validation set size case, a larger percentage, 93.8%, reject the null. Nevertheless, the large p-values indicate that the current validation test does not show evidence that the model is invalid.

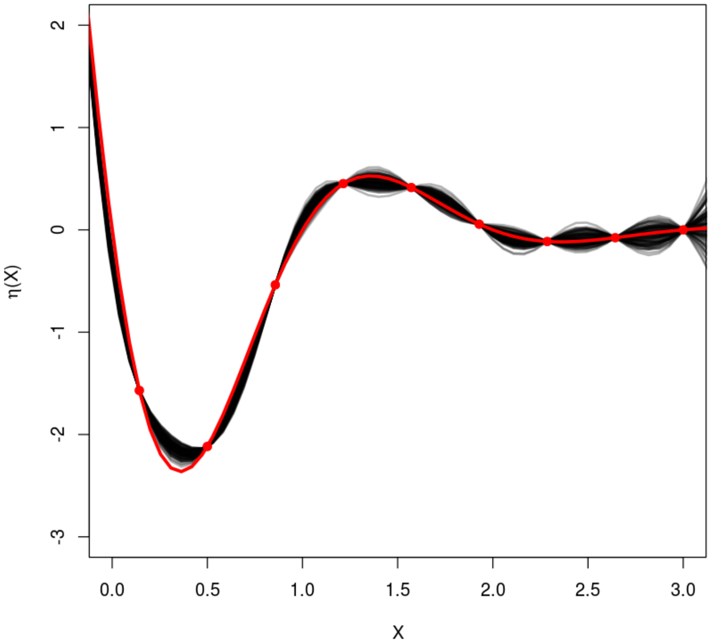


Figure 4.11: Results of emulating the validation simulator function using a Gaussian process. The curve representing the true simulator is shown in red, while 100 realizations of random functions from the Gaussian process are shown in black. The spread of the Gaussian process realizations between observed values of $\eta_v(X)$ indicate the epistemic uncertainty in the function output.

4.3.6 Testing Coverage and Power of Previous Examples

In Section 4.3.1, we walked through carrying out the validation test on a set of single instances of examples. For the remainder of this section, the examples will be repeated to investigate the performance of the proposed testing procedure by estimating the probability of false positives in examples where the model is valid and show the power to detect the invalidity of the model in the other cases, as the deviation from the hypothesized model grows.

4.3.7 False Positive Rate With the Valid Model

We can repeat the previous valid model example 1,000 times to get an estimate of the probability of obtaining a false positive given a valid model. The formal test should control the probability of false positives to be less than α, where we use $\alpha = 0.05$. We repeat the prior validation test, the validation test on the calibration data, and the validation test on the independent validation data set. For each we also repeat the $n = 25$ and $n = 1000$ sample sets.

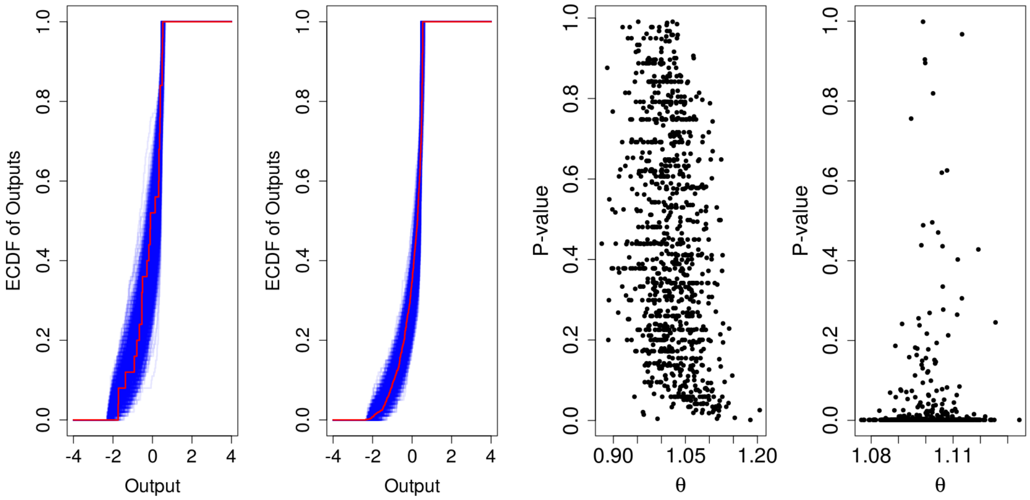


Figure 4.12: **Left:**Empirical CDF of validation data along with the collection of model CDFs using the emulator in small validation data (left) and large data (right) cases.**Right:** Plot of p-values vs. value of the parameter

Calibration				Validation			
$n = 25$		$n = 1000$		$n = 25$		$n = 1000$	
Prior	Posterior	Prior	Posterior	Prior	Posterior	Prior	Posterior
0.001	0.036	0	0	0.047	0.05	0.002	0

Table 4.1: Fraction of false positives in the proposed tests from Section 4.2.2 on 1,000 repetitions of the valid model case in Section 4.3.1

Table 4.1 shows the fraction of false rejections in the 1,000 replications. Each of the eight cases rejected less than or equal to the level $\alpha = 0.05$. In five of the eight cases, two or less replications out of 1,000 falsely rejected the model. These do indicate some evidence that the test is conservative, though the false positive probability is controlled to be less than the level of the test.

4.3.8 False Positive Rate With Emulated Simulator

Like in the previous section, we would like to show the properties of the test when the simulator must be emulated. The emulation example from Section 4.3.1 is repeated 200 times to get an estimate of the probability of obtaining a false positive given a valid, emulated model. We repeat the prior validation test and the validation test on the calibration data for both the $n = 25$ and $n = 1000$ sample sets. In Table 4.2, we see an empirical estimate of the probability of false positives when testing with level $\alpha = 0.05$ and $\gamma = 0.01$ using 200 replications of the synthetic example. For each case, the fraction of false positives is

$n = 25$		$n = 1000$	
Prior	Posterior	Prior	Posterior
0.00	0.0	0.005	0.04

Table 4.2: Fraction of false positives in the proposed tests from Section 4.2.2 on 200 repetitions of the emulated model case in Section 4.3.1

conservative; with $n = 25$, testing by propagating either the prior or posterior led to no false rejections of the model in 200 replicates.

4.3.9 Statistical Power for the Invalid Input Distribution

In the previous example in which the input distribution was invalid, the hypothesized distribution was Gaussian but the actual generating distribution for the example was a Gaussian mixture model. We showed an example where the formal test was able to successfully reject the model. In this section, we will show the power of the test to detect the invalidity as a function of the separation between modes of the mixture model.

The previous mixture model had components with means of 0 and -2 with each component of the mixture having equal probability. Each component was separated from the overall mean of -1 by 1 unit. For this section, let Δ be the distance from -1 to each mixture component center. In each case, the standard deviation for the mixture components is 0.25. We test the power of the model to detect the incorrect predictive distribution at values of $\Delta = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$.

We generate synthetic data and perform the validation test 200 times for each value of Δ. Figure 4.13 shows the fraction of correct detections of the model being invalid in each of 4 cases: the validation test using the same data used to calibrate the model with $n = 25$, the same but with $n = 1000$, the validation test using an independent validation set with $n = 25$, and the same with $n = 1000$.

4.4 Validation on Radiation Transport Model

As discussed in the previous chapter, the Center for Exascale Radiation Transport is focused on high-fidelity modeling of radiation transport. The goals of experiments at CERT are to calibrate the impurity model in PDT and show the model is valid through a sequence of experiments of increasing complexity.

One focus of this work is to utilize PDT to quantitatively predict the measured neutron-detector count rate in experiments as a function of many variables, including those whose uncertainties are judged to play important roles. CERT experiments extensively use graphite

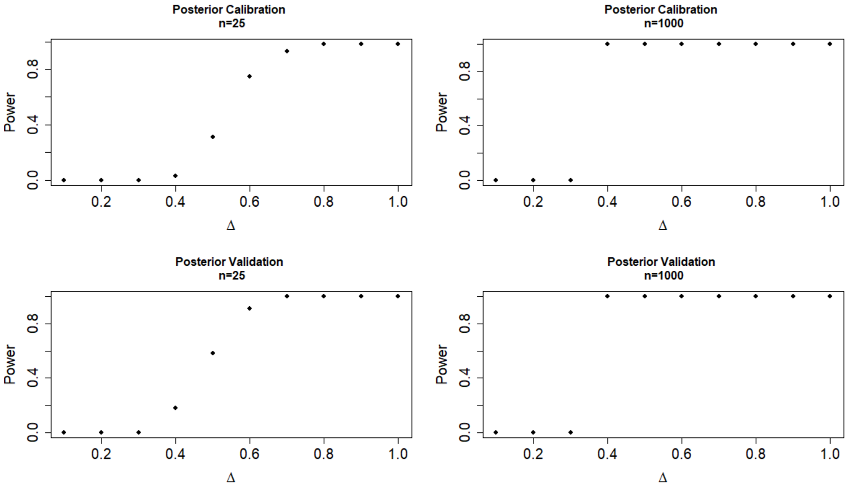


Figure 4.13: The panels show the estimated power to detect the discrepancy in the distribution function between model predictions from a Gaussian model and the generated observations from a mixture of Gaussians as a function of the separation of the mixture.

bricks whose impurities can significantly change its neutron-transport properties. The previous chapter focused on the calibration of the model, including calibrating the distribution of the impurities from brick to brick.

Understanding the distribution of impurities between bricks is critical for assessing the aleatory uncertainties for performing the model validation. In order to characterize the uncertainty in impurity for a single brick as well as the brick-to-brick variation, this calibration parameter is treated as a random effect. That is, since each brick can have different impurity concentrations, the interest lies in estimating parameters that govern the calibration distribution instead of a constant value. As mentioned previously, we will use a truncated normal distribution with hyperparameters $\mu_{imp}, \sigma^2_{imp}$ on the $[0, 1]$ interval for the population distribution.

The full model used for calibration is given in 3.8. We previously discussed the results of the calibration, so now we can move on to performing the validation tests with the model. First, we can test the model using the $n_{bricks} = 10$ observations from calibration using Algorithm 2 with $\gamma = 0.01$. Samples from the posterior distribution of the model were obtained in the calibration process and can be used for validation. The QoI for validation is the ratio of the count rate through the graphite bricks to the count rate through air.

We first drew values of parameter with epistemic uncertainty: $\boldsymbol{\theta}, \boldsymbol{\omega}_c, \lambda, \mu_{imp}$, and σ^2_{imp}. Conditioning on the drawn hyperparameters for the parent distribution of the impurities, we sampled $n = 100$ impurity values to propagate their aleatory uncertainty. Next the epistemic uncertainty in model response was incorporated by sampling the response from the Gaussian process emulator for PDT at the 100 locations. The predicted values from the emulator are combined with the sampled baseline rate λ and experimental exposure time to generate synthetic Poisson counts for both the air and graphite experiments. These $n = 100$ counts through graphite are then divided by the counts through air and the resulting collection gives an estimate of the CDF of ratio of count rates. This process is repeated $M = 10,000$ times, with the $100 * (1 - \gamma)$ highest probability estimates being kept, using the log posterior from the calibration sampling and the epistemic uncertainty from the emulation.

Figure 4.14 shows the collection of estimated CDFs in blue, with each blue curve generated at a sample from the posterior distribution after calibration of epistemic uncertain parameters. These curves are compared to the ECDF of the 10 calibration observations. The histograms showing the epistemic uncertainty in Kolmogorov-Smirnov test statistic and the p-value of the test are also shown in Figure 4.14. While many p-values are small enough to reject the model, there are values for the calibration parameters in the interval such that there is not evidence against the model, so overall there is not evidence against the validity of PDT based on the calibration data.

The large epistemic uncertainty in the test statistic seen in the bottom of Figure 4.14 indicates that more calibration data may be able to more strongly probe the model fidelity.

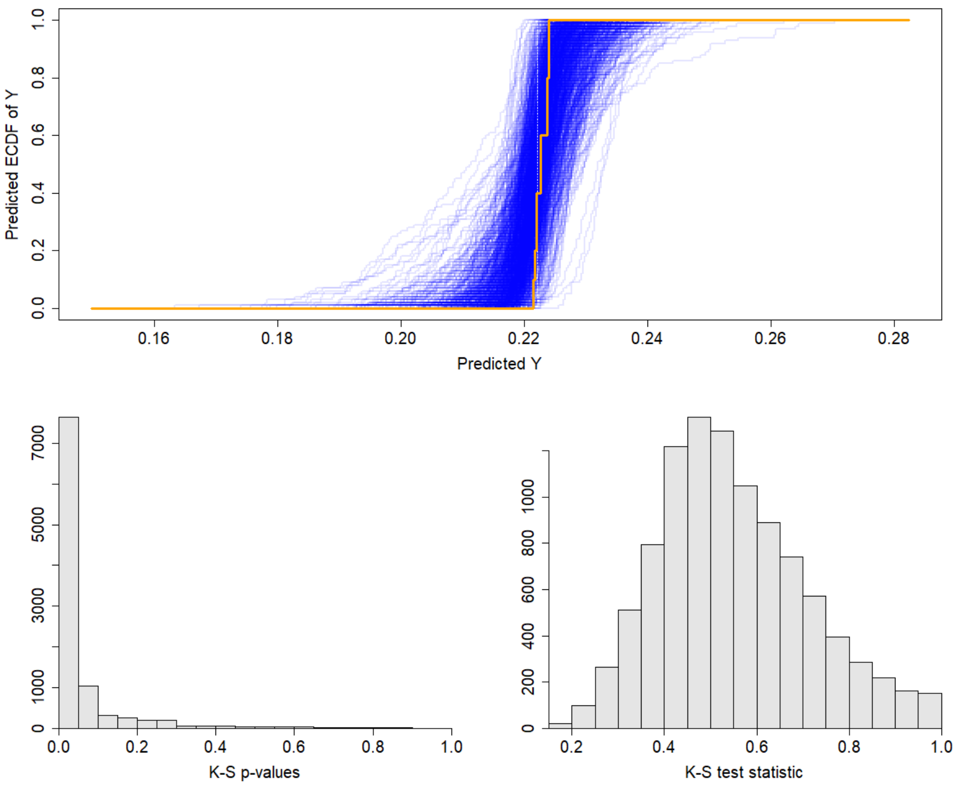


Figure 4.14: **Top:** Comparison of ensemble of model CDFs (blue) to the ECDF of the calibration data (orange) **Left:** Histogram of p-values from the above ensemble using a Kolmogorov-Smirnov test **Right:** Histogram of test statistic values from the above ensemble using a Kolmogorov-Smirnov test

A separate validation experiment consisting of $n_{bricks} = 3$ observations provides an extrapolative test of the model. We again follow Algorithm 2 with $\gamma = 0.01$ to test the model validity. The new experiment requires a new emulator for the computer model. The hyperparameters for the emulator are estimated using only the simulator results.

We again draw values of parameter with epistemic uncertainty from the posterior calibration samples: $\boldsymbol{\theta}, \lambda, \mu_{imp}$, and σ^2_{imp}. We also draw Gaussian process parameters $\boldsymbol{\omega}_c$ from the emulation in the previous paragraph. A $n = 100$ sized sample of impurity values are then drawn as before, followed by the response of the emulator and the synthetic count rate ratios. This process is also done $M = 10,000$ times, with the $100 * (1 - \gamma)$ highest probability estimates being kept.

Figure 4.15 shows the collection of estimated CDFs in blue, compared to the ECDF of the 3 calibration observations. The histograms showing the epistemic uncertainty in Kolmogorov-Smirnov test statistic and the p-value of the test are also shown in Figure 4.15. Again there is not evidence against the model. It is noteworthy that the goodness-of-fit test here is assessing statistical evidence for a difference between the predicted distribution from the model and the observed data with only 3 observations. The test has very low power to find a deviation due to the limited validation and future collected data would be valuable for a more stringent assessment of model validity. It is important to remember that the result of the test is not that the model is valid, but that there isn't any evidence that it is not valid.

4.5 Summary and Discussion

In this chapter, we propose a new approach to testing computer simulator validity in predicting the distribution of observations of a QoI. The method provides both qualitative and quantitative information regarding the ability of the simulator and statistical model to reflect the distribution of the physically observed data. The proposed method uses a Bayesian statistical model to express epistemic uncertainty about model parameters and generate predicted distributions of the QoI via Monte Carlo. The procedure avoids some challenges with the p-box method for model validation and the formal test gives asymptotic control of the probability of falsely declaring a model invalid for predicting the distribution of outcomes.

The goal of the proposed method is to give evidence that the simulator is able to generate the distribution of outcomes consistent with the observed distribution in field observations. This is only one aspect of testing whether a model is valid. The process of testing validity in different areas to probe the quality of predictions is a critical aspect of model building (Gelman and Shalizi, 2013). The cycle of building the model, using validation testing to find areas in which the model does not adequately represent reality, and improving the model is the core of computational science (Oberkampf and Barone, 2006; Gelman and Shalizi,

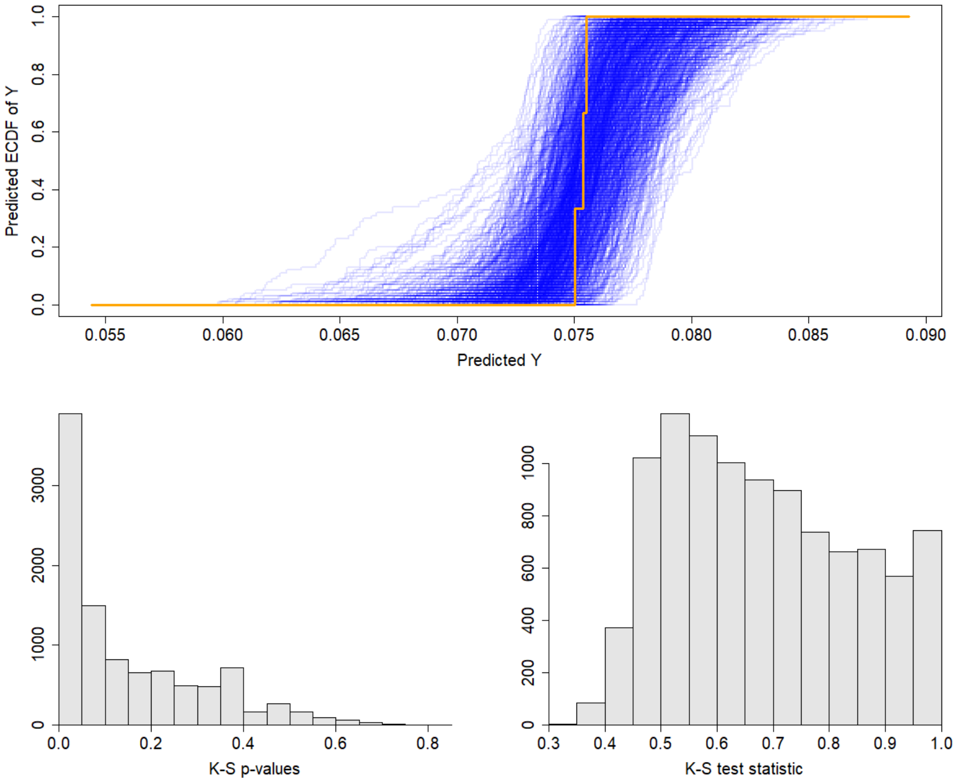


Figure 4.15: **Top:** Comparison of ensemble of model CDFs (blue) to the ECDF of a validation experiment (orange) **Left:** Histogram of p-values from the above ensemble using a Kolmogorov-Smirnov test **Right:** Histogram of test statistic values from the above ensemble using a Kolmogorov-Smirnov test

2013). The proposed method is not intended as a rubber stamp to indicate that a model is the true representation of the underlying system, rather it gives evidence that the model is capable of representing the true observed distribution.

In Section 4.2.2, the maximum p-value in the credible interval is found via Monte Carlo sampling from the posterior distribution of the calibration parameters. This is a form of optimization by random search of the high probability region. In Section 4.2.2, the control of the false positive rate relies on finding the maximum p-value in the interval. This can be difficult because the test statistics and p-value do not necessarily have derivatives - the model CDF and ECDF of the validation data are both step functions - and so cannot be found by derivative-based optimization. Use of derivative-free optimization (Conn *et al.*, 2009; Rios and Sahinidis, 2013) is a potential direction for future work to have greater assurance of finding the maximum p-value. Another potential avenue for future work is in implementation of other goodness-of-fit measures, for instance Kolmogorov-Smirnov for multiple, correlated outputs (Li *et al.*, 2014).

In Equations (4.3) and (4.4), we considered the case with only stochastically varying physical inputs $\mathbf{X}_{c,f}$ and no known inputs $\mathbf{x}_{c,f}$. If we have both $\mathbf{X}_{c,f}$ and $\mathbf{x}_{c,f}$, the predictive distribution then depends on the value of $\mathbf{x}_{c,f}$, $\tilde{F}_{c,f}(y \mid \boldsymbol{\theta}, \mathbf{x}_{c,f})$. Individual goodness-of-fit tests could be done for each value of $\mathbf{x}_{c,f}$, but if the observations are conditionally independent given $\mathbf{x}_c$ and $\boldsymbol{\theta}$, the u-pooling method of Ferson *et al.* (2008) can be used to test the distributions at once. Under the hypothesis that $y_{v,f}$ have distribution $\tilde{F}(Y_{c,f} \mid \mathbf{x}_{c,f})$, applying the probability integral transform $u_i = \tilde{F}_{v,f}(y_i \mid \mathbf{x}_{c,f})$ results in independent uniform random variates, $u_i \sim$ Uniform[0,1] (Ferson *et al.*, 2008). We can use this u-pooling approach when generating CDFs from the emulator for the proposed methodology. For each sample from the posterior distribution in Algorithm 2, a goodness-of-fit test can then be applied testing for uniformity of the u-pooled observations.

Chapter 5

Temporal Alignment of Replicate Time Series and Particle Filtering For Intrusion Detection

5.1 Introduction and Background

The detection of anomalies in time series data is of interest in a wide range of applications. For cybersecurity, the fast identification of an intrusion into a computer network may be critical to secure operation of the system. Entire industries are dedicated to developing products for intrusion detection in different contexts (Jackson and others, 1999).

Major engineered systems play an important role in national infrastructure in the United States and Canada. Examples include the power grid, waste water treatment, and pipeline operations. These facilities have control systems that manage their operation and ensure that the processes stay in safe operating regimes. Modern systems often have networked computers to communicate and coordinate operation coupled to the physical infrastructure. These networks have the potential to be vulnerable to intrusion and attack by outside actors.

Because control systems are designed to maintain stable operation for a system, understanding the typical behavior of measurements is important for identification of deviations that may indicate a security event. Doing so requires methodology that can estimate very diverse characteristic patterns that can be used to quickly identify unexpected behavior.

This chapter presents a new approach that combines a time series alignment model, particle filtering, and on-line quality monitoring to identify anomalous behavior. The goal of the methodology is to (1) estimate the characteristic behavior of the time series taking into account the varied timing of structure in the series, (2) filter new data as it arrives, and (3) use the filtering results to identify anomalies. We present an approach that can be applied to time series with diverse characteristic shapes, including those that might arise from communication/control networks. This approach is specifically intended to capture variation in overall scale and shifts in the temporal location of features in these characteristic shapes.

The proposed approach builds on the continuous profile model (CPM) developed in Listgarten *et al.* (2004). The CPM takes multiple time series replicates as input and uses a hidden Markov model (HMM) framework to temporally align them to a shared latent curve. We make several significant modifications to the CPM to accommodate anomaly detection in the sorts of data described in the application in Section 5.6.5. These modifications allow us to use particle filtering to obtain estimates of the current hidden state with uncertainty for on-line data collection. The modifications include incorporating periodicity in the latent trace as well as a number of important choices in setting the prior parameters of the HMM. To make the model practical for our large data sets, the implementation takes advantage of sparsity in the model, providing substantial computational gains over the implementation in Listgarten *et al.* (2004).

Other approaches for temporal alignment of time series have been developed. For functional data analysis, methods for alignment of curves to a common shared function are called *registration* (Ramsay, 2006). Landmark registration requires a set of features on the series be defined such that these features will be aligned and the timing of intermediate observations are warped to align between the features (Kneip and Gasser, 1992). The main drawback of this approach is that the features must occur in each observed series and ambiguity over the location of a feature can lead to poor alignment (Ramsay, 2006). For the series of interest in the applications considered here, a prior determination of key features and assurance of their existence across series was not considered feasible.

In shift registration, the observation times for each series are linearly transformed to align the curves to a common function. Gaffney and Smyth (2005) propose a two-stage procedure for clustering series based on their behavior and aligning each cluster to a common function. Liu and Yang (2009) propose a method for simultaneous alignment and clustering using a linear transformation of the time in each series. The linear transformation of time limits the flexibility of the model for alignment. The proposed method using the CPM allows the alignment to shift and scale time differently throughout a series. More flexible methods for shift registration include dynamic time warping, in which a time series is aligned to a template series to minimize the time-normalized distance (Sakoe and Chiba, 1978). This series may be another time series or an idealized prototype curve. Kneip *et al.* (2000) propose a method for curve registration to a given template via local regression. For our application data, the latent behavior of the system needs to be estimated from observations, as for most series, a prototypical example is not available. Finally, Ramsay and Li (1998) propose a method for curve registration with smooth, monotone transformations. This method allows flexible temporal warping for series alignment by iteratively estimating the shared, latent function, then using B-splines to estimate the time-warping function for each series to the latent function. While the model allows flexible temporal alignment, it can provide poor estimates when there is local structural variation in the signal, as discussed in Section 4.2 of Ramsay and Li (1998). The CPM model is able to both capture local structural variation as well as the flexible temporal alignment.

The work in this chapter is motivated by anomaly detection on heating, ventilation, and air conditioning (HVAC) data collected at the National Security Sciences Building (NSSB) in Los Alamos National Laboratory. For each time series of interest, the approach involves the following steps:

1. Characterize typical daily behavior with the modified version of the CPM to obtain a latent trace for a sample of time series replicates believed to be anomaly-free.

2. Align new measurements to the estimated latent trace using a particle filter with the CPM.

3. Use the filtered residuals and process control methods to detect behavior that could indicate an anomaly as we monitor the building.

Section 5.2 describes the original CPM of Listgarten *et al.* (2004) and our innovations are outlined in Section 5.3. In Section 5.4 we propose to use a particle filter with the CPM that supports efficient on-line prediction and filtered residuals while accounting for uncertainty in the predicted states. Section 5.5 introduces the methodology for intrusion detection integrated with the proposed alignment model. Section 5.6 explores the performance on synthetic and measured data. Finally, discussion of the model and future work is in Section 5.7.

5.1.1 Background on Hidden Markov Models

Hidden Markov models (HMM) are a form of state space model with a discrete set of S unobservable states, $\{\pi_1, \dots, \pi_S\}$. The HMM is defined by these states, the transition probabilities between states, and the emission probabilities that encode the way in which an observation is generated from a given state (Bishop, 2006).

Figure 5.1 shows a diagram of a simple discrete state space. There are three states and the allowed transitions are shown with arrows. For this illustrative example, let the transition probabilities be as follows: the probability of a transition from a state to itself is 0.95; for states 1 and 3, the probability of transitioning to state 2 is 0.05; from state 2, the probability of transitioning to state 1 or 3 is 0.025 each. This is called a “Markov model” because the probability of the current state only depends on the immediate previous state, not on earlier states or the value of previous observations.

The hidden aspect of the HMM comes from the fact that the states are not observed, but each state “emits” an observation according to the properties of the state. The emission probability is the likelihood of a particular value of an observation given a state from which it is generated. Suppose that the mean responses for each state are -1,0, and 1 respectively

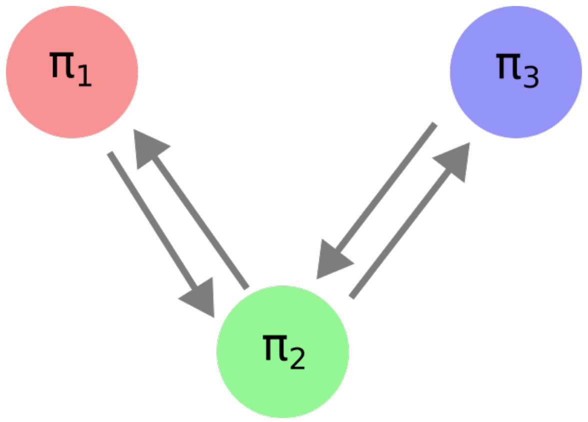


Figure 5.1: Diagram of a simple 3 latent state HMM. The states are shown as colored circles and allowable transitions are shown as arrows.

and that the observation from a state is emitted with additive Gaussian error with standard deviation 0.25, 0.5, and 0.25 respectively. Figure 5.2 shows data generated from this simple HMM. The color corresponds to the state that emitted the observation.

When one fits an HMM, the state information is unknown and must be estimated. The full likelihood for a sequence of observations from an HMM, is the product of the initial state probability, transition probabilities between states, and the emission probabilities for each observation.

Hidden Markov models have been successfully used for a wide range of applications. For example, (Jelinek, 1997) present the use of HMMs in speech recognition. Starner and Pentland (1997) utilize an HMM for real-time sign language recognition. HMMs are used widely in natural language processing (Manning *et al.*, 1999; Rabiner and Juang, 1986). Eddy (1998) provides an overview of profile hidden Markov models, used for alignment of discrete sequences for genetic and protein sequencing applications.

5.2 Hidden Markov Model For Characterizing Typical Behavior

Listgarten *et al.* (2004) developed the CPM to do temporal and scale alignment of replicate time series. Their primary goal was to estimate a latent curve that could be viewed as "a single, superresolution fusion of the data" to gain insight into the system (Listgarten *et al.*, 2004). In contrast, our goal is to use the CPM to estimate the persistent behavior of the system to facilitate detection of outliers that would indicate anomalous system behavior. The different observed time series share the same features, but the timing of these features may be varied from one observed series to the next. The estimation of the shared behavior is done by aligning time series from the system to the estimated typical behavior using the hidden Markov model and then using the estimated model with a particle filter to obtain

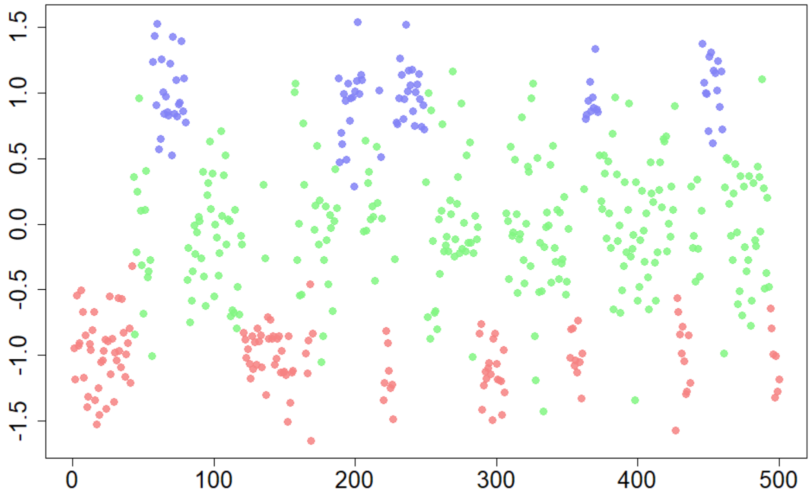


Figure 5.2: Data generated by the simple HMM in Figure 5.1. A sequence of 500 observations are shown, with the color corresponding to the state.

on-line residuals. The residuals are then used with a process control method to detect indications of intrusion.

5.2.1 The Continuous Profile Model

This section introduces the CPM as laid out in Listgarten *et al.* (2004). Suppose we have K exchangeable time series, $\mathbf{x}_k = (x_{1,k}, x_{2,k}, \ldots, x_{N_k,k})$ for $k = 1, \ldots, K$. The goal of the CPM is estimate the shared signal of the K time series, allowing the timing of this behavior to change between series and accounting for local variation, as well as differences in scale between series. Let the function $z(t)$ be the unobserved latent trace representing the shared behavior in the collection of observed time series. In the CPM, this function will be observed at a discrete number of latent times.

Let $\pi^{(m,q)} = \{\tau^{(m)}, \phi^{(q)}\}$, be a hidden Markov state, where $\tau^{(m)}$ is the latent time state and $\phi^{(q)}$ is a multiplicative scaling factor state (here we have overloaded the notation ϕ to describe both the state and the value of the scale factor applied to $z(t)$ when in state ϕ). A time state $\tau^{(m)}$ has a corresponding value of the QoI, the value of the function for that time state, $z(\tau^{(m)})$. The scaling state $\phi^{(q)}$ shifts the value for the curve in state $\pi^{(m,q)}$ to be $z(\tau^{(m)})\phi^{(q)}$. The local scale changes attempt to capture short term systematic deviations from the characteristic behavior. In total there are Q scaling states, $\phi^{(1)}, \ldots, \phi^{(Q)}$ and M latent time states, $\{\tau^{(1)}, \ldots, \tau^{(M)}\}$ leading to $M \times Q$ total hidden Markov states in our model specification. The M latent times are considered equally spaced within their range.

Each time series, indexed by k, also has a global scaling factor, u_k, which shifts the entire series relative to the unobserved function $z(t)$. With this scaling factor, the emission probability can be defined for a given latent state. For observation $X_{i,k}$, indexed by i in time series k, given the state $\pi_{i,k}$ and global scaling factor for the series u_k, the expected value of the scaled latent curve is $z(\tau_{i,k})\phi_{i,k}u_k$. An observation "emitted" from this state has additive Gaussian noise with mean 0 and variance σ_x^2 so that the model for an observation:

$$\begin{aligned} X_{i,k} &= z(\tau_{i,k})\phi_{i,k}u_k + e_{i,k} \text{ and} \\ e_{i,k} &\sim \mathsf{N}(0, \sigma_x^2). \end{aligned} \tag{5.1}$$

To specify the full hidden Markov model, we additionally need the transition probabilities. Let T_{π_{i-1},π_i} be the probability of transitioning from state π_i to π_{i+1}. Here i is indexing the observation in a series, though to clarify, the transition probabilities only depend on the current and target hidden Markov state, not on the index for a given current and target state pair. The notation with i is used here to connect to the particle filtering in later sections. The scale and time states are treated independently, therefore the transition probabilities are factored,

$$T_{\pi_{i-1},\pi_i} \equiv P(\pi_i \mid \pi_{i-1}) = P(\phi_i \mid \phi_{i-1})P(\tau_i \mid \tau_{i-1}) \equiv T_{\phi_{i-1},\phi_i}T_{\tau_{i-1},\tau_i}. \tag{5.2}$$

While the transition probabilities T_{ϕ_{i-1},ϕ_i} and T_{τ_{i-1},τ_i} for the scale and time states can be estimated from observed data, Listgarten (2007) found that the improvement in quality of fit on both test and training data was small when learning the transition probabilities and that the computational cost was a factor of 8 larger. Instead it was recommended to select the transition probabilities *a priori.*

Listgarten *et al.* (2004) impose several constraints on the possible state transitions to achieve both realism and tractability. The transitions between scale states are constrained to occur only between neighboring scales $\phi^{(q)}$ instead of allowing arbitrary scale changes between observations $x_{i,k}$. Similarly only jumps in time of up to J time states are allowed by the model and only in the forward time direction. Transitions from a time state τ_j to itself are given probability zero. In the examples we use $J = 3$. These constraints make the transition matrix extremely sparse, which gives computational benefits both for fitting the model, as described in Section 5.3, and for filtering with on-line collection of data in Section 5.4.

Listgarten *et al.* (2004) and Listgarten (2007) use uniform transition probabilities for both scale and temporal states. While we follow the uniform recommendation for the scale state transition probabilities T_{ϕ_{i-1},ϕ_i}, Section 5.3 introduces a different treatment of the time state transition probabilities T_{k,τ_{i-1},τ_i} which improved performance with on-line filtering.

The complete log-likelihood for the CPM is:

$$\ell(\{\pi_{1,1},\ldots,\pi_{N_K,K}\},\mathbf{z},\boldsymbol{\phi},\{u_1,\ldots,u_k\},\sigma_x^2) =$$
$$\sum_{k=1}^{K}\left(\log(\pi_{1,k}) + \sum_{i=2}^{N_k} T_{\pi_{i-1,k},\pi_{i,k}} - \frac{1}{2}N_k\log(\sigma_x^2) - \frac{1}{2}\sum_{i=1}^{N_k}(x_{i,k} - z(\tau_{i,k})u_k\phi_{i,k})^2/\sigma^2\right), \tag{5.3}$$

where the terms in the outermost parentheses are the log probability of the initial state of series k, the transition probability between states for series k, and the sum over the logarithm of the emission probability for observations in series k, respectively.

Generating observations from this model is done by following these steps:

1. An initial observation for the series is "emitted" from some state $\pi^{(m_1,q_1)}$ with a global scaling factor u.

2. The system then transitions to a new state along the latent curve with a multiplicative scaling according to the transition probabilities of the hidden Markov model.

3. The cycle of emission followed by transition continues until the end of the day is reached.

Each transition moves through states corresponding to different times along this curve, with the size of the steps through latent time states controlling the timing of features in $z(t)$.

Figure 5.3 illustrates the steps for the alignment model on a single series from the synthetic example. Looking at panel (a) we see how the observed time series (red) is aligned to the latent trace (blue). In panel (b), the aligned points in blue have been shifted downward by the global scaling parameter, u. In panel (c), the blue points are again shifted to show the effect of the local scaling, ϕ. For instance, the blue points for the first 8 observations of the day have been shifted vertically relative to the rest of the aligned curve to account for the extra separation between the early and later observations in the series. Lastly in panel (d) the effect of time warping is shown, as the blue values have been shifted from their latent time τ in the previous panels to the observed time with the observations. The remaining separation between the blue and red points is the residual error.

Section 5.2.2 discusses choosing the number of states. Listgarten *et al.* (2004) also impose smoothness on the latent trace $\mathbf{z}$. We discuss how Listgarten *et al.* (2004) handled each of these below and Section 5.3 describes proposed innovations to the model for use with time series for anomaly detection.

5.2.2 Number of Latent Time States

In specification of the model, the number of latent states must be chosen. The time states amount to a horizontal shift (a shift in time) for each series to the latent trace. For the time warping/alignment to work, the latent time states are required to have finer time grid

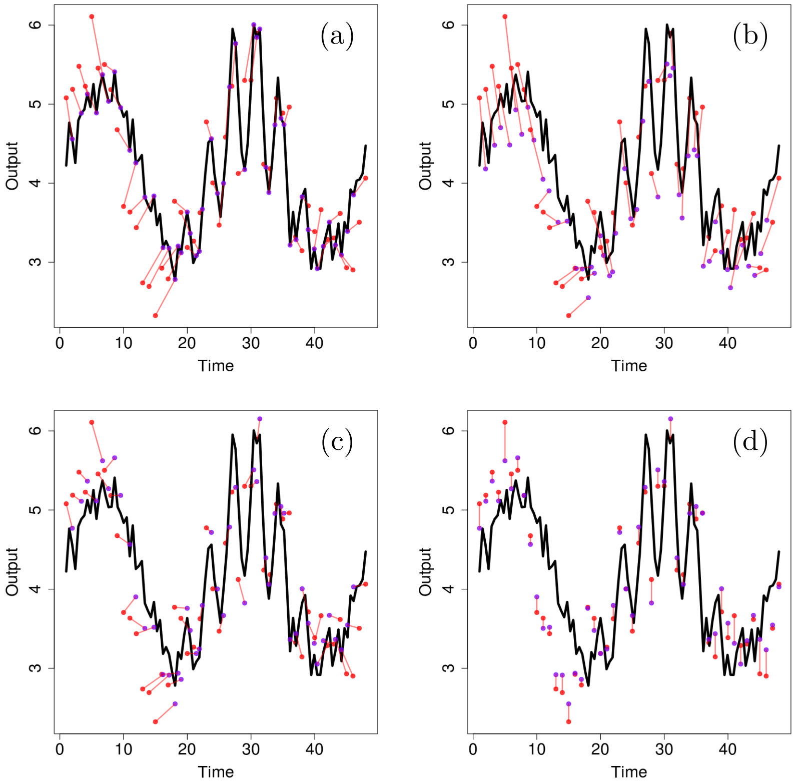


Figure 5.3: Steps of the continuous profile model. The latent curve is shown in black in each panel. Panel (a) shows the observed series (t_i, X_i) in red and the locations on the latent trace that they align $(\tau_i, z(\tau_i))$ in blue. In (b), the blue points include the global scaling, $(\tau_i, z(\tau_i)u_k)$. Panel (c) includes the local scaling factors in the blue points, $(\tau_i, z(\tau_i)u_k\phi_i)$, shifting some of the data closer to the scale of the observations. Finally, in (d), the time warping is removed, such that the blue points have been moved to their observed time, $(t_i, z(\tau_i)u_k\phi_i)$. The remaining deviations between observed points and latent points is due to irreducible noise.

than any of the measured time series ($M \gg \max N_k$). To see this, consider having the same number of latent time states as observations, $M = N_k$. Then each observation could have precisely one potential time state in a period. If an observation x_m instead was best aligned to another, later time, then observation x_M would no longer have an open state to be aligned to in the implementation of Listgarten *et al.* (2004). Instead, if $M = N_k$, their implementation would simply only allow the one-to-one matching of states and no time warping. The development of periodic edge transitions that will be discussed in Section 5.3 would allow the last observation to be aligned to the beginning of the latent curve. However, it is still undesirable for the states to only be able to warp forward in that case.

The goal of the alignment is that the observed series can have the timing shifted to align common behavior across multiple series. This goal requires states to be both compressed in time (moved more closely together) and expanded. Figure 5.4 attempts to illustrate this with a simplified diagram. The observations in blue are aligned to the latent time states, τ, in red. The upper panel shows the states aligned with equal spacing in time. The bottom panel shows that, if an observation is aligned to a later time state, it must push the last state or states off the end of the period.

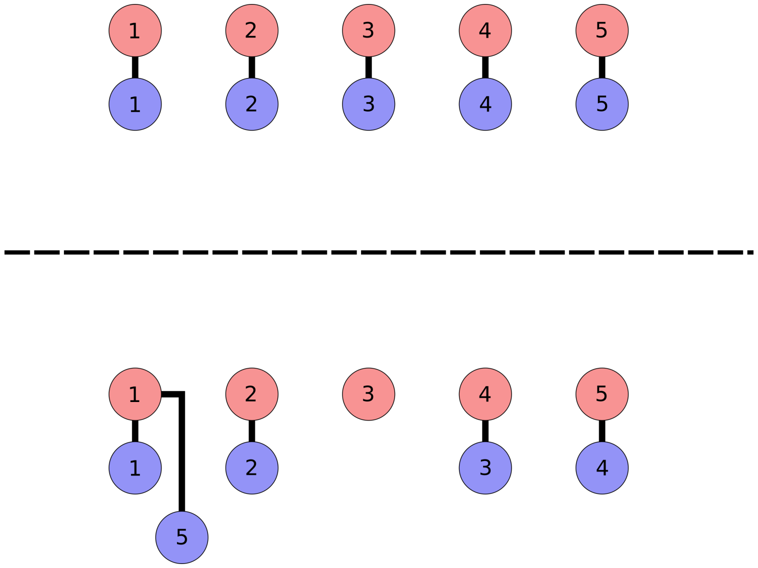


Figure 5.4: Illustration of the issue with having only $M = N_k$ latent time states. In the top panel, each observation (blue) is mapped to a time state with no gaps. The bottom panel shows that to allow observation 3 to be aligned to a later time (state 4), the remaining observations are all pushed back such that observation 5 must be pushed past the end of the “day” to the beginning.

By having a finer grid of latent time states, the transitions through the latent curve can move more slowly or more quickly as the indicated by the features in the data. Figure 5.5 shows a similar illustration to Figure 5.4, but where the number of latent times is larger than the number of observations in a series. The top panel is again equally spaced in time, but now there are open latent states. The utility of those states is clear in the lower panel, which shows how the observations can be aligned with unequal spacing through the latent states, but still keep the full series on a single “day”. Even with the periodic implementation, it is still not desirable to force the later observations past the end of the series. No single series of N_k observations will enter all the latent time states, but will transition through in order to align the observed behavior to the characteristic behavior.

For example, Figure 5.6 shows the alignment of a series from the synthetic example in Section 5.6.2. The observations for times between 10 and 15 clearly correspond to the fast drop in the latent curve and are aligned forward in latent time as such. If the number of latent states was equal to the number of observations, each subsequent observation would need to be aligned at least that far ahead. With approximately 2 latent states for each

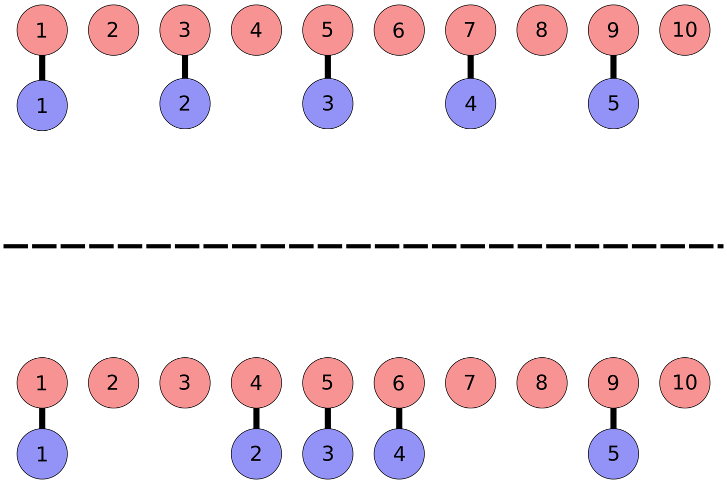


Figure 5.5: Illustration of the benefit of the finer grid of latent times, with $M = 2N_k$. In the top panel, each observation (blue) is mapped to a time state and the set of observations are equally spaced through the “day”, as in the top panel of Figure 5.4. Now there are open states however. The bottom panel shows the observations aligned such that their spacing is no longer even. The additional states allow the observations to be aligned to different parts of the latent trace without forcing an observation off the end of the “day”.

potential observed state, the later observations are not forced ahead, but can step more slowly through latent times to stay aligned to the curve.

Following Listgarten *et al.* (2004), we found that using the number of time states to be slightly more that two times the length of the longest observed times series to perform adequately. That is $M = (2 + \epsilon) \max N_k$, where ϵ is a small additional buffer of states. In our implementation we choose $\epsilon = 0.1$ but found the model insensitive to variations in the range $0.01 \leq \epsilon \leq 0.1$. As the intent of ϵ is to be a small buffer of additional states beyond a strict multiple, larger values for the additional buffer we not considered.

The choice of the number of latent scaling states Q is a balance between computational cost and the need for enough resolution to capture small local variation from the latent curve. The total number of hidden Markov states is $Q * M$ with M large due to the number of states needed for the alignment in time. As will be discussed in Section 5.3.3, the large state space is a significant computational burden, so the number of scaling states Q must remain small for computational feasibility. The trade-off for using a small Q is that the local variation away from the latent curve will be modeled with low resolution, which can lead to structure in the residuals. For the examples we used $Q = 5$, which performed well empirically.

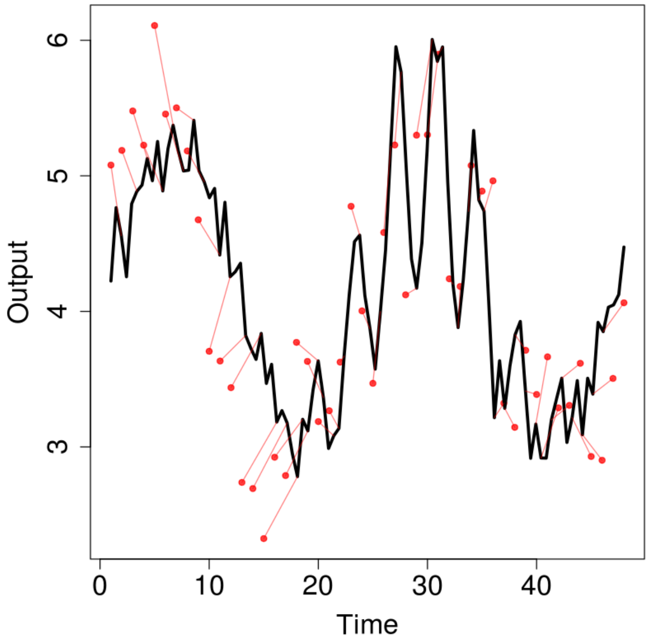


Figure 5.6: Example of alignment of observations to a latent curve. The red points show observed values of the time series and the estimated locations on the latent trace are shown with the line from the point to the thick, black curve.

5.2.3 Smoothness of the Latent Trace

The complete log-likelihood for the model was stated in Equation 5.3. This likelihood does not explicitly restrict the smoothness of the trace and therefore can result in substantial complexity in potential latent curves. To encourage smoothness in the estimated latent trace $\mathbf{z}$, Listgarten (2007) introduced a penalty to the log-likelihood on differences between adjacent values in the latent trace, $-\lambda\tilde{u}\sum_{j=1}^{\tau-1}(z(\tau_{j+1}) - z(\tau_j))^2$, where $\tilde{u} = \overline{u^2} = \frac{1}{N}\sum\limits_{k=1}^{K} u_k^2$.

This penalty is equivalent to assuming a conditional $\mathsf{N}(z_i, \frac{1}{2\lambda\tilde{u}})$ prior distribution on z_{i+1} in a Bayesian inference setting, where λ is a free parameter which can be chosen using a validation set, cross validation, or prior information. It is worth noting that the penalty is not scale-invariant, so reasonable prior knowledge may be difficult to obtain.

The complete log-likelihood with the penalty is:

$$\begin{aligned}&\ell(\{\pi_{1,1}, \ldots, \pi_{N_K,K}\}, \mathbf{z}, \boldsymbol{\phi}, \{u_1, \ldots, u_k\}, \sigma_x^2, \lambda) = \\ &\sum_{k=1}^{K}\left(\log(\pi_{1,k}) + \sum_{i=2}^{N_k} T_{\pi_{i-1,k},\pi_{i,k}} - \frac{1}{2}N_k\log(\sigma_x^2) - \frac{1}{2}\sum_{i=1}^{N_k}(x_{i,k} - z(\tau_{i,k})u_k\phi_{i,k})^2/\sigma_x^2\right) \\ &- \lambda\tilde{u}\sum_{j=1}^{\tau-1}(z(\tau_{j+1}) - z(\tau_j))^2.\end{aligned} \tag{5.4}$$

The term $\tilde{u}$ is included in Equation (5.4) to ensure the smoothness penalty λ is effective at encouraging smooth solutions (Listgarten, 2007). Without $\tilde{u}$, the penalty term in Equation (5.4) is $-\lambda \sum_{j=1}^{\tau-1}(z(\tau_{j+1}) - z(\tau_j))^2$. Because the mean for an observation is $z(\tau_{i,k})\phi_{i,k}u_k$, dividing $z(\tau_{i,k})$ by a constant B and multiplying u_k by B results in the same likelihood. The penalty term is $-\frac{\lambda}{B}\sum_{j=1}^{\tau-1}(z(\tau_{j+1}) - z(\tau_j))^2$, which can be made arbitrarily small by increasing B without having any effect on the quality of fit of the model. Including the term $\tilde{u}$ ensures that smooth solutions can be found by proportionately scaling up the penalty if that constant is increased.

5.2.4 Estimation

As is typical for HMMs, Listgarten *et al.* (2004) use the expectation-maximization (EM) algorithm to estimate the model parameters, $\mathbf{z}$, u_k, and σ_x^2 (Dempster *et al.*, 1977). For the CPM, the expectation is taken with respect to the latent states. The expected, penalized likelihood is:

$$\begin{aligned} E[\ell(\mathbf{z}, \boldsymbol{\phi}, \{u_1, \ldots, u_k\}, \sigma_x^2, \lambda)]_\pi = \\ -\frac{1}{2}\sum_{k=1}^{K}\sum_{m=1}^{M}\sum_{q=1}^{Q}\sum_{i=1}^{N_k} p_{m,q,k,i}\left(N_k \log(\sigma_x^2) + (x_{i,k} - z(\tau_m)u_k\phi_q)^2/\sigma^2\right) \\ - \lambda\tilde{u}\sum_{j=1}^{\tau-1}(z(\tau_{j+1}) - z(\tau_j))^2 + constant, \end{aligned} \tag{5.5}$$

where $p_{m,q,k,i}$ is the probability that observation i is in time state m and scale state q for time series k. Terms that do not depend on the estimated parameters have been combined into the constant.

The algorithm iterates between computing the complete state probabilities for each time step using the Baum-Welch algorithm (Baum *et al.*, 1970) and then maximizing the parameters over Equation 5.5. Expressions for the maximum penalized likelihood values of $\mathbf{z}$, u_k, and σ_x^2 are given in Listgarten (2007). In the implementation by Listgarten *et al.* (2004), ϕ are fixed.

5.3 New methodology for Anomaly Detection in Time Series

In our work we are interested in using the hidden Markov model approach to align time series from control systems to a common latent behavior. The idea behind the proposed approach can be used to address the varied timing of features in the time series from day-to-day and to use the HMM structure for efficient filtering with some novel innovation. In this section, the new methodology is described to make the continuous profile model both computationally practical and scientifically appropriate for the application as well as the on-line data acquisition environment.

5.3.1 Estimating the Value of Scaling States

In Listgarten *et al.* (2004), the values for $\phi^{(1)}, \ldots, \phi^{(Q)}$ were fixed. In the application, there is rarely prior information that can be used to set these values and it is instead preferable to estimate these to approximate the appropriate scale to capture small local variation in the time series. This is important for our detection context to allow the model to learn the typical scale of local variation and not atypical variation due to anomalous behavior. We adapt the model to allow these values to be estimated during the maximization step of the EM algorithm. Allowing the scale states to be estimated creates a confounding between the values of the scaling states $\boldsymbol{\phi}$ and the global scaling u. Multiplying all the scaling states by a constant B and dividing u by the same constant results in the same penalized likelihood. To avoid this confounding, the smallest, $\phi^{(1)}$, is fixed to be 1 and other values are restricted to be > 1. An equivalent restriction would be to fix the largest,$\phi^{(Q)}$, to be 1 and the remaining to be less than 1. Under this specification, there is a closed form solution for the scaling state $\phi^{(j)}$:

$$\hat{\phi}^{(j)} = \frac{\sum\limits_{k=1}^{K} \sum\limits_{\{s:\phi_s=\phi_j\}} \sum\limits_{i=1}^{N_k} p_{s,k,i} z_s u_k x_{i,k}}{\sum\limits_{k=1}^{K} \sum\limits_{\{s:\phi_s=\phi_j\}} \sum\limits_{i=1}^{N_k} p_{s,k,i} (z_s u_k)^2},$$

with the other parameters fixed at their current value in the EM algorithm. In the above, $p_{s,k,i}$ is the probability that observation i in series k is in state s and sum over s is over all states s with scaling state of $\phi^{(j)}$. The derivation for this is given in the Appendix for this chapter.

5.3.2 Periodic Latent Trace

For the application that motivated this work, the time series display diurnal behavior, therefore the proposed approach specifies periodic boundaries. This allows late latent time states to transition to early ones. If periodic transitions are not allowed, a new replicate series could reach the end of the latent trace before the end of the day's observations, leaving the model with no time states that have non-zero probability for the remaining observations collected that day.

This requires two innovations. First, the Markov transition matrix must include transitions from late time states to early ones by allowing transitions from state τ_j to $\{\tau_{(j+1 \mod M)}, \ldots, \tau_{(j+J \mod M)}\}$. Second, to impose periodicity and smoothness on the latent curve, the penalty λ from Equation 5.4 must also include a term with z_1 and z_M:

$-\lambda\tilde{u}\left((z_M - z_1)^2 + \sum_{j=1}^{\tau-1}(z_{j+1} - z_j)^2\right)$. This is critical for handling the residuals in an online manner with the particle filter discussed in Section 5.4.

5.3.3 Computation for Large State Spaces

In the specified model, a practical, computational concern is the size of the state space. Because the latent time and scale states are independent in the HMM, the total number of states is $Q \times M$. In the application in Section 5.6.5, we set $Q = 5$ and $M = (2+\epsilon)\max(N_k)$ with $\epsilon = 0.1$. A typical size for the maximum number, $\max(N_k)$, is 1440 (the number of minutes in a day), so the state space has around $5 \times 2.1 \times 1440 = 15,120$ states. It turns out that this is fairly large and some issues must be addressed.

The aforementioned constraints placed on transitions between the scale states and the time states make the transition matrix quite sparse. Recall that the scale transitions are constrained to occur only between adjacent scales, so that there are only three possibilities for a scale transition: move to the next smaller scale, stay at the current scale, or move to the next larger scale. Similarly the time transitions are constrained to be no more than J steps away. Because the scaling transition can remain in the same scaling state or move to one of two potential adjacent states, from a particular state $\pi_{i,k}$ a transition can only reach at most $3J$ time states in the next step.

In Listgarten *et al.* (2004), the authors use the full transition matrix of size $QM \times QM$. We instead take advantage of the imposed sparsity by working with a set of three vectors of length $3JQM$. One vector contains the indices of the beginning state of the transition, another has the indices of the target state, while the third contains the transition probabilities used by the Baum-Welch algorithm. (In practice we store the two bookkeeping vectors of indices in a single matrix of size $3JQM \times 2$.) Indeed these vectors could be made even shorter because not all states can transition to a full set of $3J$ states. The advantage is likely negligible compared to the savings already achieved over the $QM \times QM$ implementation. For the application, with $J = 3$ the transition matrix contains $15,120 * 9 \ll 15,120^2$ entries.

To make the computational gains more concrete, in the application with time series of length 1440 in Section 5.6.6, fitting the model using the full transition $15,120 \times 15,120$ transition matrix was not feasible due to memory constraints. Downsampling the data by one-tenth, so that each series contained 144 observations allowed the model to be fit in approximately 280,000 seconds on 12 cores. Using the sparse matrix approach from the previous paragraph on the full 1440 observations, each run of the model takes approximately 25,000 seconds on 12 cores and used approximately 18 GB of RAM. These substantial improvements made fitting the model to the full data feasible, though still computationally expensive. However, the cost of aligning new replicate series to an estimated model is much lower as discussed in Section 5.4.

5.3.4 Choice of Fixed Transition Probabilities

Listgarten *et al.* (2004) suggested uniform transition probabilities over reachable states for the HMM. We found for both estimating the parameters of the CPM and, as will be discussed, for filtering, using non-uniform probability improved performance. We note that choosing $M = (2 + \epsilon) \max N_k$, that is choosing to have roughly twice as many latent time states as observations in a series, implies a preferred time jump of about 2 time steps: the latent curve is estimating the daily behavior, so modulo the time warping, a single replicate time series should reach from approximately state $m = 1$ to state $m = M$ in the N_k observations. To do so, each observation needs to transition across approximately 2 time states on average. We found benefit for $J = 3$ in setting $P\{\tau_i = j + 2|\tau_{i-1} = j\} = 0.5$ and $P\{\tau_i = j + 1|\tau_{i-1} = j\} = P\{\tau_i = j + 3|\tau_{i-1} = j\} = 0.25$, which favors a time jump of size 2 as desired but still allows time to move slightly faster and slower if needed. When filtering as in Section 5.4, this implementation alleviates the problem of getting stuck in a state that is too far ahead or behind.

5.3.5 Initialization of Latent Trace for the Parameter Estimation

Parameter estimation for the HMM using the EM algorithm requires a set of parameters used to initialize the algorithm (Dempster *et al.*, 1977).

Empirically, we found the solution obtained was sensitive to the initialization of the values of $z(\tau_m)$, indicating there exist many local maxima of the penalized likelihood. Although Listgarten *et al.* (2004) suggest using one of the replicates to initialize the latent trace, empirically we found better recovery of the temporal alignment when using the values averaged over the replicate series as follows:

1. For $i = 1, \ldots, \max(N_k)$ calculate $\bar{x}_i$, as the average QoI for observed time series at time index i in the K series.

2. For any i that has no observed values at its index, set the $\bar{x}_i = \bar{x}_{i-1}$, starting with the smallest i.

3. Upsample from the $\max(N_k)$ values of $\bar{x}_i$ to the M values $z(\tau_m)$ by equally spacing the values of $\bar{x}_i$ in time and filling adjacent states

For step 2, if for some reason no series has an observation at $i = 1$, then use larger adjacent indices first. For synthetic data in Section 5.6.2, improved recovery of the known latent curve was found.

5.4 Particle Filtering for State Estimation With On-line Data

To use the proposed CPM to detect intrusions a new time series, the model is estimated to characterize typical daily behavior for a collection of (we hope) intrusion-free replicates as demonstrated above. We can then apply the estimated model, including the estimated latent trace $z(\tau)$, error variance of the data σ_x^2, and scaling states $\phi^{(1)}, \ldots, \phi^{(Q)}$, to measurements as they arrive in real-time. In particular, the model can be used for filtering, estimating the current latent state $\pi_{t,k}$ given all data up to the latest observation. This will allow for examination of filtered residuals to detect intrusions as discussed in Section 5.5.

The estimated HMM is a probabilistic generative model for the unobserved states. We can use Bayesian inference to estimate the latent states for the sequential observations as they arrive, given the fixed, estimated CPM. This approach can be seen as a form of modularization (Liu *et al.*, 2009), in which a Bayesian model has a submodel with parameters that are estimated by optimization, while sampling methods are used to estimate the remaining parameters. For the CPM, the maximum penalized likelihood estimates using Equation 5.4 can be viewed as a maximum *a posteriori* (MAP) estimate with improper uniform prior distributions on the global scaling, local scaling, and residual variance parameters and the conditional prior described in Section 5.2.3 for, $z(\tau^{(1,1)}), \ldots, z(\tau^{(M,Q)})$, the values of the latent trace. Because of the computational expense of fitting the CPM, modularization is necessary; however, by fixing model parameters at their MAP estimates, the modularized model does not account for uncertainty in the latent trace. Using the particle filter to sample latent states during on-line alignment of new time series allows uncertainty in the latent state to be accounted for in intrusion detection.

For a new series indexed by g, the prior distribution for the latent state of the initial observation is $p(\pi_{1,g})$. When observing the first value in a time series, the prior distribution can be updated to obtain a posterior distribution for the initial state $p(\pi_{1,g} \mid x_{1,g})$. The distribution for sequence of hidden Markov states up to the second observation after the first observation is then:

$$p(\pi_{1:2,g} \mid x_{1,g}) = p(\pi_{1,g} \mid x_{1,g}) T_{\pi_{1,g},\pi_{2,g}}.$$

This distribution can be updated by conditioning on the second observation and the sequence transitioned forward another step. The posterior distribution can continue to be updated as each new observation arrives in the on-line fashion.

We propose a particle filter approach, outlined in Algorithm 3, to estimate the posterior distribution of the current hidden state as observations arrive. The particle filter is a natural method for performing Bayesian inference with sequential models. Liu and Chen (1998) proposed particle filtering for dynamical systems and showed application to an econometrics model and to target tracking in clutter. Fearnhead and Clifford (2003) show an application of particle filtering to on-line data for an oil drilling application. We will closely follow these approaches to the particle filter.

Instead of a single estimate of the current state, the particle filter gives us n_p estimates, one for each particle. Each particle, $j = 1, \ldots, n_p$, is weighted by the probability that the observation came from that state, $w_j = P(x_i \mid u_j, \pi_{i,j})$. The weights indicate the relative probability of different potential paths through the hidden Markov state space. The diversity of the particles captures the uncertainty in paths that are consistent with the observed data. As more observations are collected, the weight can become concentrated on a few particles.

In order to ensure that the population of particles is still capturing the uncertainty in later states, the current particles can be resampled with replacement according to the weights w_j. To determine when resampling is appropriate, we follow the standard procedure in particle filtering literature (Djuric *et al.*, 2003; Doucet *et al.*, 2000). At each step the effective sample size of the weighted collection of particles is $ESS = \frac{1}{\sum w_j^2}$. If $ESS < n_{thresh}$, we then resample particles with replacement. We use $n_{thresh} = n_p/2$. Tracking multiple particles ensures that potential paths through the latent space that are of similar quality continue to be followed until the data rules them out and resampling avoids tracking particles whose weight is very small. This increases robustness in filtering for the on-line environment.

To visualize, the particle filtering procedure is illustrated through one of the examples that is discussed in Section 5.6.2. Figure 5.7 shows the latent curve and the latent curve shifted by the values for the scaling states as solid gray lines. In Figure 5.7a, the initial states are shown as red circles, with the opacity indicating the fraction of particle filtering samples in that state. The more opaque red points indicate states with a higher posterior probability for the current observation.

In Figure 5.7b, one observation from the time series has been observed. The particles, following the transition probabilities, enter new potential states and are then weighted by their relative fit to the first observation. The weight after the single observation is concentrated on a smaller number of particles than previously. This indicates more posterior certainty about the path the time series is taking through the latent trace.

Figure 5.7c, 5.7d and 5.7e show later steps after the 8, 20, and 40 observations respectively. We can see how the alignment uncertainty tracks through as data is added in a sequential, on-line fashion. Note that the observation in Figure 5.7c is warped forward in time for most particles, while for 5.7d temporal alignment of observation to latent times requires less time warping. By 5.7e, the latent time and observed time are roughly equal. Figure 5.7f shows the last observation of the day, the 48th. The uncertainty in state passes through the end boundary and back to the beginning of the latent trace.

One mentioned benefit of the HMM and the particle filtering approach, is that it can naturally handle missing observations in the on-line stream of data. To illustrate this, in

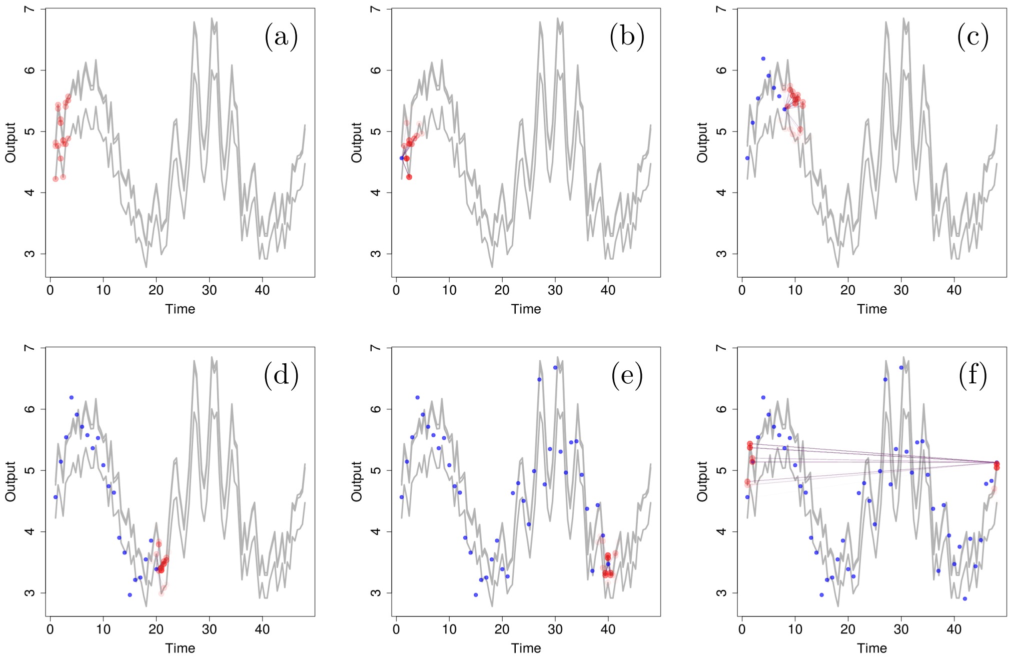


Figure 5.7: On-line alignment of particles to the latent trace at (a) the initialization, (b) the first observation, (c) the 8th, (d) the 20th, (e) the 40th, and (f) the 48th. In each image the solid gray lines indicate the latent trace scaled by each scaling state value. The observed data is shown in blue and the estimated current state of the last observation is shown in red. The opacity of the red points indicates the estimated posterior probability of each state, with more opaque indicating higher probability.

Figure 5.8, we show the the alignment using the first four observations, but then dropping the others except those at times 10, 20, 30, and 40. In 5.8a, we see the filtered alignment of the observation at a time of 20. Because of the missing intermediate observations, there is substantially more uncertainty in the latent state of this observation than there was in 5.7d. The observation at time 30 in Figure 5.8b, the particle filtering approach indicates uncertainty over which of the two peaks the observation may be aligned. Using the particle filtering approach to continue the alignment in face of that uncertainty is particularly useful here, as future observations may be informative as to which of the two peaks this series passed through. Choosing an optimal state with the forward model would force a choice of a particular state and be unable to adapt to future information. Lastly, we again see that the 40th observation in 5.8d has significant uncertainty in latent state due to the lack of information about intermediate steps and the fact that late values of the latent trace have similar values.

Because the number of latent states reachable from a particular state is small, $3J$, sampling the particles is relatively fast, even when the full state space is large. For the example in Section 5.6.7 with 15,120 total hidden Markov states with $J = 3$, where there are $3 \times 3 = 9$ states reachable from any other, updating $n_p = 500$ particles took approximately 0.14 sec-

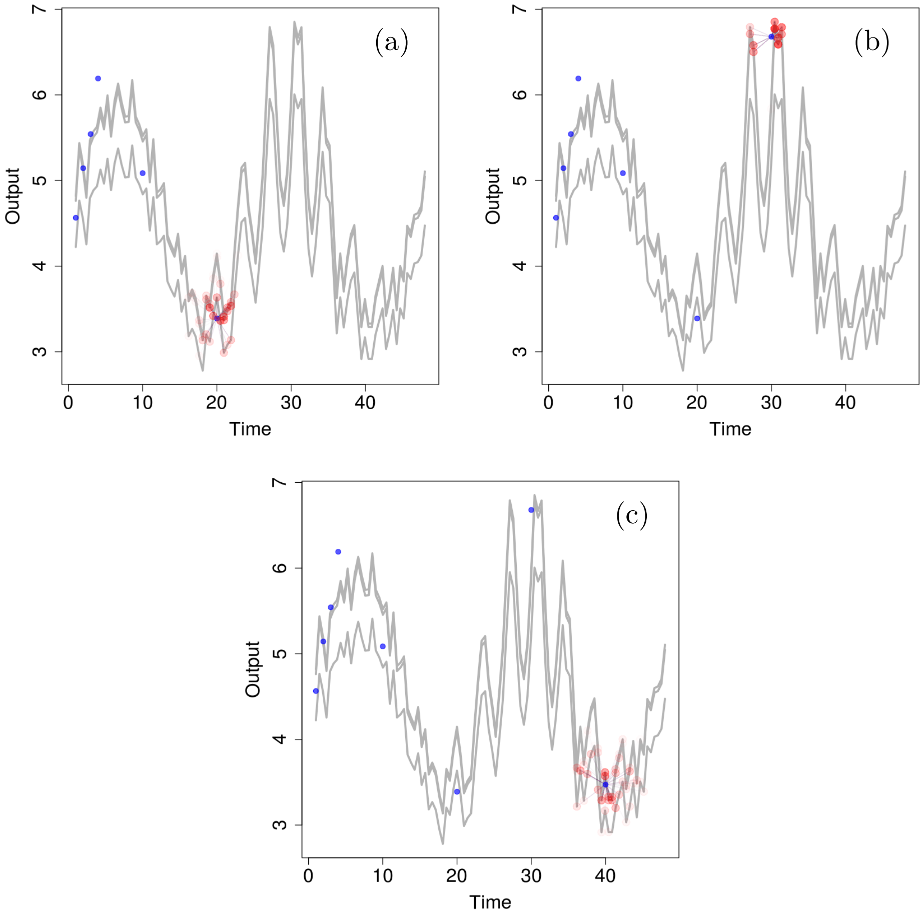


Figure 5.8: On-line alignment of particles to the latent trace missing gaps in the observed series. Observations are at (a) 20, (b) 30, and (c) 40. The plots indicate how the missing observations affect the uncertainty in alignment to the model.

onds on a 2.56 GHz Intel Xeon X5660 and scales linearly with the number of particles. Because each particle is independent, parallelization can further reduce the computation time or allow for more particles to be used.

5.5 Intrusion detection

The core motivation in this work is to use the estimated latent trace **z** to characterize the normal behavior in an engineering control system for the purposes of intrusion detection. After estimating the parameters of the CPM model and obtaining residuals via particle filtering, a method for determining evidence of anomalous behavior, indicative of an intrusion, is used with the filtered residuals.

Algorithm 3 Online Particle Filter with Trained Model

```
  **Initialization:** Draw $n_p$ particles $\{\tau_{0,j}, \phi_{0,j}\} \sim P(u_k, \pi_{0,k})$ for replicate $k$. Set all weights, $w_j = 1/n_p$
  **Input:** Data $x_{i,k}$, Particles $\{\tau_{1:(i-1),j}, \phi_{1:(i-1),j}\}$, Residuals $e_{1:(i-1),j}$, $j \in \{1, \ldots, n_p\}$
  **for** $j = 1$ **to** $n_p$ **do**
      Sample $\pi_{i,j} \mid \pi_{i-1,j}$ from the transition distribution $T_{\pi_{i-1},\pi_i}$ in Equation 5.2
      Calculate residual $e_{i,j} = x_i - z(\tau_{i,j}) u_j \phi_{i,j}$
      Set weight $w_j = w_j * P(x_i \mid u_j, \pi_{i,j})$
      Normalize weights $w_j = w_j / \sum_j w_j$
  **if** $\frac{1}{\sum w_j^2} < n_p/2$ **then**
      Resample $n_p$ particles with replacement with weights $w_{1:n_p}$
      Reset all weights to $w_{1:n_p} = 1/n_p$
  **Output:** Particles $\{\tau_{1:i,j}, \phi_{1:i,j}, u_j\}$, Residuals $e_{1:i,j}$, $j \in \{1, ..., n_p\}$
```

5.5.1 Background

Techniques for identifying intrusions have been widely developed (Lee and Stolfo, 1998; Marchette, 2001). Intrusion detection also has strong connections to anomaly and fraud detection (Bolton and Hand, 2002; Chandola *et al.*, 2009). The approach to identify intrusions depends on the available data for the system and the likely threats. One approach is to identify deviations from the typical operation of the system, typically via clustering (Marchette, 2001) or by identification of local neighborhoods of typical data and identifying observations outside these regions (Breunig *et al.*, 2000). These methods do not assume any structure to the impact of an intrusion on the network, allowing them to be more robust to new types of threats; however, the cost of this flexibility is decreased power of detection.

Another approach is to assume that threats have a known signature and develop methods for identifying these patterns in the systems of interest. The attack signature may be known from previous, labeled data collected on the same or similar systems. This data can be used with classification algorithms to identify data characteristic of an intruder (Bolton and Hand, 2002; Marchette, 2001).

Often, training data for classifiers on intrusions in a system may be difficult to come by. In that case another method is to generate synthetic data expected to be similar to the signature of an attack and use this as training data for a classifier (Abe *et al.*, 2006). The synthetic data does not need to exactly match the structure of an intrusion, it only needs to more closely resemble a true attack than normal data from the classifier's view. Synthetic data would incorporate expert knowledge on the types of vulnerabilities in the system.

Unfortunately, because both of these methods rely on new attacks resembling the data used to train the classifier, they would not be robust to new forms of attack. Hybrid approaches utilizing both clustering and classification potentially provide a balanced approach to achieve benefits of both methods while mitigating their risks (Tsai *et al.*, 2009).

The examples will consider a simple form of known attack signature: a *level change* in the characteristic behavior, i.e., a consistent increase or decrease in the overall scale of the data for some duration. Section 5.6.1 describes the method for injecting such level changes into the two data sets. Any intrusion detection method that works with model residuals will be compatible with the filtered residuals computed as described in Section 5.4. In Section 5.5.2 outlines a standard method from the changepoint literature which is a form of the cumulative sum (CUSUM) technique (Qiu, 2013). Then we demonstrate the use of this CUSUM technique on the model's filtered residuals for the synthetic data and NSSB data in Section 5.6.4 and Section 5.6.7.

5.5.2 CUSUM Method for Detection

The final step to the proposed methodology for intrusion detection is the application of a process control procedure to the filtered residuals to identify when the system behavior is atypical (Qiu, 2013). We use a standard intrusion detection method called CUSUM (for **cu**mulative **sum**) to demonstrate the utility of the filtered residuals in this context (Page, 1954). Using the filtered residuals computed as in Section 5.4 as input, we apply a form of CUSUM test statistic as described in Qiu (2013). It can be implemented in many ways. We compute a test statistic C_i for residual e_i at time i:

$$\begin{aligned} C_i &= \max(C_i^+, |C_i^-|) \text{ where,} \\ C_i^+ &= \max(0, C_{i-1}^+ + e_i), \text{ and} \\ C_i^- &= \min(0, C_{i-1}^- + e_i). \end{aligned} \tag{5.6}$$

In the particle filter framework, we could either compute this statistic for each particle j and report C_i averaged over all n_p particles, or calculate C_i on the residuals averaged over all n_p particles. Little empirical difference was found between these approaches, so, for the reported examples, C_i is calculated on the averaged residuals for computational convenience.

Large C_i occur when residuals show a strong change. The average run length until a false detection is a common metric for the quality of a process control algorithm. Because the replicate series are split by day, we instead control for the number of detections per day on data with no intrusions. For both the synthetic data set and the NSSB data, we choose a threshold on C_i so that the false detections per day are 1 per 4 days using the filtered residuals from the training data. When C_i exceeds this threshold, we declare that an intrusion has occurred.

5.6 Model Performance

In this section, the performance of the proposed methodology for intrusion detection is investigated. The approach is applied to a synthetic data example and an HVAC application. For both examples, the proposed modified CPM approach is applied to a set of intrusion free time series to estimate the latent trace. Next, we apply the particle filter to on-line data and obtaining filtered residuals. We introduce a level change as the signature of an intrusion for this set of data. We then explore the performance of the CUSUM detection statistic for identifying the level change. Because we do not have data with known intrusions in the application, a strategy for inserting a type of intrusion is described in Section 5.6.1. The performance of the methodology is investigated on these intrusions.

5.6.1 Injecting level changes

To simulate a simple attack signature, we add a constant level change to the series to represent an intrusion and consider level changes with different magnitudes and different durations. The time of the intrusion is randomly chosen at the equivalent of either 1 AM or 1 PM. These times are chosen so that the performance to detect intrusions can be separated between intrusions at times when the latent trace is more smoothly varying (night) and times where the variation in the latent trace is much greater (day). We draw the magnitude and duration of the intrusion with a 2D maximin Latin Hypercube design using the `lhs` package in `R` with the time rounded to the nearest minute (Carnell, 2012). The distribution for the magnitude of the level change is $\mathsf{N}(0, 10\sigma_x)$ while the duration of the intrusion was sampled uniformly on the interval [1,16] observations in the synthetic example and [1,360] minutes for the NSSB example.

For the two data sets, the proposed alignment model is first estimated using a set of intrusion-free training data. The test set is then constructed as follows. For the synthetic example, a test set of 1000 replicate time series is generated from the proposed CPM as described in Section 5.6.2, and level changes are injected into 500 of the series. For the measured NSSB data set, we generate 200 level change parameter settings and then make the intrusion cases by injecting level changes into one of the 14 series from March 2016 mentioned in Section 5.6.6 chosen at random. In both cases the estimated latent trace is used to calculate residuals e_i in the test sets.

5.6.2 Synthetic data

We first consider an example where the data are generated from a known distribution, with know intrusions. For computational convenience we suppose that we acquire only two measurements per hour (48 per day) instead of one per minute as in the NSSB data. We use the following function to generate data exhibiting approximately periodic behavior and containing signal variability that changes over time $t \in [0, 48]$:

$$x(t) = 4 + \sin\left(\frac{2\pi t}{24}\right) + \mathrm{e}^{-\frac{1}{100}(t-30)^2} \sin\left(\frac{2\pi t}{4} + 3\right). \tag{5.7}$$

This function is shown in black in Figure 5.9 and serves as the latent trace. Figure 5.9 also shows 60 replicate time series generated from this latent trace using the continuous profile model that we define in Section 5.2. First, a scale factor for the first day's time series, u_1, is drawn from a uniform distribution (i.e., Uniform(0.97, 1.03)). Next, the initial latent time for the first observed series, $\tau_{1,1}$, is sampled from $\tau^{(1)}, \ldots, \tau^{(6)}$ with uniform probability, as well as the first scaling state $\phi_{1,1}$ with uniform probability from all the potential scaling states. The first observation, $x_{1,1}$ is emitted by drawing a residual $e_{1,1} \sim N(0, 0.3^2)$ and adding it to the latent value $z(\tau_{1,1}\phi_{1,1}u_1)$. The model then transitions to the next pair of time and scale states, $\tau_{2,1}$, $\phi_{2,1}$ according to the transition probabilities. Another observation is generated and the process repeats until all $N_k = 48$ observations have been generated for the first time series. This process is then repeated 59 more times to generate the full 60 replicate series.

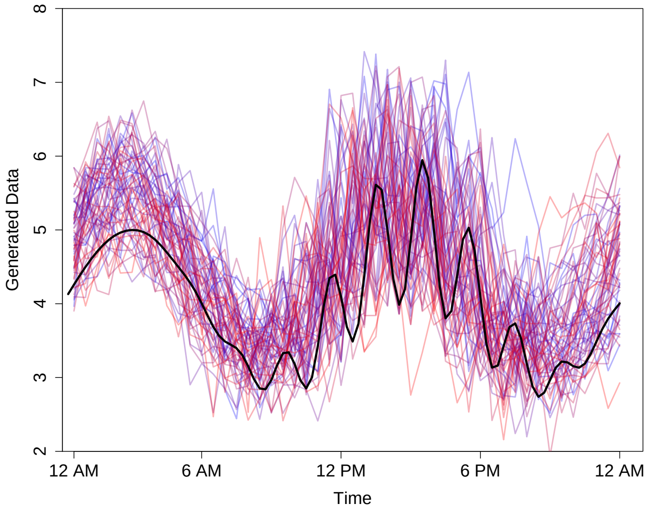


Figure 5.9: Sixty replicate time series of synthetic data generated from the underlying latent trace (in black) using the continuous profile model described in Section 5.2. We used the following parameter settings (defined in Section 5.2) for each observation i in replicate series k: error standard deviation of the data $\sigma_x = 0.3$, global scaling parameter $u_k \sim \mathsf{Unif}(0.97, 1.03)$, local scaling parameters $\phi_{i,k} \in \{1.0, 1.1, 1.2\}$, $M = 100$, and $\tau_{i,k}$ is in the length 100 sequence of evenly spaced values between 1 and 48.

5.6.3 Estimation of Characteristic Behavior With Synthetic Data

The left panel of Figure 5.10 shows the same 60 replicate time series originally shown in Figure 5.9 with the true latent trace used to generate them in black. The right side shows the true latent trace again in black along with the estimated latent trace found using the proposed method in green. The points in the right panel of Figure 5.10 show the scatter of each of the 60 replicate time series around the estimated latent trace after aligning them to the trace. We fit the CPM with a smoothing parameter $\lambda = 1.67$, which we set by minimizing the mean squared error over different values of λ on a validation set of 60 independently generated series.

In the figure we see that the estimated latent trace in green has recovered the main structure in the true latent curve in black, including the change in signal variance and the magnitude of the large-scale oscillations in the signal. Short-scale structure in the estimated latent trace at early times indicates that the value of smoothing parameter found by cross validation may have been smaller than optimal for reconstructing the early smoothness. The model also captures the correct time alignment, though the estimated latent trace is slightly shifted to the right of the true latent trace at early times.

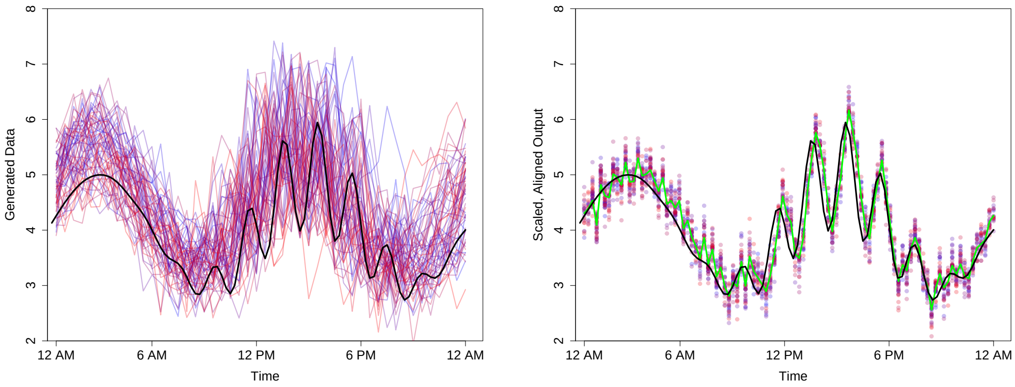


Figure 5.10: **Left:** Sixty replicate series of synthetic data generated from the underlying latent trace (in black) using the model described in Section 5.2. **Right:** The estimated latent trace in green along with the replicate time series shown as points after aligning them to the estimated latent trace. The true latent trace is included in black for comparison.

The left panel of Figure 5.11 is a plot of the corresponding residuals for each series. Encouragingly, the residuals are centered around zero and have little discernible pattern. The Q-Q plot suggests that the residuals can be reasonably described as normally distributed, while the autocorrelation plot confirms that they are uncorrelated as desired.

5.6.4 Intrusion Detection With Synthetic Data

With the estimated model, we now perform on-line filtering for new series. Figure 5.12 shows four stages of the filtering where the observed data includes a level change simulating

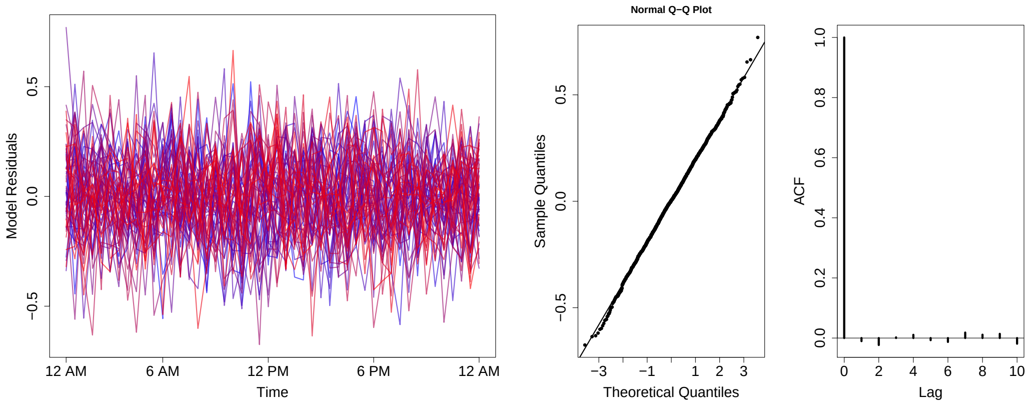


Figure 5.11: **Left:** Observed residuals from the esimated latent trace for the synthetic data. The resulting residuals are centered on zero and show little residual pattern. **Center:** The Q-Q plot suggests that the residuals are normally distributed. **Right:** The autocorrelation function (ACF) plot shows little autocorrelation in the residuals.

an intrusion into the network. In the 24th observation in Figure 5.12a, the alignment looks similar to Figure 5.7d, with significant uncertainty over the current value of the latent state, though one state has distinctly more posterior probability than the nearby states. The level change is inserted at 28th observation, which is shown in Figure 5.12b. Because the data point appears somewhat atypical, the posterior probability is concentrated almost entirely on a single state. However, the residual is not large because the point is plausibly emitted from that state. However, the subsequent observations are not able to align to the latent trace. Figure 5.12c and Figure 5.12d show the results of the particle filter at the 31st and 40th observations. With the level change, the resulting residuals are quite large and indicative of anomalous behavior.

Applying the particle filtering method results in the filtered residuals shown in Figure 5.13. The left panel shows the resulting residuals from 500 intrusion-free replicate series while the right panel shows the filtered residuals from the 500 test series with an added level change as described in 5.6.1. The intrusion-free residuals appear homoscedactic and have few outliers. Large residuals are visibly more frequent in the intrusion data on the right image.

The residuals from intrusion-free series can be used to choose the threshold for identification of an intrusion. Figure Figure 5.14 shows how the false detection rate decays as the threshold increases for the intrusion-free residuals in this example. To meet the stated threshold of one false intrusion every four days, the threshold for the CUSUM test statistic was set to 4.0.

For the 500 series with an injected level change, the CUSUM method above correctly identified 64.6% of the intrusions. Figure 5.15 shows the distribution of true positives (black) and false negatives (red) across different intrusion durations and magnitudes. The

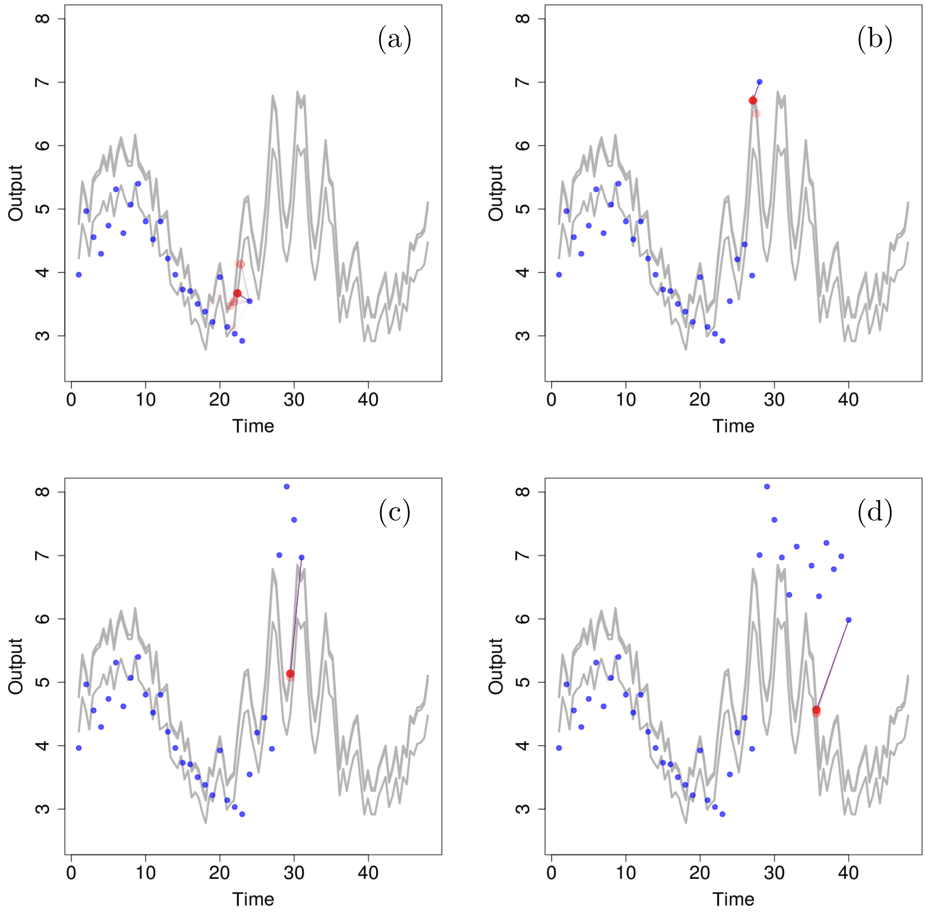


Figure 5.12: On-line alignment of particles to the latent trace with a synthetic intrusion at (a) the 24th observation, (b) the 28th, (c) the 31st, and (d) the 40th. This example shows the filtering approach on data with an injected level change and the large residuals that result.

way the false negatives are clustered around a magnitude of zero shows, not surprisingly, that "quieter" intrusions are harder to detect. This effect is mitigated as intrusions get longer as shown by the narrowing band of red points around zero as duration increases. That is, the ability to detect smaller magnitude intrusions improves as durations get longer. In the 500 series in the test set that did not have injected level changes, an average of 0.232 false detections per day were identified, which is slightly less than a rate of one false intrusion every four days used to set the detection threshold.

Analyzing the delay in detecting intrusions is another important metric for security. The histogram in Figure 5.16 shows the number of intrusions detected at different delays after onset. We can see that detections happen quickly after the beginning of the intrusion, with the majority of detections happening within the first 4 observations after the beginning of the anomaly. Particularly large values for the detection delay may actually be false positives

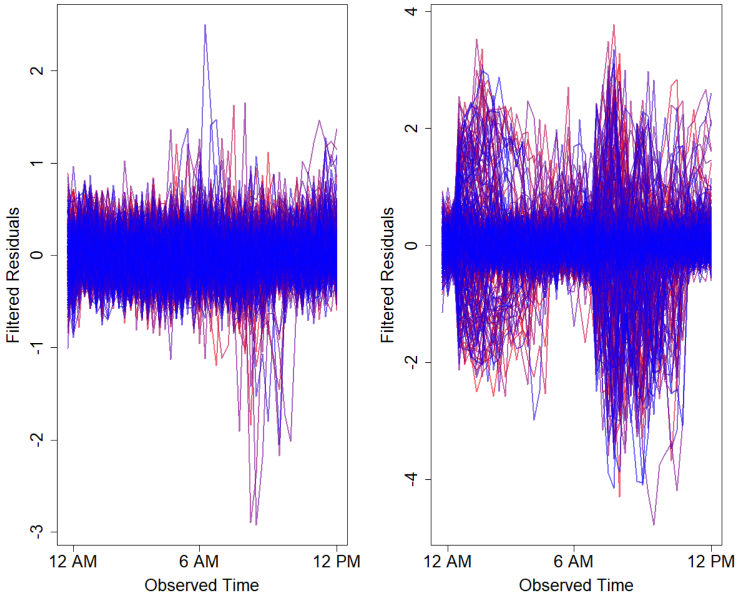


Figure 5.13: **Left:** Filtered residuals from 500 replicate series in the synthetic example without injected level change simulating an intrusion. **Right:** Filtered residuals for 500 replicate series with injected level changes at either 1 AM or 1 PM.

on the post-intrusion residuals, though they may instead result from the end of an intrusion being detected as a level change.

5.6.5 Real World Example

Modern HVAC systems report time series data from a large number of diverse sensors throughout a building. Available data for this application includes temperature, air flow, and air valve positions. Figure 5.17 shows daily replicate time series from these three sensor types at two different locations in the building. We can see that from day-to-day the overall behavior for the time series in each panel of the figure is quite similar, but different days show variation in overall scale and the location of consistent features, as well as measurement error. Monitoring these for anomalies can provide early indications of failure, or even of a cyber-attack on the HVAC system.

We developed the approach to analyze time series from the heating, ventilation, and air conditioning (HVAC) system of the National Security Sciences Building at Los Alamos National Laboratory. We collected 894 time series from sensors throughout this 275,000 square foot building. Measurements for each series are recorded approximately once per minute.

Broadly speaking, each of the six series in Figure 5.17 shows higher variability in the behavior of the series during plausible “working hours” of 0600-1830 and smoother changes

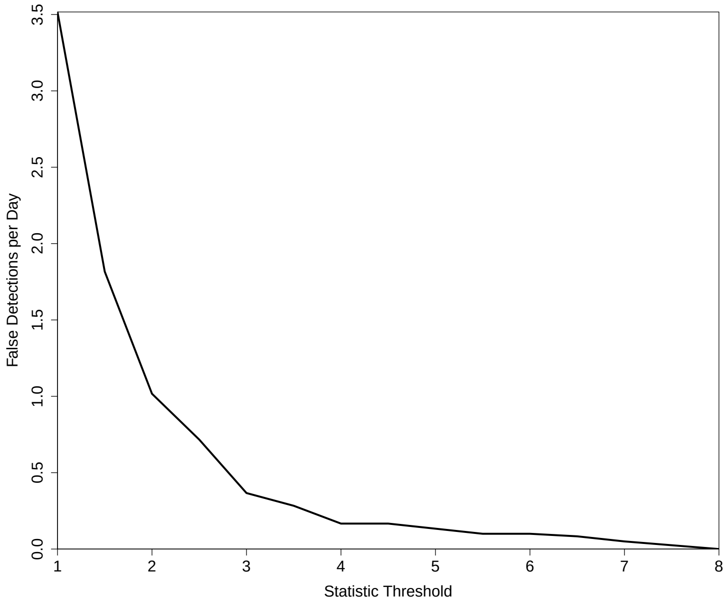


Figure 5.14: False detections per day when applying the CUSUM method to the filtered residuals for the synthetic data in Section 5.6.1. A threshold of 4 is the smallest with a detection rate of at most 1 per 4 days on the filtered residuals using the training data.

during the overnight periods. This work focuses on the middle of the week - Tuesdays, Wednesdays, and Thursdays - where human behavior will be most consistent.

Figure 5.17 shows time series from Tuesdays and Thursdays in February 2016. The panels show the temperature, air flow rate through the HVAC system, and what percentage open the air flow value was for two rooms in the NSSB. The replicates for each time series have similar shapes over the course of each day, however the timing of features in each curve vary between observed series. In this chapter the alignment and detection model are applied to the interior location temperature data from the above series.

5.6.6 Demonstration with Application Data

The left side of Figure 5.18 shows 12 replicate time series for the temperature of the interior location in the NSSB on Tuesdays, Wednesdays, and Thursdays in February 2016. These come from an arbitrarily chosen temperature sensor. We fit the alignment model using the E-M algorithm as described in Section 5.3. We used leave-one-out cross-validation (Stone, 1974) to find an optimal smoothing parameter $\lambda = 1096.6$ by minimizing the mean-squared prediction error.

Fourteen additional replicate series are withheld from March 2016 as a test set for intrusion detection in Section 5.6.7. The right side of Figure 5.18 shows the estimated latent trace in green along with the scatter of each replicate around the latent trace after alignment. We can see that the structure in the latent trace corresponds to behavior somewhat visible

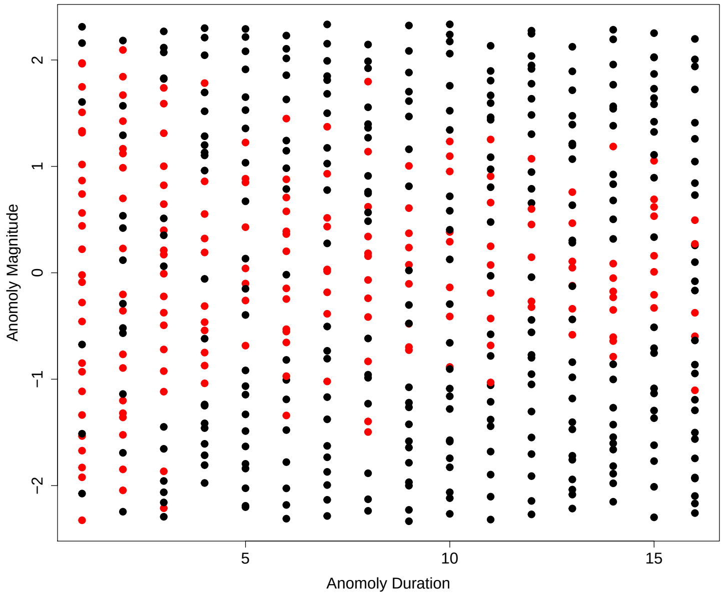


Figure 5.15: True positives (black) and false negatives (red) when using the CUSUM method against the 500 synthetic test cases with injected intrusions as the magnitude and duration of the intrusion are varied. The method correctly identified 64.6% of the intrusions. We see an interaction between duration and magnitude in that longer intrusions are more reliably identified at smaller magnitudes than short. Intrusion detection methods that are more sophisticated than CUSUM may be able to improve on these results.

on the replicate series. The temperature steeply drops at the beginning of the day, slowly rises until lunch, drops toward the late afternoon, and finally spikes again at the end of the day before slowly and smoothly rising overnight.

Figure 5.19 displays the residuals from fitting the the proposed CPM to the NSSB data. The residuals are approximately normal but show heteroscedasticity between the daytime and nighttime portions of the day.Tsay (1988) proposed a method for variance adjustment to residuals in a time series to identify and remove heteroscedasticity from residuals, which we give in detail in Appendix 3 and summarize here. The method first finds a split point which maximizes the variance ratio between the partitions. If the ratio exceeds a specified threshold, the residuals in the second split are scaled by the square root of the variance ratio to equalize the sample variances in the splits. This algorithm iterates until the largest difference in variance between splits is smaller than the threshold. We apply this method to adjust the variance of the model residuals to mitigate the heteroscedasticity. Assuming that the future filtered residuals display similar heteroscedasticity, the split points and scaling values will be applied to the filtered residuals before the calculation of the CUSUM detection statistic.

The adjusted residuals also show autocorrelation, indicating some structure in the residuals that is not captured in the model. This may be a result of using a constant smoothing

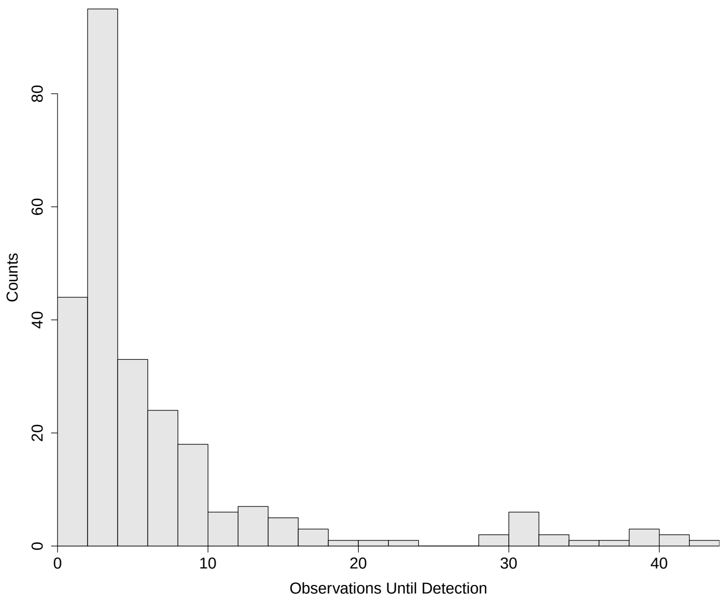


Figure 5.16: The number of observations delay between the onset of a synthetic intrusion and its identification in the on-line residuals. Most synthetic intrusions are identified quickly by the model.

penalty λ across the entire latent space, which restricts the flexibility of the latent trace in regions that vary quickly in order to avoid overfitting in more slowly varying regions.

5.6.7 Intrusion Detection With NSSB Data

Figure 5.20 shows the filtered residuals for the March 2016 NSSB data. The left image shows the fourteen intrusion-free replicate series, while the 200 replicate series with injected level changes are shown on the right. The residuals show some heteroscedasticity between the day and night behavior and shows some large residuals at the transition between these states. The largest intrusions are clearly visible in the right image at 1 AM and 1 PM.

The CUSUM method detected 42% of these intrusions, which is consistent with the performance on the synthetic data. Figure 5.21 shows the ability to discriminate intrusions in this application example as a function of the size and duration of the intrusion. We can see that the ability to detect level changes is strongly dependent on the magnitude of the change, as in Section 5.6.4. A more complex model for detecting intrusions in the residuals may provide more power, but this shows value in using the residuals from the CPM for detecting intrusions in practical time series.

Calculating filtered residuals and the detection test statistics from the fourteen series in from March 2016 without injected level changes, an average of 0.214 false detections per day were identified, which, as in Section 5.6.4 is slightly better than rate of one false intrusion every four days used to set the detection threshold.

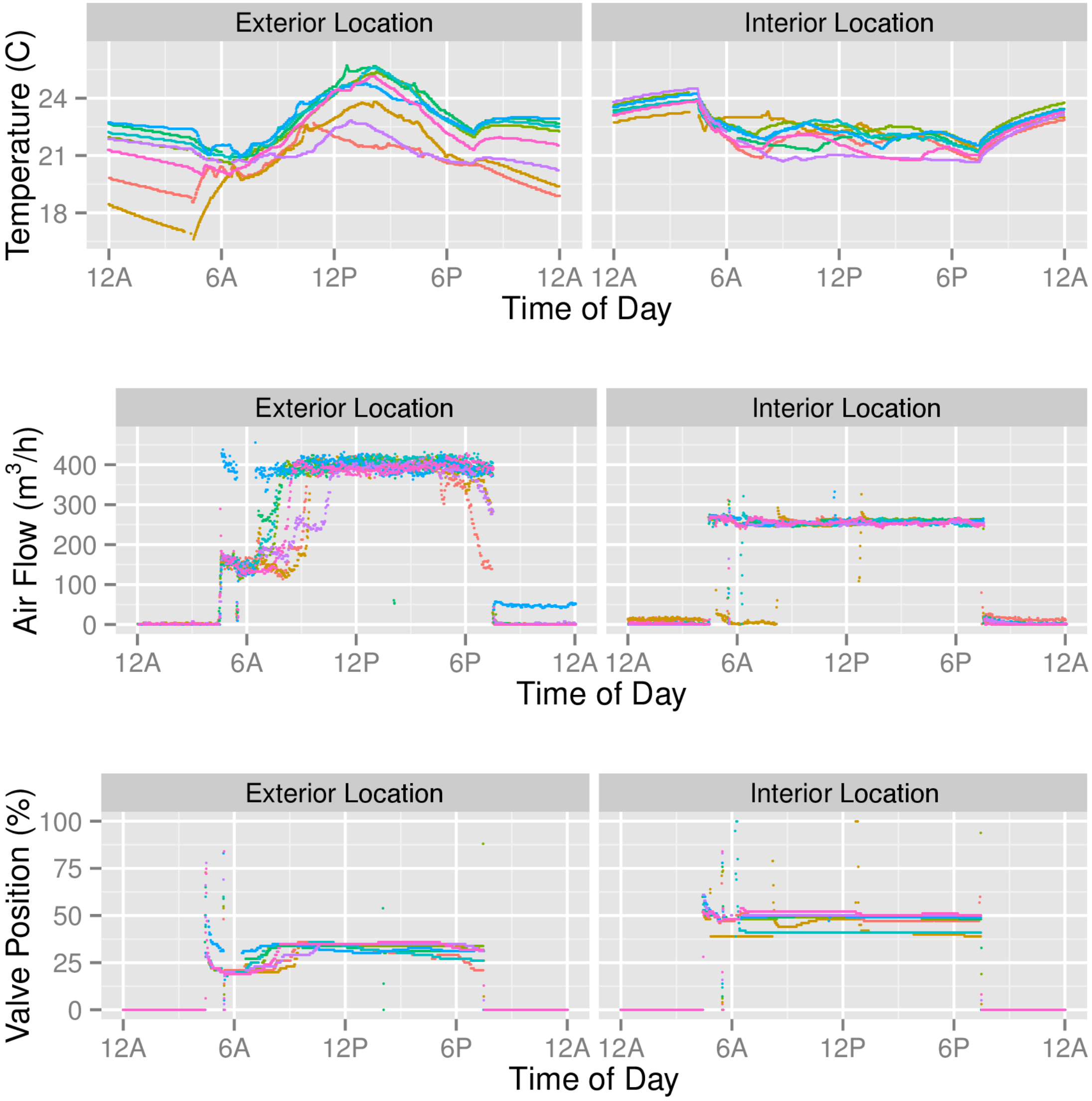


Figure 5.17: Time series for temperature, air flow, and valve positions from a position near the exterior and position in the center of the National Security Sciences Building (NSSB) at Los Alamos National Laboratory. Different colors represent measurements from different Tuesdays and Thursdays in February 2016. The series are typical in the sense they display strong diurnal patterns.

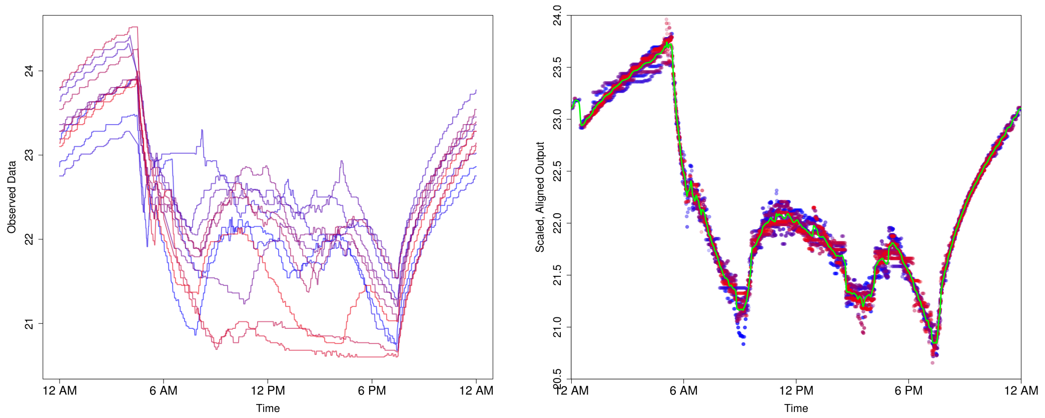


Figure 5.18: **Left:** Daily replicate series of raw temperatures from one sensor in the NSSB for Tuesdays, Wednesdays, and Thursdays in February 2016. We see similar but shifted and warped behavior across the different replicate series. **Right:** The estimated latent trace in green along with the results of aligning the replicate series to the estimated latent trace shown as points. The latent trace appears to reproduce the approximate characteristics we see visually in the raw replicate observations.

5.7 Discussion

A method for estimating nominal behavior of a set of time series with shared latent signal in the presence of time-warping and using particle filtering to align new series in an on-line environment has been developed in this chapter. The model is flexible to capture complex shared behavior and can be applied to a wide range of time series models. Using a particle filtering approach with the hidden Markov model to obtain on-line residuals handles uncertainty in aligning new replicate time series to the characteristic behavior of the system, while providing a fast algorithm for on-line updating of the alignment for new data. Accounting for uncertainty in on-line alignment of time series makes the alignments more robustness in the residuals.

Our proposed methodology is available in the `R` package `alignts`. This package has been built with `R` and Fortran 95 with OpenMP support, with the latter available to ease the computational burden with large state spaces.

There are a number of areas that can be addressed for future research, both with the alignment model and with the filtering and detection of intrusion. For the alignment model:

- While the computational cost of the model is greatly reduced for large state spaces, the model is still quite expensive to fit. This, in turn, provides a barrier to extensive cross validation for choosing aspects of the model such as the number of scaling states Q.

- The model as presented assumes homoscedasticity in the residuals and required post-hoc adjustment of the filtered residuals for anomaly detection. In the examples the

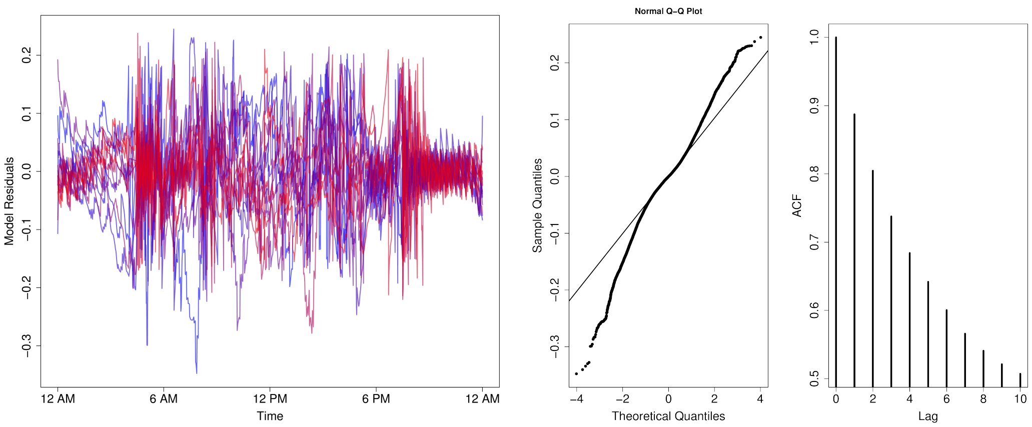


Figure 5.19: **Left:** Observed residuals from the alignment model shown on the right in Figure 5.18 for the NSSB data. We see some evidence of heteroscedacticity between the day and night residuals. **Center:** The Q-Q plot shows evidence of heavy-tailed residuals. **Right:** The ACF plot shows evidence of autocorrelation in the residuals.

method proposed by Tsay (1988) is used. A more holistic way to handle heteroscedasticity is to include hidden Markov states for different variances and allow the model to transition between states, similar to the time and scaling states in the CPM. However, these additional states greatly increase the computational cost due to the expanded HMM.

- If the alignment model is estimated on data that is not representative of future observations, the natural change in behavior of future observations may look like anomalies. In the example, the model is estimated on data from February and then used with particle filtering for on-line alignment on data from March. Because of seasonal changes in weather, it is possible that training a model on one month would not be appropriate for anomaly detection in the next. The training data should be reflective of the range of expected behavior for the system.

- By splitting the days into replicate series, the end of a day represents a vulnerability. An invader with knowledge of the hand-off time could make changes to the system that would be adjust for by the global scaling parameter and fit normally within the model. The prior distribution on the global scaling parameter for the filtering provides some protection against this as it controls the support for the values of the parameter. One solution to the vulnerability introduced by the hand-off at edge of the day is to perform the filtering in two series at the same time, one starting from midnight and another starting from an offset time, such that the hand-off for one series happens during filtering for the other. The redundancy allows for coverage of system behavior during the hand-offs.

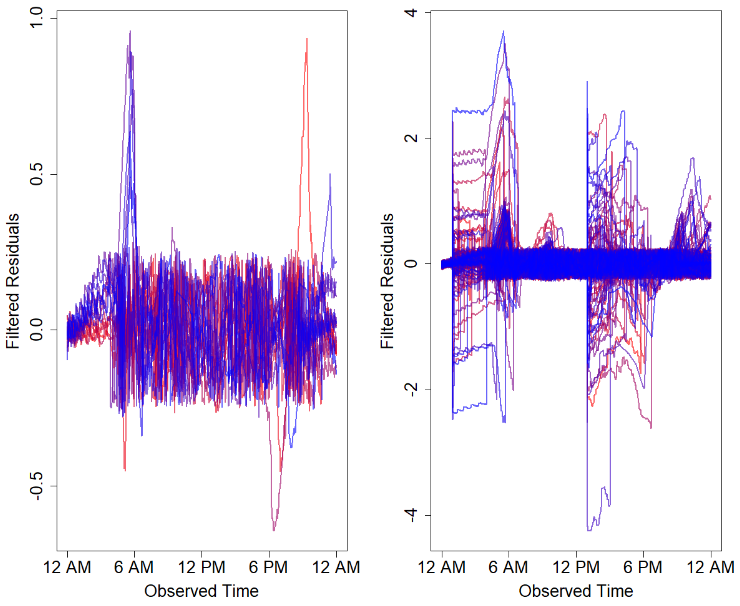


Figure 5.20: **Left:** Filtered residuals from 14 replicate series in the from the NSSB data in March 2016 without injected level change simulating an intrusion. **Right:** Filtered residuals for the 200 replicate series with injected level changes at either 1 AM or 1 PM.

- The nature of the proposed model, once estimated, only requires on-line data be split into replicate days because of the global scaling parameter. Over time, constantly filtered data is likely to drift from the scale of the latent curve if this parameter is kept fixed. An extension of the filtering to allow smooth variation of the uniform scaling parameter could remove the need to split the data by day.

- While the intrusion detection examples presented in this chapter used a simple level change to represent an intrusion, true intrusions could manifest in a wide variety of ways. The goal of the alignment is to remove the typical behavior and the goal of the particle filter is to get on-line residuals with uncertainty. These residuals can be used with many other methods beyond the CUSUM or with other breakpoint detection models in order to find wider classes of system intrusion.

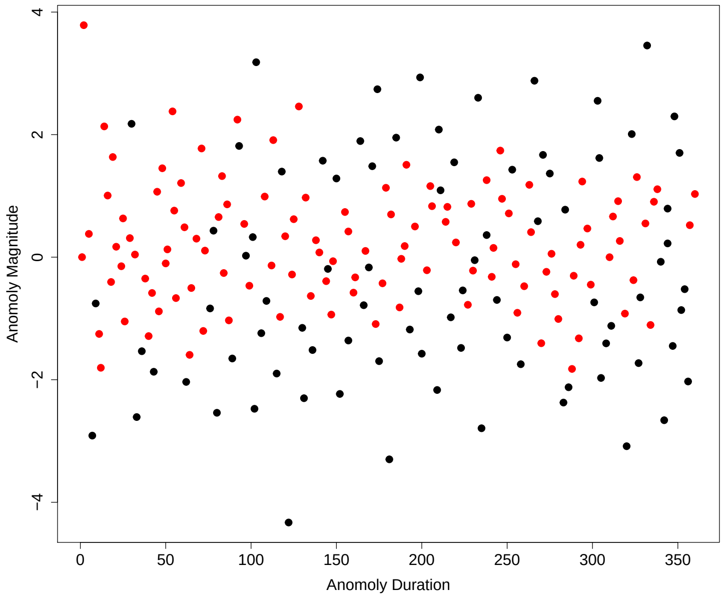


Figure 5.21: True positives (black) and false negatives (red) when using the CUSUM method against the 200 test cases as the timing, magnitude, and duration of the intrusion are varied. The method correctly identified 42% of the intrusions.

Chapter 6

Conclusions and Future Work

In this thesis, methodologies were developed for model calibration and validation for computer simulators. The new methodology for model calibration with Gaussian process emulators and Poisson measurement error was proposed in Chapter 3. The proposed Bayesian hierarchical formulation and sampling generalize to other non-Gaussian observation error and provide for flexibility to handle other constraints on the QoI. The approach was applied to calibrate a radiation transport model, PDT, developed by collaborators in CERT at Texas A&M University, as well as to a synthetic data problem closely paralleling a scattering cross section in nuclear theory. Further applications of this methodology to categorical, over-dispersed Poisson outcomes, or other constraints are of future interest. Additionally, development of alternative methods for fitting the Bayesian hierarchical model including potentially using variational Bayesian methods or integrated, nested Laplace approximations (Rue *et al.*, 2009) is of future interest.

In Chapter 4, a new approach to validation of computer models was proposed using goodness-of-fit testing within a Bayesian model assessment framework. The goal of the validation test was to assess evidence that the observed distribution of field data was consistent with the distribution function generated by propagating aleatory inputs through the emulated simulator. The proposed procedure summarizes the evidence with respect to the epistemic uncertainty in the problem and provides a formal hypothesis test for evidence of disagreement between the distributions. The methodology was applied to validation tests of PDT. Further interest at CERT lies in developing methodology for sequential validation testing through a hierarchy of experiments. Exploration of the role of the discrepancy as a tool for understanding model-form error in validation and for directing improvement of the model is an area of future work. Some of the core ideas have been proposed by Goldstein and Rougier (2009) with reified Bayesian models and Joseph and Yan (2015) on engineering-driven statistical adjustment, where the discrepancy is given a purposeful form and treated as part of the full model.

In Chapter 5, we proposed innovations to a “hidden Markov model”-based time series alignment method and the use of particle filtering for on-line intrusion detection in cyber-physical networks. The proposed procedure is to first estimate the model using replicate time series and then apply a Bayesian particle filter to obtain on-line residuals accounting for uncertainty in the estimated path through the latent Markov states. These residuals are then used with a process control test for identifying changepoints in the residuals that may be evidence of an intrusion into the system. The model was applied to data from the HVAC system of the NSSB at Los Alamos National Laboratory. Further work to reduce size of the hidden Markov state space would greatly improve computational performance. The process control method used in Chapter 5 tests for only a simple form of anomaly in the time series - a change in level of the residuals. More complex residual analysis may provide improved detection statistics that capture a more broad range of signatures of intrusion.

Bibliography

Naoki Abe, Bianca Zadrozny, and John Langford. Outlier detection by active learning. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 504–509. ACM, 2006.

Michael P Adams, Marv L Adams, W. Daryl Hawkins, Timmie Smith, Lawrence Rauchwerger, Nancy M Amato, Teresa S Bailey, and Robert D Falgout. Provably optimal parallel transport sweeps on regular grids. *Proc. International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*, 2013.

Alan Agresti. *Categorical Data Analysis*. John Wiley & Sons, 2013.

Shan Ba, V Roshan Joseph, et al. Composite gaussian process models for emulating expensive functions. *The Annals of Applied Statistics*, 6(4):1838–1860, 2012.

Leonardo S Bastos and Anthony O'Hagan. Diagnostics for gaussian process emulators. *Technometrics*, 51(4):425–438, 2009.

Leonard E Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The annals of mathematical statistics*, 41(1):164–171, 1970.

MJ Bayarri and James O Berger. P values for composite null models. *Journal of the American Statistical Association*, 95(452):1127–1142, 2000.

Maria J Bayarri, James O Berger, Rui Paulo, Jerry Sacks, John A Cafeo, James Cavendish, Chin-Hsu Lin, and Jian Tu. A framework for validation of computer models. *Technometrics*, 49(2), 2007.

MJ Bayarri, JO Berger, John Cafeo, G Garcia-Donato, F Liu, J Palomo, RJ Parthasarathy, R Paulo, Jerry Sacks, and D Walsh. Computer model validation with functional output. *The Annals of Statistics*, pages 1874–1906, 2007.

Maria J Bayarri, James O Berger, Eliza S Calder, Keith Dalbey, Simon Lunagomez, Abani K Patra, E Bruce Pitman, Elaine T Spiller, and Robert L Wolpert. Using statistical and computer models to quantify volcanic hazards. *Technometrics*, 51(4):402–413, 2009.

Roger L Berger and Dennis D Boos. P values maximized over a confidence set for the nuisance parameter. *Journal of the American Statistical Association*, 89(427):1012–1016, 1994.

Christopher M Bishop. Pattern recognition. *Machine Learning*, 128, 2006.

Richard J Bolton and David J Hand. Statistical fraud detection: A review. *Statistical science*, pages 235–249, 2002.

Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *ACM Sigmod Record*, volume 29, pages 93–104. ACM, 2000.

Eric Brochu, Vlad M Cora, and Nando De Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

Rob Carnell. *lhs: Latin Hypercube Samples*, 2012. R package version 0.10.

Avishek Chakraborty, Derek Bingham, Soma S Dhavala, Carolyn C Kuranz, R Paul Drake, Michael J Grosskopf, Erica M Rutter, Ben R Torralva, James P Holloway, Ryan G McClarren, et al. Emulation of numerical models with over-specified basis functions. *Technometrics*, 59(2):153–164, 2017.

V Chandola, A Banerjee, and V Kumar. Anomaly detection: Survey. *ACM Computing Survey, ACM*, 2009.

O.F. Christensen and P.J. Ribeiro Jr. georglm - a package for generalised linear spatial models. *R-NEWS*, 2(2):26–28, 2002. ISSN 1609-3631.

Andrew R Conn, Katya Scheinberg, and Luis N Vicente. *Introduction to derivative-free optimization*. SIAM, 2009.

Paul G Constantine, Eric Dow, and Qiqi Wang. Active subspace methods in theory and practice: applications to kriging surfaces. *SIAM Journal on Scientific Computing*, 36(4):A1500–A1524, 2014.

Ralph B D'Agostino and Michael A Stephens. Goodness-of-fit techniques, 1986.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

PJ Diggle, PJ Ribeiro, and Model-based Geostatistics. *Springer Series in Statistics*. Springer, 2007.

Peter Diggle. A kernel method for smoothing point process data. *Applied statistics*, pages 138–147, 1985.

Petar M Djuric, Jayesh H Kotecha, Jianqui Zhang, Yufei Huang, Tadesse Ghirmai, Mónica F Bugallo, and Joaquin Miguez. Particle filtering. *IEEE signal processing magazine*, 20(5):19–38, 2003.

Arnaud Doucet, Simon Godsill, and Christophe Andrieu. On sequential monte carlo sampling methods for bayesian filtering. *Statistics and computing*, 10(3):197–208, 2000.

James Durbin. Weak convergence of the sample distribution function when parameters are estimated. *The Annals of Statistics*, pages 279–290, 1973.

Robert G Easterling and James O Berger. Statistical foundations for the validation of computer models. In *Computer Model Verification and Validation in the 21st Century Workshop, Johns Hopkins University*, 2002.

Sean R. Eddy. Profile hidden markov models. *Bioinformatics (Oxford, England)*, 14(9):755–763, 1998.

Paul Fearnhead and Peter Clifford. On-line inference for hidden markov models via particle filters. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(4):887–899, 2003.

Marco AR Ferreira and Herbert KH Lee. *Multiscale modeling: a Bayesian perspective.* Springer Science & Business Media, 2007.

Scott Ferson, Vladik Kreinovich, Lev Ginzburg, Davis S Myers, and Kari Sentz. Constructing probability boxes and dempster-shafer structures. Technical report, Technical report, Sandia National Laboratories, 2003.

Scott Ferson, William L Oberkampf, and Lev Ginzburg. Model validation and predictive capability for the thermal challenge problem. *Computer Methods in Applied Mechanics and Engineering*, 197(29):2408–2430, 2008.

Scott J Gaffney and Padhraic Smyth. Joint probabilistic curve clustering and alignment. In *Advances in neural information processing systems*, pages 473–480, 2005.

Yarin Gal, Yutian Chen, and Zoubin Ghahramani. Latent gaussian processes for distribution estimation of multivariate categorical data. In *International Conference on Machine Learning*, pages 645–654, 2015.

Aytekin Gel, Tingwen Li, Balaji Gopalan, Mehrdad Shahnam, and Madhava Syamlal. Validation and uncertainty quantification of a multiphase computational fluid dynamics model. *Industrial & Engineering Chemistry Research*, 52(33):11424–11435, 2013.

Andrew Gelman and Donald B Rubin. Inference from iterative simulation using multiple sequences. *Statistical science*, pages 457–472, 1992.

Andrew Gelman and Cosma Rohilla Shalizi. Philosophy and the practice of bayesian statistics. *British Journal of Mathematical and Statistical Psychology*, 66(1):8–38, 2013.

Andrew Gelman, Xiao-Li Meng, and Hal Stern. Posterior predictive assessment of model fitness via realized discrepancies. *Statistica sinica*, pages 733–760, 1996.

Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian Data Analysis.* CRC Press, 2013.

Subimal Ghosh and PP Mujumdar. Climate change impact assessment: Uncertainty modeling with imprecise probability. *Journal of Geophysical Research: Atmospheres*, 114(D18), 2009.

Joslin Goh, Derek Bingham, James Paul Holloway, Michael J Grosskopf, Carolyn C Kuranz, and Erica Rutter. Prediction and computer model calibration using outputs from multifidelity simulators. *Technometrics*, 55(4):501–512, 2013.

Michael Goldstein and Jonathan Rougier. Reified bayesian modelling and inference for physical systems. *Journal of Statistical Planning and Inference*, 139(3):1221–1239, 2009.

Robert B Gramacy and Daniel W Apley. Local gaussian process approximation for large computer experiments. *Journal of Computational and Graphical Statistics*, 24(2):561–578, 2015.

Robert B Gramacy and Herbert K H Lee. Bayesian treed gaussian process models with an application to computer modeling. *Journal of the American Statistical Association*, 103(483):1119–1130, 2008.

Robert B Gramacy and Herbert KH Lee. Cases for the nugget in modeling computer experiments. *Statistics and Computing*, 22(3):713–722, 2012.

Robert B Gramacy and Heng Lian. Gaussian process single-index models as emulators for computer experiments. *Technometrics*, 54(1):30–41, 2012.

Robert B Gramacy and Nicholas G Polson. Particle learning of gaussian process models for sequential design and optimization. *Journal of Computational and Graphical Statistics*, 20(1):102–118, 2011.

Robert B Gramacy, Derek Bingham, James Paul Holloway, Michael J Grosskopf, Carolyn C Kuranz, Erica Rutter, Matt Trantham, R Paul Drake, et al. Calibrating a large computer experiment simulating radiative shock hydrodynamics. *The Annals of Applied Statistics*, 9(3):1141–1168, 2015.

Katrin Heitmann, David Higdon, Charles Nakhleh, and Salman Habib. Cosmic calibration. *The Astrophysical Journal Letters*, 646(1):L1, 2006.

Petter Helgesson, Denise Neudecker, Henrik Sjöstrand, Michael Grosskopf, Donald L Smith, and Roberto Capote. Assessment of novel techniques for nuclear data evaluation. In *16th International Symposium of Reactor Dosimetry (ISRD16)*, 2017.

Daniel A. Henderson, Richard J. Boys, Kim J. Krishnan, Conor Lawless, and Darren J. Wilkinson. Bayesian Emulation and Calibration of a Stochastic Computer Model of Mitochondrial DNA Deletions in Substantia Nigra Neurons. *Journal of the American Statistical Association*, 104(485):76–87, 2009.

D. A. Henderson, R. J. Boys, and D. J. Wilkinson. Bayesian Calibration of a Stochastic Kinetic Computer Model Using Multiple Data Sources. *Biometrics*, 66(1):249–256, 2010.

Dave Higdon, Marc Kennedy, James C Cavendish, John A Cafeo, and Robert D Ryne. Combining field data and computer simulations for calibration and prediction. *SIAM Journal on Scientific Computing*, 26(2):448–466, 2004.

Dave Higdon, James Gattiker, Brian Williams, and Maria Rightley. Computer model calibration using high-dimensional output. *Journal of the American Statistical Association*, 103(482), 2008.

Dave Higdon, Jim Gattiker, Earl Lawrence, Charles Jackson, Michael Tobis, Matt Pratola, Salman Habib, Katrin Heitmann, and Steve Price. Computer model calibration using the ensemble kalman filter. *Technometrics*, 55(4):488–500, 2013.

Richard G Hills and Timothy G Trucano. Statistical validation of engineering and scientific models: Background. *Sandia National Laboratories, Albuquerque, NM, Report No. SAND99-1256*, 1999.

Richard Guy Hills and Timothy G Trucano. Statistical validation of engineering and scientific models: A maximum likelihood based metric. Technical report, Sandia National Labs., Albuquerque, NM (US); Sandia National Labs., Livermore, CA (US), 2002.

Kathleen A Jackson et al. Intrusion detection system (ids) product survey. *Los Alamos National Laboratory*, 1999.

P. E. Jacob, L. M. Murray, C. C. Holmes, and C. P. Robert. Better together? Statistical learning in models made of modules. *ArXiv e-prints*, August 2017.

Frederick Jelinek. *Statistical methods for speech recognition*. MIT press, 1997.

Xiaomo Jiang and Sankaran Mahadevan. Bayesian inference method for model validation and confidence extrapolation. *Journal of Applied Statistics*, 36(6):659–677, 2009.

Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

V Roshan Joseph and Lulu Kang. Regression-based inverse distance weighting with applications to computer experiments. *Technometrics*, 53(3):254–265, 2011.

V Roshan Joseph and Huan Yan. Engineering-driven statistical adjustment and calibration. *Technometrics*, 57(2):257–267, 2015.

Durga Rao Karanki, Hari Shankar Kushwaha, Ajit Kumar Verma, and Srividya Ajit. Uncertainty analysis based on probability bounds (p-box) approach in probabilistic safety assessment. *Risk Analysis*, 29(5):662–675, 2009.

Marc C Kennedy and Anthony O'Hagan. Predicting the output from a complex computer code when fast approximations are available. *Biometrika*, 87(1):1–13, 2000.

Marc C Kennedy and Anthony O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464, 2001.

Alois Kneip and Theo Gasser. Statistical tools to analyze data representing a sample of curves. *The Annals of Statistics*, pages 1266–1305, 1992.

A Kneip, X Li, KB MacGibbon, and JO Ramsay. Curve registration by local regression. *Canadian Journal of Statistics*, 28(1):19–29, 2000.

Wenke Lee and Salvatore J Stolfo. Data mining approaches for intrusion detection. In *Usenix Security*, 1998.

H Lee, R Gramacy, Crystal Linkletter, and G Gray. Optimization subject to hidden constraints via statistical emulation. *Pacific Journal of Optimization*, 7(3):467–478, 2011.

Dooho Lee, Nam H Kim, and Hyeon-Seok Kim. Validation and updating in a large automotive vibro-acoustic model using a p-box in the frequency domain. *Structural and Multidisciplinary Optimization*, 54(6):1485–1508, 2016.

Wei Li, Wei Chen, Zhen Jiang, Zhenzhou Lu, and Yu Liu. New validation metrics for models with multiple correlated responses. *Reliability Engineering & System Safety*, 127:1–11, 2014.

David V Lindberg and Herbert KH Lee. Optimization under constraints by applying an asymmetric entropy measure. *Journal of Computational and Graphical Statistics*, 24(2):379–393, 2015.

Crystal Linkletter, Derek Bingham, Nicholas Hengartner, David Higdon, and Q Ye Kenny. Variable selection for gaussian process models in computer experiments. *Technometrics*, 48(4), 2006.

Jennifer Listgarten, Radford M Neal, Sam T Roweis, and Andrew Emili. Multiple alignment of continuous time series. In *Advances in Neural Information Processing Systems*, pages 817–824, 2004.

Jennifer Listgarten. *Analysis of sibling time series data: alignment and difference detection.* PhD thesis, University of Toronto, 2007.

Jun S Liu and Rong Chen. Sequential monte carlo methods for dynamic systems. *Journal of the American statistical association*, 93(443):1032–1044, 1998.

Xueli Liu and Mark CK Yang. Simultaneous curve registration and clustering for functional data. *Computational Statistics & Data Analysis*, 53(4):1361–1376, 2009.

Fei Liu, MJ Bayarri, JO Berger, et al. Modularization in bayesian analysis, with emphasis on analysis of computer models. *Bayesian Analysis*, 4(1):119–150, 2009.

J Loeppky, Derek Bingham, and W Welch. Computer model calibration or tuning in practice. *Technometrics, submitted for publication*, 2006.

Sankaran Mahadevan and Ramesh Rebba. Validation of reliability computational models using bayes networks. *Reliability Engineering & System Safety*, 87(2):223–232, 2005.

Christopher D Manning, Hinrich Schütze, et al. *Foundations of statistical natural language processing*, volume 999. MIT Press, 1999.

David J Marchette. *Computer Intrusion Detection and Network Monitoring: A Statistical Viewpoint.* Springer Science & Business Media, 2001.

Peter McCullagh, John A Nelder, and P McCullagh. *Generalized linear models*, volume 2. Chapman and Hall London, 1989.

Joshua Mullins, You Ling, Sankaran Mahadevan, Lin Sun, and Alejandro Strachan. Separation of aleatory and epistemic uncertainty in probabilistic model validation. *Reliability Engineering & System Safety*, 147:49–59, 2016.

Iain Murray, Ryan Prescott Adams, and David JC MacKay. Elliptical slice sampling. In *AISTATS*, volume 13, pages 541–548, 2010.

Radford M Neal. Regression and classification using gaussian process priors. *Bayesian statistics*, 6:475, 1998.

Jeremy E Oakley and Benjamin D Youngman. Calibration of stochastic computer simulators using likelihood emulation. *Technometrics*, 59(1):80–92, 2017.

William L Oberkampf and Matthew F Barone. Measures of agreement between computation and experiment: validation metrics. *Journal of Computational Physics*, 217(1):5–36, 2006.

William L Oberkampf and Scott Ferson. Model validation under both aleatory and epistemic uncertainty. Technical report, Sandia National Laboratories (SNL-NM), Albuquerque, NM (United States), 2007.

William L Oberkampf and Christopher J Roy. *Verification and validation in scientific computing*. Cambridge University Press, 2010.

Ewan S Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954.

Martyn Plummer, Nicky Best, Kate Cowles, and Karen Vines. Coda: Convergence diagnosis and output analysis for mcmc. *R News*, 6(1):7–11, 2006.

Douglass E Post and Lawrence G Votta. Computational science demands a new paradigm. *Physics today*, 58(1):35–41, 2005.

Matthew T Pratola and David M Higdon. Bayesian additive regression tree calibration of complex high-dimensional computer models. *Technometrics*, 58(2):166–179, 2016.

Peter ZG Qian and CF Jeff Wu. Bayesian hierarchical modeling for integrating low-accuracy and high-accuracy experiments. *Technometrics*, 50(2):192–204, 2008.

Zhiguang Qian, Carolyn Conner Seepersad, V Roshan Joseph, Janet K Allen, and CF Jeff Wu. Building surrogate models based on detailed and approximate simulations. *Journal of Mechanical Design*, 128(4):668–677, 2006.

Peihua Qiu. *Introduction to statistical process control*. CRC Press, 2013.

Joaquin Quiñonero-Candela and Carl Edward Rasmussen. A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, 6(Dec):1939–1959, 2005.

Lawrence Rabiner and B Juang. An introduction to hidden markov models. *ieee assp magazine*, 3(1):4–16, 1986.

James O Ramsay and Xiaochun Li. Curve registration. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(2):351–363, 1998.

James O Ramsay. *Functional data analysis*. Wiley Online Library, 2006.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2006.

Ramesh Rebba and Sankaran Mahadevan. Computational methods for model reliability assessment. *Reliability Engineering & System Safety*, 93(8):1197–1207, 2008.

Ramesh Rebba, Sankaran Mahadevan, and Shuping Huang. Validation and error estimation of computational models. *Reliability Engineering & System Safety*, 91(10):1390–1397, 2006.

C Shane Reese, Alyson G Wilson, Michael Hamada, Harry F Martz, and Kenneth J Ryan. Integrated analysis of computer and physical experiments. *Technometrics*, 46(2):153–164, 2004.

Luis Miguel Rios and Nikolaos V Sahinidis. Derivative-free optimization: a review of algorithms and comparison of software implementations. *Journal of Global Optimization*, 56(3):1247–1293, 2013.

Christian P. Robert and George Casella. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

Christopher J Roy and Michael S Balch. A holistic approach to uncertainty quantification with application to supersonic nozzle thrust. *International Journal for Uncertainty Quantification*, 2(4), 2012.

Christopher J Roy and William L Oberkampf. A comprehensive framework for verification, validation, and uncertainty quantification in scientific computing. *Computer Methods in Applied Mechanics and Engineering*, 200(25):2131–2144, 2011.

Håvard Rue, Sara Martino, and Nicolas Chopin. Approximate bayesian inference for latent gaussian models by using integrated nested laplace approximations. *Journal of the royal statistical society: Series b (statistical methodology)*, 71(2):319–392, 2009.

Jerome Sacks, William J. Welch, Toby J. Mitchell, and Henry P. Wynn. Design and analysis of computer experiments. *Statist. Sci.*, 4(4):409–423, 1989.

Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, 26(1):43–49, 1978.

Shankar Sankararaman and Sankaran Mahadevan. Model validation under epistemic uncertainty. *Reliability Engineering & System Safety*, 96(9):1232–1241, 2011.

K Sargsyan, HN Najm, and R Ghanem. On the statistical calibration of physical models. *International Journal of Chemical Kinetics*, 47(4):246–276, 2015.

Terrance Savitsky, Marina Vannucci, and Naijun Sha. Variable Selection for Nonparametric Gaussian Process Priors: Models and Computational Strategies. *Statistical science : a review journal of the Institute of Mathematical Statistics*, 26(1):130–149, 2011.

Lifeng Shang and Antoni B Chan. On approximate inference for generalized gaussian process models. *arXiv preprint arXiv:1311.6371*, 2013.

Nickolay Smirnov. Table for estimating the goodness of fit of empirical distributions. *The annals of mathematical statistics*, 19(2):279–281, 1948.

Ralph C Smith. *Uncertainty quantification: theory, implementation, and applications*, volume 12. Siam, 2013.

Thad Starner and Alex Pentland. Real-time american sign language recognition from video using hidden markov models. In *Motion-Based Recognition*, pages 227–243. Springer, 1997.

Mervyn Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the royal statistical society. Series B (Methodological)*, pages 111–147, 1974.

Michalis K Titsias and Neil D Lawrence. Bayesian gaussian process latent variable model. In *International Conference on Artificial Intelligence and Statistics*, pages 844–851, 2010.

Chih-Fong Tsai, Yu-Feng Hsu, Chia-Ying Lin, and Wei-Yang Lin. Intrusion detection by machine learning: A review. *Expert Systems with Applications*, 36(10):11994–12000, 2009.

Ruey S Tsay. Outliers, level shifts, and variance changes in time series. *Journal of forecasting*, 7(1):1–20, 1988.

Shuchun Wang, Wei Chen, and Kwok-Leung Tsui. Bayesian validation of computer models. *Technometrics*, 2012.

Dongbin Xiu and George Em Karniadakis. The wiener–askey polynomial chaos for stochastic differential equations. *SIAM journal on scientific computing*, 24(2):619–644, 2002.

Jun Yuan and Szu Hui Ng. Calibration, validation, and prediction in random simulation models: Gaussian process metamodels and a bayesian integrated solution. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 25(3):18, 2015.

Ruoxue Zhang and Sankaran Mahadevan. Bayesian methodology for reliability model acceptance. *Reliability Engineering & System Safety*, 80(1):95–103, 2003.

Appendix A

Appendix 1: Generalized Calibration Sampling Details

This appendix outlines the details of the sampling discussed in Section 3.3.1. The parameters to update are:

- Correlation scale parameters for the emulator and discrepancy: $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$
- Precision parameters for the emulator and discrepancy: κ, κ_δ
- Calibration parameters: $\boldsymbol{\theta}$
- Likelihood parameters: $\boldsymbol{\omega}_o$
- Link-transformed latent mean values: $g(\mu(\mathbf{x}))$

A.1 Sampling $\beta, \gamma, \kappa, \kappa_\delta, \theta$

We use single-site Metropolis-Hastings to update the correlation scale parameters $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$, the precision parameters κ and κ_δ, and $\boldsymbol{\theta}$. We use a log-normal proposal distribution for all but $\boldsymbol{\theta}$ as they have support on positive reals. For $\boldsymbol{\theta}$, a normal proposal distribution is used. The scale of the proposal distribution is tuned to have an acceptance rate near 30% using trial runs of the sampler. The normal proposal for $\boldsymbol{\theta}$ with bounded support can lead to a large number of rejections if the posterior distribution for $\boldsymbol{\theta}$ is concentrated near a boundary of the prior support; however, the tuning largely mitigates this problem.

A.2 Sampling $g(\mu(\mathbf{x}))$

We sample $g(\mu(\mathbf{x}))$ using elliptical slice sampling proposed by (Murray *et al.*, 2010). The proposed latent mean vector is generated from a mixture of the current value, $g(\mu(\mathbf{x}))$, and a draw from the Gaussian process prior distribution, $\boldsymbol{\nu}$. Possible proposed vectors lie on an ellipse between the draw from the prior distribution and the current value, with the angle

ϕ along the ellipse determining how close the proposed vector is to the current vector. For $\phi = 0$, the proposal is equal to the current vector while for $\phi = \pi$, the proposal is equal to ν. Slice sampling is used to determine the angle parameter by proposing values from an iteratively shrinking the interval on the ellipse until a vector is accepted. Demonstration of detailed balance for the method and other details can be found in Murray *et al.* (2010). Pseudocode for the sampling is found below. We have found that initializing the sampler for a value of $g(\mu(\mathbf{x}))$ at the sample mean of all replicates at $\mathbf{x}$ worked well, after possibly adding a small constant to ensure the sampler was initialized on the proper support.

```
procedure EllipicalSliceSampler($\mathbf{g} = g(\mu(\mathbf{x}))$)
    Draw $\boldsymbol{\nu}$ from the Gaussian process prior distribution N(0,$\Sigma$)
    Draw a random uniform $u \sim$ Uniform$(0, 1)$
    Obtain log-likelihood acceptance threshold: thresh $= logL(\mathbf{g}) + log(u)$
    Draw an angle $\phi \sim$ Uniform$(0, 2\pi)$
    Set $\phi_{min} = \phi - 2\pi, \phi_{max} = \phi$
    repeat
        Propose $\mathbf{g}^* = \mathbf{g}\text{cos}(\phi) + \boldsymbol{\nu}\text{sin}(\phi)$
        Calculate log-likelihood for the proposal: $ll = logL(\mathbf{g}^*)$
        if $ll >$ thresh then
            Accept $\mathbf{g}^*$
        else
            if $\phi > 0$ then
                Set $\phi_{max} = \phi$
            else
                Set $\phi_{min} = \phi$
            Draw an angle $\phi \sim$ Uniform$(\phi_{min}, \phi_{max})$
    until Accept $\mathbf{g}^*$
    return $\mathbf{g}^*$
```

Appendix B

Appendix 2: Closed-form Solution For the Maximum Penalized Likelihood Value for CPM Scaling State Parameter

In this appendix we derive the closed form for the optimal ϕ_j in Section 5.3. The expected log likelihood, isolating terms with ϕ_j, is:

$$\sum_{k=1}^{K}\sum_{s=1}^{S}\sum_{i=1}^{N_k} p_{s,k,i}(x_{i,k} - z_s u_k \phi_s)^2/\sigma^2 + C,$$

with parameters as defined in Section 5.3 and the sum with s is summing over all hidden Markov states and C is a constant with respect to ϕ_j. Taking the derivative with respect to ϕ_j and setting it equal to zero, we obtain:

$$\sum_{k=1}^{K}\sum_{\{s:\phi_s=\phi_j\}}\sum_{i=1}^{N_k} p_{s,k,i} z_s u_k (x_{i,k} - z_s u_k \hat{\phi}_j)/\sigma^2 = 0$$

$$\sum_{k=1}^{K}\sum_{\{s:\phi_s=\phi_j\}}\sum_{i=1}^{N_k} p_{s,k,i} z_s u_k x_{i,k} - \hat{\phi}_j \sum_{k=1}^{K}\sum_{\{s:\phi_s=\phi_j\}}\sum_{i=1}^{N_k} p_{s,k,i}(z_s u_k)^2 = 0$$

$$\hat{\phi}_j \sum_{k=1}^{K}\sum_{\{s:\phi_s=\phi_j\}}\sum_{i=1}^{N_k} p_{s,k,i}(z_s u_k)^2 = \sum_{k=1}^{K}\sum_{\{s:\phi_s=\phi_j\}}\sum_{i=1}^{N_k} p_{s,k,i} z_s u_k x_{i,k}$$

$$\hat{\phi}_j = \frac{\sum_{k=1}^{K}\sum_{\{s:\phi_s=\phi_j\}}\sum_{i=1}^{N_k} p_{s,k,i} z_s u_k x_{i,k}}{\sum_{k=1}^{K}\sum_{\{s:\phi_s=\phi_j\}}\sum_{i=1}^{N_k} p_{s,k,i}(z_s u_k)^2}.$$

Appendix C

Appendix 3: Residual Adjustment for Heteroscedasticity

The following is the procedure proposed by Tsay (1988) to adjust the residuals for heteroscedasticity. We use this method to on the training residuals in Section 5.6.6.

1. Estimate the parameters of the CPM model with the training data and obtain filtered residuals, $\mathbf{r}$.
2. For each possible split of the series of residuals, with a minimum number of observations, h, in each split, calculate the ratio of variances of the two partitions.
3. Let λ_V be the maximum variance ratio between partitions. Compare λ_V to a specified threshold C.
    - If $\lambda_V < C$ then the algorithm terminates and the resulting residuals are returned.
    - If $\lambda_V \geq C$, let d_0 be the index of the split that maximized the variance ratio.
4. Adjust the residuals such that:
    - For $t < d_0$, the residuals are unadjusted; $r_t = r_t$.
    - For $t \geq d_0$, $r_t = \bar{r} + (r_t - \bar{r})/\sqrt{\lambda_V^*}$, where $\lambda_V^* = \lambda_V$ if the second split has the larger variance and $\lambda_V^* = 1/\lambda_V$ if the second split has the smaller variance.
5. Return to Step 2 using the adjusted residuals.

In the application, we used a minimum observations per split of 10 and a critical value $C = 3$.