Statistical Inference Utilizing Agent Based Models

by

Daniel Heard

Department of Statistical Science
Duke University

Date: ______________________________

Approved:

____
David Banks, Supervisor

____
Sayan Mukherjee

____
Jim Berger

____
Jim Moody

Dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in the Department of Statistical Science
in the Graduate School of Duke University
2014

Abstract

Statistical Inference Utilizing Agent Based Models

by

Daniel Heard

Department of Statistical Science
Duke University

Date: ______________________

Approved:

____
David Banks, Supervisor

____
Sayan Mukherjee

____
Jim Berger

____
Jim Moody

An abstract of a dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Statistical Science in the Graduate School of Duke University
2014

Abstract

Agent-based models (ABMs) are computational models used to simulate the behaviors, actions and interactions of agents within a system. The individual agents each have their own set of assigned attributes and rules, which determine their behavior within the ABM system. These rules can be deterministic or probabilistic, allowing for a great deal of flexibility. ABMs allow us to observe how the behaviors of the individual agents affect the system as a whole and if any emergent structure develops within the system. Examining rule sets in conjunction with corresponding emergent structure shows how small-scale changes can affect large-scale outcomes within the system. Thus, we can better understand and predict the development and evolution of systems of interest.

ABMs have become ubiquitous—they used in business (virtual auctions to select electronic ads for display), atomospheric science (weather forecasting), and public health (to model epidemics). But there is limited understanding of the statistical properties of ABMs. Specifically, there are no formal procedures for calculating confidence intervals on predictions, nor for assessing goodness-of-fit, nor for testing whether a specific parameter (rule) is needed in an ABM. Motivated by important challenges of this sort, this dissertation focuses on developing methodology for uncertainty quantification and statistical inference in a likelihood-free context for ABMs.

Chapter 2 of the thesis develops theory related to ABMs, including procedures for model validation, assessing model equivalence and measuring model complexity.

Chapters 3 and 4 of the thesis focus on two approaches for performing likelihood-free inference involving ABMs, which is necessary because of the intractability of the likelihood function due to the variety of input rules and the complexity of outputs. Chapter 3 explores the use of Gaussian Process emulators in conjunction with ABMs to perform statistical inference. This draws upon a wealth of research on emulators, which find smooth functions on lower-dimensional Euclidean spaces that approximate the ABM. Emulator methods combine observed data with output from ABM simulations, using these to fit and calibrate Gaussian-process approximations. Chapter 4 discusses Approximate Bayesian Computation for ABM inference, the goal of which is to obtain approximation of the posterior distribution of some set of parameters given some observed data.

The final chapters of the thesis demonstrates the approaches for inference in two applications. Chapter 5 presents models for the spread of HIV based on detailed data on a social network of men who have sex with men (MSM) in southern India. Use of an ABM allows us to determine which social/economic/policy factors contribute to the transmission of the disease. We aim to estimate the effect that proposed medical interventions will have on the spread of HIV in this community. Chapter 6 examines the function of a heroin market in the Denver, Colorado metropolitan area. Extending an ABM developed from ethnographic research, we explore a procedure for reducing the model, as well as estimating posterior distributions of important quantities based on simulations.

To Veronica & Monica Heard

Contents

List of Tables

List of Figures

List of Abbreviations and Symbols

Symbols

diag $(\cdot)$	A diagnoal matrix.
$\mathbb{E}(\cdot)$	Expected value of a random variable.
$\mathcal{N}(\mu, \sigma^2)$	A univariate normal distribution with mean μ and variance σ^2.
$\mathcal{N}_p(\mu, \Sigma)$	A p-dimensional normal distribution with mean μ and covariance matrix Σ.

Abbreviations

ABC	Approximate Bayesian Computation.
ABM	Agent Based Model(ling).
CANDID	Cell-phone Assisted Network Detection and Identification.
FMO	Fungal Meningitis outbreak.
GASP	Gaussian Process Response Surface.
IDMS	Illicit Drug Market Simulation.
MMSB	Mixed membership stochastic block model.
MSM	Men who have sex with men.
PCA	Principal Components analysis.
SVD	Singular Value Decomposition.

Acknowledgements

I would like to begin by thanking my parents, Margaret and Dupree Heard, and my brother, Peter, for your unwavering faith in me and for setting a positive example which I have striven to emulate. Thank you to my incredible wife, Veronica, for your support, hard work and sacrifices to help me be successful, and for being an amazing mother to Monica.

Thank you to my friends Shaun, Wakashan, Marc, Percy, Cory, Austyn, David, Delrik, Kody and others who, in one way or another, assisted me along my journey.

I want to thank the Department of Statistical Science at Duke for giving me the opportunity to study here. Thank you to David Banks, my thesis advisor and *consigliere* whose suggestions and advice were invaluable during this process. Thank you to Sayan Mukherjee, Jim Berger and Jim Moody for serving on my committee and asking thought-provoking questions to cause me a healthy amount of discomfort. Thank you Karen, Nikki and Anne for always making sure my paperwork was taken care of. Thank you to Georgiy Bobashev, Joey Morris and RTI International for the research opportunity and financial support. Thanks to the other DSS students who have accompanied me during my study, both to celebrate and commiserate with me: Tim, Tsuyoshi, Tommy, Nick, Mary Beth, Thais and Maria.

If I have made any contribution, either to the academic community or to the world at large, then, to paraphrase Malcolm X, all of the credit is due to God. Only the mistakes have been mine.

1

Introduction

1.1 Overview

Many methods exist for studying the development of systems. Often, these systems are quite complex, making it difficult to understand functions of agents within the system and their effects on the system as a whole. Agent Based Models (ABMs) are computational models used to simulate the behaviors, actions and interactions of agents (individuals or collective entities) within a system. The individual agents are autonomous, having their own set of rules which determines their behavior and how they develop within the system. These models allow us to observe how the simultaneous behaviors of individual agents affect the system as a whole and examine the resulting emergent structure. Thus, we can better understand and predict the evolution of systems of interest and the appearance of complex phenomena. In particular, ABMs are becoming a valuable tool for simulating and better understanding human systems (Bonabeau, 2002).

Three of the earliest examples of ABMs include von Neumann machines or cellular automata (Kemeny, 1955); John Conway's Game of Life (Gardner, 1970), which looked at the evoluation of a universe determined by its initial state in which cells interact with one another; and Thomas Schnelling's study of segregation (Schnelling, 1971). ABMs grew in popularity in the 1990s because of the ease of implementation that came with improvement of available computer technology. Social sciences began using ABMs to explore social phemomena over time and the growth of societies over time, notably Epstein and Axtell (1996). ABMs have been compared to other modelling approaches in various problem settings, e.g., Hooten and Wikle (2010). ABMs can be implemented in a wide variety of software, ranging from traditional statistical software to dedicated ABM simulation platforms (Railsback et al., 2006).

1.2 ODD Protocol

The ABM development process became more standardized with the publication of the ODD (Overview, Design concepts, and Details) Procotol by Grimm et al. (2006). This protocol is intended to make model descriptions more complete and more easily understood, primarily for academic literature. This, in turn, enables reproducbility of ABMs and addresses major concerns previously associated with ABMs (Lorek and Sonnenschein, 1999). An extensive discussion of ODD protocol can be found in Polhill et al. (2008), and a discussion of the growing use of the protocol is presented in Grimm et al. (2010). The protocol's basic structure is presented in Figure 1.2.

The 'Purpose' section explains what the model is intended to do and its general goals. The 'State Variables and Scales' section outlines the structure of the model, identifying all of the entities in the model, such as types of agents, spatial structure and other local and global variables. Additionally, the variables that determine the state of these entities at any given point during the simulation are specified. The 'Process Overview and Scheduling' section lists all processes that occur in the model and in what order they occur. This covers the hierarchy of agent behaviors, effects on the environment and associated updates to states of model entities. The 'Design Concepts' section describes the general concepts upon which the model is based.

Overview	Purpose
	State Variables and Scales
	Process Overview and Scheduling
Design Concepts	Design Concepts
Details	Initialization
	Input
	Submodels

FIGURE 1.1: The elements of the ODD protocol. The three categories on the left serve to explain the general structure of the protocol, not to describe the model. Model description following ODD protocol consists of the the seven elements on the right.

This is primarily to provide an understanding of why certain design decisions were made. Such concepts include a summary of emergent behavior, agent objectives, agent interactions and a description of agents' perceptions and thought processes, if any exist. The 'Initialization' section identifies how the model is started and often provides references to support initial values of variables. The 'Input' section describes any other external inputs to the model (such as time-series market data in financial models or temperature data for a climate model). The 'Submodels' section explains in detail the equations and algorithms used in the model, as well as defining any parameters included in the model.

While ABMs are becoming ubiquitous, there is still a good deal of theory that remains to be developed. In particular, there has yet to be a standard model validation protocol specifically for ABMs, taking into account the multiple levels of behavior being simulated. Additionally, there is a lack of theory regarding the rigorous comparison of two ABMs, specifically in the context of model equivalence and model complexity.

1.3 Approaches for Inference using ABMs

ABMs are of particular interest in statistics because we can use the inputs and outputs from these models as data to study complex relationships and, in a Bayesian context, perform statistical inference. Because of the variety of input rules and the complexity of outputs, the likelihood function for realistic ABMs is intractable; thus, any inference involving these models must be likelihood-free. The two likelihood-free methods of utilizing Agent Based models for statistical inference are emulation and Approximate Bayesian Computation (ABC).

1.3.1 Gaussian Process Emulators

Gaussian Process emulation deals with developing statistical models to approximate complex computer model output. This Bayesian approach has been implemented and studied widely, with some of the earliest work done by Anthony O'Hagan (1978) and Jerome Sacks et al. (1989). Recent advances have been made in the development and implementation of Bayesian computer model emulators (Lopes, 2011).

In many application of ABMs, the sophistication of the system being modeled leads to a complex model with detailed sets of agent rules. This model copmlexity can, naturally, lead to computationally burdensome simulations. As the number of ABM parameters grows, emulation becomes more useful, allowing ABM outcomes of interest to be predicted at settings without running the model at these settings. After specifying the set of parameters for the ABM and running it for a fixed set of inputs, a Gaussian Process Response Surface (GASP) will be fit to the ABM. The use of a Gaussian Pricess gives flexibility in modeling realizations which can be used to interpolate data points and make probability statements. Many variations on Gaussian Process emulation have been made which allow for their implementation in a broader class of problems with fewer assumptions needed.

1.3.2 *Approximate Bayesian Computation*

Approximate Bayesian Computation (ABC) is a class of methods that allows for approximate computation in the analysis of complex models. For problems in which the likelihood function is intractable (such as ABMs) or very expensive to compute, ABC allows us to perform statistical inference by simulating from an approximation to the posterior distribution. The approach involves sampling a parameter set θ from a prior distrubition and generating data y conditional on the sampled parameters. If the generated data is close enough (according to some appropriate distance metric) to the observed data, then the sampled θ is accepted as a draw from the posterior. Some of the earliest ABC methods were introduced by Tavare et al. (1997) and Pritchard et al. (1999) for applicatioins in genetics. Many extensions have been made since, such as Marjoram et al. (2003), Toni et al. (2009), Beaumont et al. (2009) and others. Further developments of ABC in complex dynamic systems (cf. Bonassi, 2013) are still being made. ABC has been applied to reinforcement learning problems (cf. Dimitrakakis and Tziortziotis, 2013), which deals with similar issues to those that arise with ABMs.

For ABMs, we can identify model parameters in which we are particularly interested. This will serve as the θ for which shall generate an approximate posterior distribution. This approach gives a straightforward method of inference about agent behaviors contributing to system development. This inference allows us to identify reasonable constraints on agent behavior in a more rigorous manner than previously possible.

1.4 Dissertation Outline

This dissertation analyzes multiple approaches for the development and implementation of statistical inference utilizing ABMs. In chapter 2, I present theory which addresses some of the areas related to ABMs which have not been thoroughly investigated. I propose a model validation protocol as well as examining methods for model comparison in terms of complexity and determining model equivalence. In chapter 3, I discuss the utility of Gaussian Process emulators for inference, examining multiple approaches for different problem settings. Chapter 4 analyzes Approximate Bayesian Computation techniques for assessing sets of parameters and rule sets for ABMs and performing posterior inference. Chapters 5 and 6 look at ABM applications which demonstrate both emulator and ABC techniques and examine the utility of ABMs in conjunction with other methods for analysis of HIV transmission in a network in southern India and the dynamics of a heroin market.

2

ABM Theory

2.1 Overview

ABMs are often used to simulate complex real world processes and in many cases are used for qualitative insight. A limitation in the use of ABMs has been the disconnect between existing theory, both in a mathematical sense and in the context of many theories of social behavior, discussed in detail in Chattoe (2003). To truly leverage ABMs and be able to explore 'what if' questions related to systems of interest, one must be precise in model specification to ensure it correctly simulates the system. Additionally, one should be able to quantitatively compare models for a given system based on various criteria in order to better understand system behavior and make determinations about the inclusion and treatment of model elements.

To this end, model validation is a crucial component in the development of ABMs. This procedure ensures that the dynamics being simulated in the model are a reasonable representation of the system and that the model itself is correctly capturing large-scale system behavior.

When considering multiple models for a system, there are certain quantitative comparisons which it is important to be able to make. Identifying whether models are equivalent (in some sense) can be helpful in examining treatment of different variables and implementing the agent rule sets. Determining the relative complexity of two models can be a useful step in model selection for ABMs.

2.2 ABM Validation

Validating ABMs is an important component of model development. A general approach for validation and verification of computer models is presented in chapter 3. Although validation of ABMs has some elements in common with validation of more traditional computer models, the process for ABMs is slightly different because the aggregate emergent structure must be considered in tandem with agent-level parameters and rules. The Virtual Overlay Multi-Agnet System (VOMAS) verification and validation technique for ABMs is based in software engineering (Niazi et al., 2009). In this approach, a VOMAS is developed along with the ABM, in which the agents gather data through logs, providing run-time support of the validation process by checking for violations of user-specified settings. While the VOMAS approach is a thorough approach, it can become cumbersome in many cases, as it requires the development of two models. Some other work has been done exploring model validation strategies specifically for ABMs (Windrum et al., 2007; Fagiolo et al., 2007; Marks, 2012, 2013), however, it is very much an open problem.

Here, I propose a model validation protocol for ABMs and discuss it in relation to a model I developed for the multi-state fungal meningitis outbreak (FMO) of 2012-2013 within the state of Michigan (Centers for Disease Control & Prevention, 2013b).

In 2012, the New England Compounding Center (NECC) distributed contaminated lots of Methylprednisolone Acetate for steriod injections to health care facilities in multiple states. Michigan had the highest incidence rate, with 264 cases and 19 deaths as of October 2013. A map of the facilities in the state is shown in Figure 2.2. Details of nationwide cases and facilities are presented in appendix A.

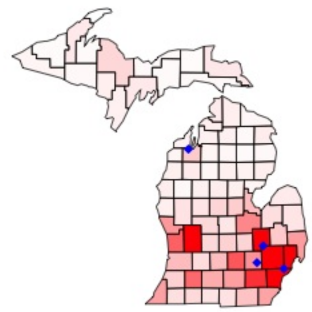


Figure 2.1: Map of Michigan population density by county with blue diamonds representing locations of the four facilities receiving contaminated Methylprednisolone Acetate.

The output quantity of interest for this model is the number of cases of infection at a given time. We have data on the number of cases, taken at 52 time points between October 2012 and August 2013, as well as information on the specific products and quantities received by the facilities, as well as shipping dates from NECC which provide an estimate of when initial infections could have occured (US Food and Drug Administration, 2012).

2.2.1 Internal Validation

Internal validation involves examining components within the model in order to identify potentially problematic areas, as well as to determine if the model is operating in a way that is consistent with the system it represents.

Step 1: Assess Face Validity

Assessing face validity of a model involves determining if the model's structure and behavior are reasonable according to domain experts. This involves examination of rule sets for individual agents as well as the dynamics of the interactions of agents with each other and the system itself. Bharathy and Silverman (2010) and Xiang et al. (2005) discuss some specific techniques for evaluating model face validity. In the case of the heroin market ABM (discussed in chapter 6), the face validity of many of the model's functions follows from the fact that it was developed from the ethnography of a domain expert.

For the FMO model, the central dymanic is the means by which the disease is spread. Here, the method by which agents contract the disease differs from traditional disease spread models because fungal meningitis is not contagious, as infection is only obtained through direct introduction into the blood stream. Thus, disease is passed from facilities to individual patients and not spread from patient to patient. While the locations of the facilities which received contaminated substances were identified, the number of cases for which the individual facilities were responsible was not identified. Demographic information for infected individuals (age, gender, race, etc.) was not provided either. To account for this, the affected facilities' weekly visitor counts were determined in part by population density within 100 miles of each facility, based upon US Census estimates (US Census Bureau, 2013).

Additionally, an agent's age influenced their probability of visiting one of the affected facilities, based on available information on visitor demographics in 2012 for similar facilities in the state (Michigan Headache & Neurological Institute, 2012). Out of 4606 annual visitors, 2.3% were age 6-17, 18.5% were age 18-40, 65.3% were age 41-65, 13.4% were age 66-85 and 0.5% were age 86 and older.

Once an agent visits an affected facility, the probability of receiving a tainted injection and contracting fungal meningitis was determined by an exponential model for modeling population growth (Brauer and Castillo-Chavez, 2012; Diekmann and Heesterbeek, 2000). The standardized observed case counts approximately fit the curve

$$F(x) = 1 - \exp(-0.14x + 0.08)$$

where x is the week (with week 1 representing October 6, 2012). This model was chosen to capture the pattern of the outbreak, with decreasing numbers of case counts over time as the tainted materials are identified and removed. Because of the importance of the time component in the outbreak, the above model was more appropriate for this application than a compartmental model (SIR, SEIR, etc.). This model fit well, with a sum of squared residuals over the first 14 weeks of 9.43. Differentiating this curve gives a function of the form of a scaled exponential distribution probability distribution function,

$$f(x) = 0.14 \exp(-0.14x + 0.08)$$

which can be normalized to give a proper pdf.

Because symptoms of infection were reported to appear 1 to 4 weeks after receiving a contaminated injection (Centers for Disease Control & Prevention, 2013a), there is some uncertainty in the time between infection and case confirmation. To account for this and the lack of any additional information, based on the principle of maximum entropy, the time for an infected agent to be confirmed as a case was drawn from a discrete uniform (1,4).

A main emphasis of assessing face validity is understanding specific dynamics of the system being modeled and not merely beginning with an off-the-shelf model which may represent a system that functions quite differently from the system of interest.

Step 2: Determine Evaluation Criteria

Output from ABMs can be complex and multidimensional. While all of the outputs may be related, it is possible that only a small subset are of primary interest when simulating the system. In the case of ABMs for networks, the are several different network measures (e.g. degree distribution) that could be of primary interest, but other associated values (e.g. density, clustering) are produced in the model output. Additionally, one should establish the range of inputs over which output evaluation is sought.

In the case of the FMO model, we are interested in the count of infections at time points throughout the simulation. We have data on case counts to which we will compare model simulation to evaluate the quality of the model's representation of this outbreak. Because we have the relative proportion of patient visits by age to facilities similar to those affected in the outbreak, there is an estimate of on individual agent's probability of visiting the facilities; but, because this is not exact, it is necessary to simulate behavior with a range of probabilities around these estimates.

2.2.2 External Validation

External validation involves comparing elements of the model to other sources. This component is significant in that it goes beyond determining the logical justifiability of the model to obtain a quantitative determination of how well the ABM represents the system.

Step 3: Output Analysis

The general strategy for external validation is the analysis of some output quantity. This approach can be separated into a number of more specific techniques which are discussed below.

Predictive Validation

In cases where longitudinal data is available, it is possible to develop the model using data only up to a certain time point, and then forward simulations can be checked against the subsequent data to determine the predictive accuracy of the model.

Because longitudinal data were available for the Meningitis outbreak, predictive validation was used for the FMO model. The model begins simulation in September 2012 and was built by fitting weekly data from October 2012 through December 2012 and then validated by comparing simulated case counts in the model to the observed case counts for January 2013 through October 2013. Case counts were provided at smaller time intervals in October 2012 when the outbreak was first identified, and then the intervals grew longer in December 2012, so some interpolation for case counts was necessary in the predictive validation. Because of the uncertainty in the time between infection and confirmation of fungal meningitis, as discussed earlier, the criterion for model agreement with the actual outbreak was the predicted model case counts falling within the range of observed case counts in a four week window.

Cross-model Validation

In many cases, different conceptual models can be used to simulate a particular phenomena. Cross-model validation leverages these instances and compares results of the ABM simulation to results from other models. This approach is useful in that it allows both qualitative and quantitative comparisons. In particular, when comparing a particular output quantity of an ABM to a different model which itself has been verified and validated, agreement of the two models strongly implies validation of the ABM. Axtell et al. (1996) discusses strategies for cross-model validation in detail.

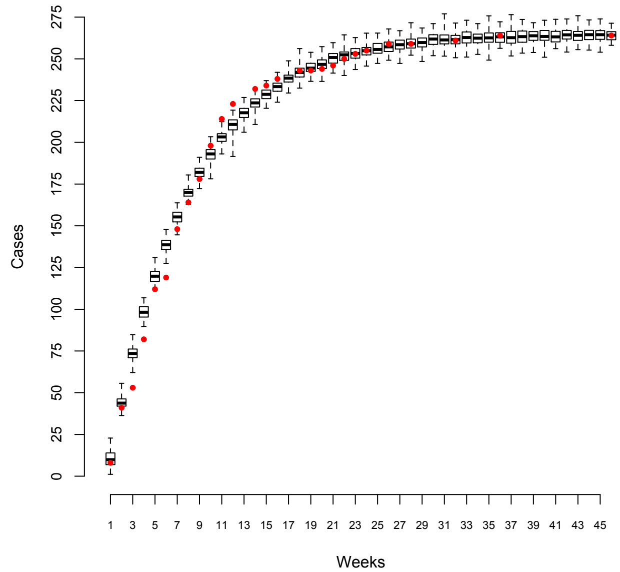
When validating an ABM with another model which has already been validated, one may seek to assess equivalence between the models. Depending on the features of the respective models, there are different notions of equivalence which can be considered. Some possible model equivalence measures are discussed in section 2.3.

Regarding the fungal meningitis outbreak, while there were models developed to simulate the biological effects of the outbreak on individuals and treatment strategies (Pappas et al., 2013), there were no models developed specifically to simulate how the outbreak progressed. The closest validated model to which to compare the FMO model is an ABM for contamination events (Zechman, 2011).

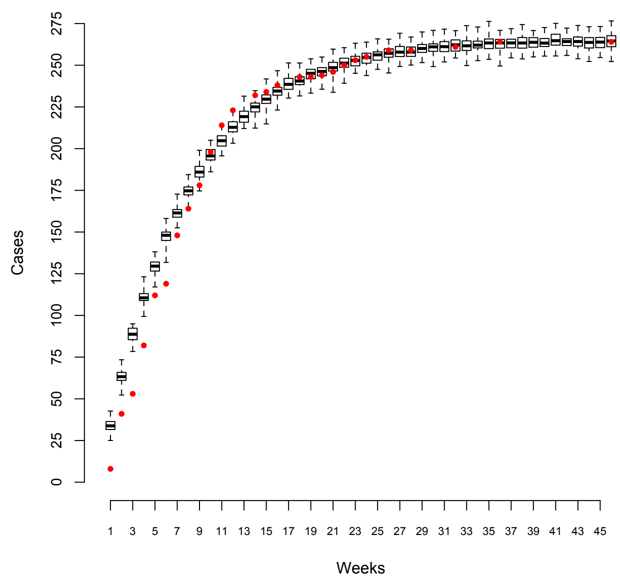
The Zechman model was developed in AnyLogic (XJ Technologies, 2013) to simulate response of indivudals to water contamination, incorporating a spatial component based on the proximity of individuals to the location of contamination, as well as timing and communcation which alters individuals' water consumption and influences change of use in response to the contamination. The simulation incorporates the EPANET water distribution system model (Shang et al., 2008).

By making adjustments to the Zechman model such as reducing the number of nodes (water sources) within the network, treating all nodes as commercial (to simulate a high number of visitors), and restricting the consumer demand at the nodes, the model's operation closely resembles that of the FMO model. The Zechman model runs on a finer-scaled time step than the FMO model, which must also be taken into account. The simulations under the framework of the Zechman model give case counts consistent with those from the FMO model. While both models give average simulated case counts which capture actual case counts within a four week window, the Zechman model over-predicts cases early in the simulation (October 2012). A comparison of the simulation results is shown in Figure 2.2.

In addition to the comparison of the simulated case counts of the two models, the FMO model was found to be a less complex model than the Zechman model for the outbreak (see discussion of model complexity in section 2.4).



(a)



(b)

FIGURE 2.2: Comparison of simulated fungal meningitis case counts from (a) the FMO model and (b) the Zechman model from October 2012 through August 2013 based on 100 simulations from each model. Red points represent actual case counts from the outbreak.

Statistical Validation

Statistical validation is the most rigorous external validation method and can significantly increase the credibility of an ABM and can be performed in conjunction with predictive or cross-model validation. Hypothesis tests for equivalence of ABM output with either true system measurements or output of other models are a useful approach. Additionally, based on the data being used to validate the model, one can establish tolerance bounds for model output. These represent an acceptable range of model output values which can be considered to represent reasonable system behavior. One can then determine an acceptable proportion of output values which should fall within the tolerance bounds to evaluate model performance. Additionally, tolerance bounds allow for varying degrees of accuracy in different model applications, as well as different degrees of uncertainty at different input settings. Discussion on applying statistical techniques to model validation in specific settings can be found in Kleijinen (1999) and Sanchez (2001).

A more rigorous and involved technique of statistical validation of ABMs involves approximating the model using Gaussian process techniques, as discussed in chapter 3.

Step 4: Feedback/feed forward

Model validation is very much an interative process, and the final step of the procedure involves using the findings of previous model validation steps to make adjustments to the model.

2.3 Model Equivalence

Given the variety of strategies for developing ABMs, one important topic to consider is that of model equivalence, as determined based upon some model output quantity. There has been some exploration of model equivalence related to model validation in Robinson et al. (2005) and Yang et al. (2004), as well as discussion of the issues involving hypothesis testing as it relates to model equivalence in Welsh (1996) and Berger (2003) among others.

There are several topics to consider regarding separate models for a system and making comparisons among them. An important area which must be considered is that of the input sets/spaces of the models. These sets will fall into one of four possible categories: 1) The sets are the same, 2) There is partial overlap in the sets, 3) One set contains the other, 4) The sets are disjoint. The criteria for determining model equivalence is largely dependent upon the category into which the input sets fall.

When looking to assess model equivalence, there are three main types of equivalence one could seek to identify. We restrict the initial discussion of model equivalence to the regions of the input space that both models have in common. We will discuss the assessment of equivalence of two models, but the theory can be extended to consider larger collections of models.

For the subsequent discussion, let the two models be represented as maps $f_1 : \mathcal{S}_1 \subseteq \mathbb{R}^p \longrightarrow \mathcal{A}_1 \subseteq \mathbb{R}$ and $f_2 : \mathcal{S}_2 \subseteq \mathbb{R}^p \longrightarrow \mathcal{A}_2 \subseteq \mathbb{R}$. We begin by considering the case where $\mathcal{S}_1 = \mathcal{S}_2 = \mathcal{S}$ and $\mathcal{A}_1 = \mathcal{A}_2 = \mathcal{A}$.

Equivalence in mean

Two models are equivalent in mean if, given a fixed set of inputs, the mean functions of the output are equivalent. ABMs need not be deterministic, a fixed set of inputs can produce significant output variation from simulation to simulation for a given model. So, while exact equivalence (two models producing the exact same output for a fixed set of inputs) is rare in practice, equivalence in mean looks at the average of some output value of multiple simulations of the two models. Although the maps f_1 and f_2 are not random, for a given point $x_0 \in \mathcal{S}$, we can define a probability space $(\Omega, \mathcal{F}, \mathsf{P})$ where $\Omega = \mathcal{A}$, $\mathcal{F}$ is the Borel sets on $\mathcal{A}$ and, under the probability measure P, $\mathsf{P}(A \subseteq \mathcal{A})$ is the probability that a simulation run at input settings x_0 will have an output value in A, which can be estimated based on a large-scale batch of simulations. We can consider the mean function to be the expected value of the simulation using the specified probability space.

Equivalence in mean modulo monotonicity

Equivalence in mean modulo monotonicity is a slightly relaxed version of equivalence in mean where, given two models, there exists a monotone (or anti-monotone) function $m : \mathcal{A} \longrightarrow \mathcal{A}$ such that $m \circ f_1 = f_2$. In this case, for a fixed set of inputs $\mathbf{x} \in \mathcal{S}$, there exists a monotone function which maps the mean function of one model to the mean function of the other model. The monotone function can be viewed as a calibration function.

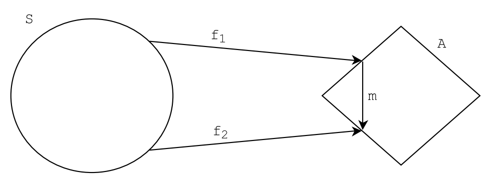


Figure 2.3: An illustration of equivalence in mean modulo monotonicity.

As an example, consider two ABMs for weather forecasting, in which the agents are cubic kilometers of atmosphere and they interact by exchanging pressure, temperature and moisture. Model 1, for a fixed set of inputs, could have temperature predictions of 65°F, 70°F and 75°F, while model 2 could have temperature predictions of 80°F, 85°F and 90°F. If this is the case for all sets of inputs, we can establish a monotone function mapping the output of model 1 to the output of model 2.

If the output space $\mathcal{A} \subset \mathbb{R}$ is connected, then we can consider this equivalence in an alternative way. By the connectedness of $\mathcal{A}$, f_1 and f_2 are homotopic (i.e. there exists a family of continuous functions $h_t : \mathcal{S} \to \mathcal{A}$ for $t \in [0,1]$ such that $h_0 = f_1$ and $h_1 = f_2$ and the map $(\mathcal{S}, t) \longmapsto h_t(\cdot)$ is continuous from $\mathcal{S} \times [0,1]$ to $\mathcal{A}$.) One can trivially define $h_t = (1-t)f_1 + tf_2$. If h_t is monotone in t, then this homotopy formulation can serve as a surrogate for the m function discussed above.

Equivalence in distribution

Two models are equivalent in distribution if the distribution of the outputs is the same for the models given a fixed set of inputs. Equivalence in mean (or equivalence in mean modulo monotonicity) can be achieved by models which have outputs that differ greatly at each simulation. Consider again two weather forecasing ABMs. Model 1, for a fixed set of inputs, could have temperature predictions ranging between 65°F and 75°F, while model 2 could have temperature predictions ranging between 40°F and 100°F. While both would have the same mean, the variances differ greatly.

The notion of equivalence in distribution takes variation of the outputs into consideration. While two models may not have outputs which match exactly, having the same distribution of outputs demonstrates a high-level agreement in the predicted behavior of the system. Certain tests for equivalence such as a Kolmogorov-Smirnov test and statistical measures such as Mahalanobis distance enable the assessment of equivalence in distribution. A detailed discussion of such tests is presented in Welleck (2010).

Theorem 1. *(Topological Equivalence of Models) Let $\mathcal{S} \subset \mathbb{R}^p$ be compact. Let $f_1 : \mathcal{S} \longrightarrow \mathcal{A} \subseteq \mathbb{R}$ and $f_2 : \mathcal{S} \longrightarrow \mathcal{A} \subseteq \mathbb{R}$ be continuous maps, both of which represent ABMs of a particular system. Suppose f_1 and f_2 each have a finite number of critical points (points where the derivative vanishes), denoted y_{1i} and y_{2i}, respectively (by the compactness of $\mathcal{S}$, this is equivalent to the critical points being isolated). The models f_1 and f_2 are topologically equivalent if and only if the atoms of their critical values are isomorphic.*

For clarity, I define some terms before presenting a proof.

Two maps are topologically equivalent if there exist two homeomorphisms $h : \mathcal{S} \longrightarrow \mathcal{S}$ and $h' : \mathcal{A} \longrightarrow \mathcal{A}$ such that $f_1 h = h' f_2$.

Let $f : \mathcal{S} \longrightarrow \mathbb{R}$ be a continuous map with a critical value c. If critical level set $f^{-1}(c) \subset \mathcal{S}$ contains only finitely many critical points, then a connected component Γ of $f^{-1}(c)$ is called an atom of the critical value c of the function f.

Proof of Theorem 1: This proof closely follows work presented in Sharko (2003). *Necessity.* Suppose f_1 and f_2 are topologically equivalent. Then there exists a homeomorphism h on $\mathcal{S}$ mapping the critical level sets of f_1 to the critical level sets of f_2. The continuity of h ensures that connected components of the critical level sets are mapped to each other. Hence, the atoms of the critical values of f_1 and f_2 are isomorphic.

Sufficiency. Let ϕ be an isomorphism between the atoms of the critical values of f_1 and f_2. By a theorem presented in Prishlyak (2002), because of the continuity of the maps, each isolated critical point of f_1 and f_2 has a closed neighborhood in which the map is topologically equivalent to a polynomial of the form $x^n + c$ for some non-negative integer n and some constant c. Let $Cr_i(f_1)$ and $Cr_i(f_2)$ denote these neighborhoods of the critical points. Following Sharko (2003), using ϕ, we can construct homeomorphisms h_i mapping $Cr_i(f_1)$ to $Cr_i(f_2)$. By Chapman (1972) and Anderson (1967), we can extend the homeomorphisms h_i to a homeomorphism h_1 defined on closed neighborhoods of curves that join critical points of f_1, denoted $U_i(f_1)$, and maps these neighborhoods to $U_i(f_2)$, the corresponding neighborhoods of the critical points of f_2. One can choose h_1 in such a way that it maps the level curves of f_1 in $Cr_i(f_1) \cup U_i(f_1)$ to the level curves in $Cr_i(f_2) \cup U_i(f_2)$. By construction, the closure of the complement of $(Cr_i(f_1) \cup U_i(f_1)) \bigcup (Cr_i(f_2) \cup U_i(f_2))$ in $\mathcal{S}$ will consist of the union of sets homeomorphic to cylinders (or hypercylinders depending on the dimension of $\mathcal{S}$), as $Cr_i(f_1) \cup U_i(f_1)) \bigcup (Cr_i(f_2) \cup U_i(f_2)$ is closed as the union of closed sets and any separated sets which make up this complement will be homeomorphic to (hyper-)cylinders (Salzmann, 1969). By Chapman (1972) and Anderson (1967), we can again extend the homeomorphism h_1 to these (hyper-)cylinders so that the level sets of f_1 map to the level sets of f_2. Identifying an appropriate homeomorphism h on $\mathbb{R}$, we obtain topological equivalence of f_1 and f_2. □

With respect to the ABMs represented by f_1 and f_2, the conjugate homeomorphism h and h' can be considered as a tuning function and a calibration function, respectively.

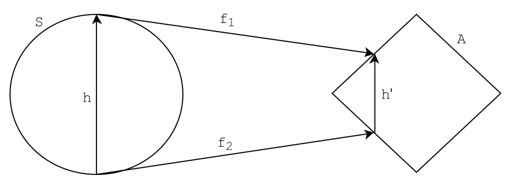


FIGURE 2.4: An illustration of topological equivalence of maps.

It is appropriate to note here that in the case where $\mathcal{S}_1 = \mathcal{S}_2$ but, without loss of generality, $\mathcal{A}_1 \subset \mathcal{A}_2$, it follows that the model f_1 is a proper subset of f_2. In a model selection scenario, this would make f_2 a more favorable choice, unless there were other issues, such as model complexity (discussed in section 2.4), that favored f_1. This case would then require some decision rule for a model selection.

2.3.1 Non-equivalent Input Spaces

The above discussion considered models for which the input spaces were the same. However, in many cases, different models for the same system may use different sets of inputs and have input spaces which are not the same. Here, we examine the three cases where $\mathcal{S}_1 \neq \mathcal{S}_2$ and identify conditions for model equivalence in each of these cases.

Intersection of Input Spaces

The first possible scenario for non-equivalent input spaces is the case where $\mathcal{S}_1$ and $\mathcal{S}_2$ partially overlap so that there exist $\mathbf{x}_1 \in \mathcal{S}_1 \backslash \mathcal{S}_2$ and $\mathbf{x}_2 \in \mathcal{S}_2 \backslash \mathcal{S}_1$ while $\mathcal{S}_1 \cap \mathcal{S}_2 \neq \varnothing$.

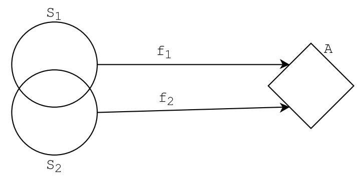


FIGURE 2.5: An illustration of equivalence in mean modulo monotonicity.

In order to establish equivalence in mean, equivalence in mean modulo monotonicity, or equivalence in distribution of f_1 and f_2, one must first be able to establish a homeomorphism h (tuning function) between $\mathcal{S}_1$ and $\mathcal{S}_2$. If this can be done, then equivalence of each type can be determined as described in Section 2.2. If the homeomorphism h maps the atoms of the critical values of f_1 and f_2 to one another, then the models are topologically equivalent by Theorem 1. None of the forms of equivalence discussed above require the two models to have the same behavior on the set $\mathcal{S}_1 \cap \mathcal{S}_2$.

If we look at the restrictions $f_1|_{\mathcal{S}_1 \cap \mathcal{S}_2}$ and $f_2|_{\mathcal{S}_1 \cap \mathcal{S}_1}$, it is possible to establish equivalence of f_1 and f_2 on the set $\mathcal{S}_1 \cap \mathcal{S}_2$. For equivalence in mean, equivalence in mean modulo monotonicity, and equivalence in distribution, one would only need to look at the behaviors of the maps restricted to this set. To establish topological equivalence on the set, one could look at the atoms of critical values of the restricted maps and, if they are isomorphic, then the maps are topologically equivalent when restricted to the intersection of the input spaces by Theorem 1. In we find $f_1|_{\mathcal{S}_1 \cap \mathcal{S}_2}$ and $f_2|_{\mathcal{S}_1 \cap \mathcal{S}_2}$ to be equivalent, then f_1 and f_2 are demonstrate partial equivalence. If the two models demonstrate are not equivalent over their entire input spaces, but are partially equivalent on $\mathcal{S}_1 \cap \mathcal{S}_2$, this can inform the nature of the discrepancy between the two models.

Containment of Input Spaces

The second possible scenario for non-equivalent input spaces is the case where, without loss of generality, $\mathcal{S}_1 \subset \mathcal{S}_2$.

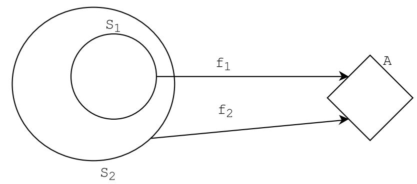


FIGURE 2.6: An illustration of containment of input spaces.

In order to establish equivalence in mean, equivalence in mean modulo monotonicity, equivalence in distribution or topological equivalence for this case, the procedure is essentially the same as the case where $\mathcal{S}_1$ and $\mathcal{S}_2$ partially overlap.

If we look at the restriction $f_2|_{\mathcal{S}_1}$, it is possible to establish equivalence of f_1 and f_2 on $\mathcal{S}_1$. In we find f_1 and $f_2|_{\mathcal{S}_1}$ to be equivalent, then f_1 and f_2 are partially equivalent, as they are satisfy conditions for equivalence on the entire input space of f_1. If this is the case, the behavior of f_2 on the set $\mathcal{S}_2 \backslash \mathcal{S}_1$ determines if the models fully satisfy the conditions for equivalence.

If we find that $f_2(\mathcal{S}_2 \backslash \mathcal{S}_1) \subseteq f_2(\mathcal{S}_1)$, then f_2 is a more complex model in the Kolmogorov sense than f_1 (see discussion of model complexity in section 2.4) and f_1 is a more parsimonious representation of the system of interest.

If there exists $y \in f_2(\mathcal{S}_2 \backslash \mathcal{S}_1)$ such that $y \notin f_2(\mathcal{S}_1)$, then f_2 is capable of capturing behavior beyond that which f_1 can, and f_1 is a proper subset of f_2.

Disjoint Input Spaces

The third possible scenario for non-equivalent input spaces is the case where $\mathcal{S}_1 \cap \mathcal{S}_2 = \varnothing$.

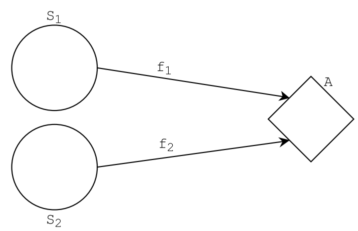


FIGURE 2.7: An illustration of disjoint input spaces.

In order to establish equivalence in mean, equivalence in mean modulo monotonicity, equivalence in distribution or topological equivalence for this case, the procedure is essentially the same as the other cases of non-equivalent input spaces. In general, it is unlikely that two models with completely disjoint input spaces (i.e. having no inupt variables in common) will be equivalent.

For any set of non-equivalent input spaces, in the case where $\mathcal{S}_1$ and $\mathcal{S}_2$ are of different dimensions (i.e. f_1 has more input variables than f_2), the two models cannot be topologically equivalent, since one cannot establish a homeomorphism between $\mathcal{S}_1$ and $\mathcal{S}_2$. Letting $\mathcal{S}_1 \subset \mathbb{R}^m$ and $\mathcal{S}_2 \subset \mathbb{R}^n$ for $m \neq n$, since $\mathcal{S}_1$ and $\mathcal{S}_2$ are simply connected by construction, the (open) interiors of $\mathcal{S}_1$ and $\mathcal{S}_2$ are homeomorphic to $\mathbb{R}^m$ and $\mathbb{R}^n$, respectively by the Riemann mapping theorem. As a result of invariance of domain (Brouwer, 1912) and the Jordan-Brouwer separation theorem (Alexander, 1922), $\mathbb{R}^m$ and $\mathbb{R}^n$ are not homeomorphic. Hence, $\mathcal{S}_1$ and $\mathcal{S}_2$ cannot be homeomorphic, otherwise one could construct a homeomorphism between $\mathbb{R}^m$ and $\mathbb{R}^n$ by composition.

2.4 Model Complexity

A related concept to model equivalence is that of model complexity. Some work has been done on quantifying the complexity of statistical models (Vanpaemel, 2009; Spiegelhalter et al., 2002), but there has been limited exploration of this topic as related to ABMs. A general discussion of the complexity of simulation models as well as advantages and disadvantages of increasing the level of detail of a simulation model is presented in Chwif et al. (2000). A major limitation at this point is that there is no widely accepted definition of what a complex model is, nor is there any general complexity measure of a given simulation model. When comparing two models, one may have an intuition as to when one model is more complex than another (specfically, this occurs when one creates a 'reduced' or 'simplified' version of an ABM, as described in chapter 6). In these cases and in less straightforward cases where two different models exist for the same system, having some metric for comparing the complexity of the models can be useful. Here, we propose two measures of complexity for simulation models.

A naive method for comparing model complexity is by comparing model run-time. For two models of a system with the same time step, the model which takes longer to simulate a fixed period of time is, in a sense, more complex than another.

The notion of Kolmogorov complexity (Gammerman and Vovk, 1999; Li and Vitaanyi, 2008) identifies the complexity of an object as the shortest program written in a fixed language which can produce the object. Given two models for a system in a particular language, one could use this concept to compare the complexity of two models using length of the programs. There is also the ability to incorporate an interpreter, which enables models to be translated between programming languages and, hence, the comparison of models in different languages.

Based on results proven in Thomas (1991), one can place an upper bound on the

Kolmogorov complexity of a model, which is useful for comparing the complexity of models for which the codes have similar structure.

A more formal approach for assessing model complexity is based in the information theoretic approach of model compression. Several compression algorithms exist, but the two most prominent lossless compression algorithms are Huffman coding and Lempel-Ziv coding (Hufman, 1952; Ziv and Lempel, 1977, 1978). One can then use the compression ratio ($\frac{\text{uncompressed model size}}{\text{compressed model size}}$) as a measure model complexity, as proposed by Khalatur et al. (2003) and Evans and Bush (2001). Considering model complexity in this way, the model with a higher compression ratio is the more complex model. An example of this approach is presented in chapter 6.

2.4.1 Irrelevant rules

Here, we make a proposition regarding the inputs for an ABM and examine its implications when applied to a model.

Proposition 2. *Inclusion of an 'irrelevant' rule will not affect the emergent structure of an ABM.*

To begin, one must define the emergent structure of interest for the model. Because the mechanisms of ABMs can allow a rule/input to affect different output and model behavior, we refer to a rule as 'irrelevant' with respect to a specific emergent structure.

One can perform principal component analysis (PCA) (Jolliffe, 2005; Jackson, 1991; Forzani, 2006) on the set of inputs (possibly agent rules, if they can be represented as continuous parameters) and regress the emergent behavior based on the principal components (Jolliffe, 1982; Martens and Naes, 1989; Mevik and Wehrens, 2007).

Defining $\mathbf{Y}$ as an $n-$dimensional measure of the emergent behavior of interest, and $\mathbf{X}$ as the $n \times p$ set of inputs, then, instead of least squares regression:

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon,$$

one first performs PCA on the (scaled) inputs $\mathbf{X}$ using the singular value decomposition $\mathbf{X} = \mathbf{U}\Delta\mathbf{W}^T$ where Δ is a diagonal matrix containing the non-negative singular values of $\mathbf{X}$, and the columns of $\mathbf{U}$ and $\mathbf{W}$ are both sets of orthonormal vectors, which are the left and right singular vectors of $\mathbf{X}$. The quantity $\mathbf{W}\Delta^2\mathbf{W}^T$ gives a spectral decomposition of $\mathbf{X}^T\mathbf{X}$. The score matrix $\mathbf{T}$ can be written as $\mathbf{T} = \mathbf{X}\mathbf{W}$, where $\mathbf{W}$ is the loadings matrix and the jth column of $\mathbf{T}$ gives the jth principal component of $\mathbf{X}$. The principal component regression takes on the form

$$\mathbf{Y} = \mathbf{T}\gamma + \epsilon.$$

With the aim of excluding components that are not significant in explaining the emergent behavior of the model while simplifying analysis, Massy (1965) proposes two criteria for deleting components. The first method is to delete components with eigenvalues below some specified cutoff, λ_0, as they are unimportant as predictors of the original inputs, $\mathbf{X}$. The second method is to delete the components for which the absolute value of γ is below some specified cutoff, γ_0, as they are relatively unimportant as predictors of the emergent behavior, $\mathbf{Y}$. Although criteria for selecting λ_0 and γ_0 are somewhat ad-hoc, Cangelosi and Goriely (2007) suggest choosing λ_0 such that $\frac{\lambda_0}{\sum_{i=1}^{p}\lambda_i} \leqslant 0.05$. Massy (1965) shows that, after scaling $\mathbf{X}$, the γ's can be viewed as correlation coefficients between $\mathbf{Y}$ and the principal components, so standard guidelines for interpreting correlation coefficients (Rodgers and Nicewander, 1988) can be used to choose γ_0. Both strategies will result in a regression of the form

$$\mathbf{Y} = \mathbf{T}^*\gamma^* + \epsilon,$$

where $\mathbf{T}^*$ is the $n \times k$ score matrix after the removal of $p - k$ principal components and γ^* is the corresponding set of coefficients.

To identify which of the original inputs, $\mathbf{X}$, are relevant to predicting the emergent behavior, $\mathbf{Y}$, based on the principal components, there are at least two strategies. One strategy, forward identification, as presented by King and Jackson (1999), is to identify as relevant the inputs which have the highest loading on each of the k principal components used in the regression and deem the remaining $p-k$ inputs as irrelevant. A second strategy, backward elimination, as proposed by Krzanowski (1987), identifies the input with the highest loading on each of the $p-k$ principal components that were deleted as predictors from the regression, and selects these inputs as irrelevant.

Using this procedure, it is possible that multiple inputs may be identified as irrelevant with respect to a particular emergent behavior. Given the complex nature of most ABMs, this should not be surprising. Because the rules/inupts are identified as irrelevant only with respect to a specific emergent behavior, however, it does not mean these should be removed from the model entirely, because they still potentially contribute to other model functions and development of other emergent behavior.

The fact that emergent behavior is not affected by irrelevant rules is an important feature of ABMs, since it mitigates the risk of reducing simulation quality. While parsimony is an important consideration in model development, inclusion of an irrelevant rule will not undermine the utility of the model. Adding an irrelevant rule will, however, increase the complexity of a model (at least in the Komolgorov sense) and make the model less favorable, *ceteris paribus*, to a model without the rule.

Greenhouse Model

As an illustration of this concept, we examine an ABM of technology use among a community of greenhouse owners developed by Kasmire et al. (2013). Agents in this model are greenhouse owners, who make decisions about what technology to use to maxamize crop production, and technology markets, that collaborate with one another to make improvements to existing technology. The emergent behavior we consider for this analysis is the total production of the greenhouse owners over a fixed period of time.

The model has $p = 48$ input settings, $\mathbf{X}$, that influence the behavior of technology markets and greenhouse owners. After running 100 simulations, we scale the columns of $\mathbf{X}$ and perform PCA and regression as described above. Table 2.4.1 gives the principal components' eigenvalues, γ coefficients from regression and indicates which components were retained using different levels of γ_0.

As we decrease γ_0, the number of principal components retained increases and the fit of the regression model improves, up to a certain point. The chosen values of γ_0 result in regression models consisting of 1, 8, 16 and 36 principal components as covariates, respectively. The regression model based on 36 principal components shows significantly better fit than the other models, pointing to the necessity of multiple factors in explaining the complex function of this ABM.

Of the criteria considered in this analysis, a value of $\gamma_0 = 0.01$ is the most conservative in terms of deleting principal components. As a result, this value will result in the fewest number of inputs being identified as irrelevant. This indicates the importance of the value of γ_0 in this procedure, especially in model development when identifying an input as irrelevant could result in its removal from the ABM. Based on the four values of γ_0 used above, we can examine which of the original 48 inputs were identified as relevant with respect to the total production of the greenhouse owners, using both the forward and backward strategies discussed earlier in section 2.4.1. A summary of the model inputs and their relevance is given in Table 2.4.1.

The forward identification and backward elimination methods for classification of the original inputs agree in 78.6% of cases. All but one input was identified as irrelevant for some criteria, while there were five inputs that were identified as irrelevant based on all criteria. Four of the five inputs identified as irrelevant with

Table 2.1: Eigenvalues and γ coefficients for the 48 principal components of the Greenhouse model and indications of the components retained for values of γ_0. Adjusted R^2 is given for the four regression models to measure goodness-of-fit (adjusted R^2 for regression using all 48 principal components is 0.863).

	Eigenvalue	γ	$\gamma_0 = 0.2$	$\gamma_0 = 0.1$	$\gamma_0 = 0.05$	$\gamma_0 = 0.01$
Component 1	5.268	-0.116		•	•	•
Component 2	3.581	-0.183		•	•	•
Component 3	1.631	-0.125		•	•	•
Component 4	1.414	0.032				•
Component 5	1.397	-0.004				
Component 6	1.359	-0.144		•	•	•
Component 7	1.222	0.045				•
Component 8	1.098	-0.026				•
Component 9	1.091	0.087			•	•
Component 10	1.043	0.070			•	•
Component 11	1.004	-0.091			•	•
Component 12	1.002	-0.022				•
Component 13	1.000	-0.005				
Component 14	1.000	0.028				•
Component 15	1.000	0.020				•
Component 16	1.000	0.015				•
Component 17	1.000	-0.022				•
Component 18	1.000	0.006				
Component 19	1.000	0.001				
Component 20	1.000	0.019				•
Component 21	1.000	-0.005				
Component 22	1.000	-0.001				
Component 23	1.000	0.033				•
Component 24	1.000	-0.002				
Component 25	1.000	-0.001				
Component 26	0.986	0.024				•
Component 27	0.985	-0.075			•	•
Component 28	0.984	-0.002				
Component 29	0.984	0.009				
Component 30	0.984	-0.020				•
Component 31	0.982	0.107		•	•	•
Component 32	0.974	0.023				•
Component 33	0.844	0.882	•	•	•	•
Component 34	0.830	-0.194		•	•	•
Component 35	0.697	-0.014				•
Component 36	0.600	-0.028				•
Component 37	0.578	-0.011				•
Component 38	0.487	0.118		•	•	•
Component 39	0.419	-0.012				•
Component 40	0.305	-0.058			•	•
Component 41	0.288	0.078			•	•
Component 42	0.277	-0.053			•	•
Component 43	0.249	-0.016				•
Component 44	0.233	-0.031				•
Component 45	0.196	-0.006				
Component 46	0.144	-0.003				
Component 47	0.088	0.057			•	•
Component 48	0.028	0.017				•
Adjusted R^2	-	-	0.002	0.327	0.527	0.861

respect to the total greenhouse production (InfluenceRangeMain, TechA CO2Max, TechC HumMin, TechC LightMin) directly affect the behavior of technology markets and not greenhouse owners. Although these rules were found to be irrelevant with respect to total greenhouse production, they are central to the development of the technology markets and would likely be identified as relevant with respect to an emergent behavior based on technology improvement.

2.5 Discussion

The statistical theory related to ABMs is currently developing. While some theory and approaches for traditional models can be applied to ABMs, there are many features which are unique to ABMs and which require further investigation. Model validation protocol, assessments of model equivalence and model complexity for ABMs require the consideration of the multiple levels of behavior within the model to make determinations on these topics.

Model selection is a topic that remains to be thoroughly explored for ABMs. Elements from model selection for traditional statistical models such as predictive accuracy and error minimization can be incorporated into a model selection procedure for ABMs, but often the lack of observed data and non-linearity of model behavior can limit the utility of such techniques (as discussed in chapter 3). While some of the theory on model complexity and model equivalence presented earlier in this chapter can be incorporated into a model selection protocol, more attention should be given to important elements in such a procedure.

Table 2.2: Original inputs which were identified as relevant with respect to total greenhouse production using forward identification (F) and backward elimination (B) methods, for values of $\gamma_0 = (0.2, 0.1, 0.05, 0.1)$. Bullets ($\bullet$) indicate variables identifed as relevant and dashes (-) indicate irrelevant variables. A description of the inputs can be found in Kasmire et al. (2013).

	$\gamma_0 = 0.2$		$\gamma_0 = 0.1$		$\gamma_0 = 0.05$		$\gamma_0 = 0.01$	
Variable	F	B	F	B	F	B	F	B
StubbornnessFactor	-	-	-	-	-	-	-	-
InitialBankaccount	-	-	-	-	-	-	$\bullet$	-
CostPriceCooperativeMultiplicator	-	-	-	-	-	-	$\bullet$	$\bullet$
InfluenceRangeSec	-	-	-	-	-	-	$\bullet$	-
InfluenceRangeMain	-	-	-	-	-	-	-	-
CostPriceRange	-	-	-	$\bullet$	-	$\bullet$	-	$\bullet$
CostPriceMin	-	-	-	-	-	-	$\bullet$	$\bullet$
CostpriceMax	-	-	-	-	$\bullet$	-	$\bullet$	$\bullet$
EnergyUseMin	-	-	-	-	-	-	$\bullet$	$\bullet$
EnergyUseMax	-	-	$\bullet$	-	$\bullet$	-	$\bullet$	-
LifespanMin	-	-	-	-	-	-	$\bullet$	$\bullet$
LifespanMax	-	-	-	-	$\bullet$	-	$\bullet$	$\bullet$
ImproveSameTechCounter	-	-	-	$\bullet$	-	$\bullet$	-	$\bullet$
TechA TempMin	-	-	-	-	$\bullet$	-	$\bullet$	$\bullet$
TechA CO2Max	-	-	-	-	-	-	-	-
TechA HumMin	-	-	-	-	-	-	-	$\bullet$
TechA LightMax	-	-	-	-	-	-	$\bullet$	-
TechB TempMax	-	-	-	-	-	-	-	$\bullet$
TechB CO2Min	-	-	-	-	-	-	$\bullet$	-
TechB HumMin	-	-	-	-	-	-	-	$\bullet$
TechB LightMin	-	-	-	-	-	-	$\bullet$	-
TechC TempMax	-	-	-	-	-	-	-	$\bullet$
TechC CO2Max	-	-	-	-	-	-	$\bullet$	$\bullet$
TechC HumMin	-	-	-	-	-	-	-	-
TechC LightMin	-	-	-	-	-	-	-	-
TechD TempMax	-	-	-	-	-	-	$\bullet$	-
TechD CO2Max	-	-	-	-	-	-	$\bullet$	-
TechD HumMin	-	-	-	-	-	-	$\bullet$	-
TechD LightMin	-	-	-	-	-	-	-	$\bullet$
VeggiesIdealTemp	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$
VeggiesIdealCO2	-	-	$\bullet$	-	$\bullet$	-	$\bullet$	$\bullet$
VeggiesIdealHum	-	-	-	-	$\bullet$	-	$\bullet$	$\bullet$
VeggiesIdealLight	-	-	-	-	-	-	$\bullet$	$\bullet$
VeggiesPotentialGrowth	-	-	-	-	$\bullet$	-	$\bullet$	$\bullet$
FlowersIdealTemp	-	-	-	-	-	-	$\bullet$	$\bullet$
FlowersIdealCO2	-	-	-	$\bullet$	-	$\bullet$	$\bullet$	$\bullet$
FlowersIdealHum	-	-	-	-	-	-	$\bullet$	$\bullet$
FlowersIdealLight	-	-	-	-	$\bullet$	-	$\bullet$	$\bullet$
FlowersPotentialGrowth	-	-	$\bullet$	-	$\bullet$	$\bullet$	$\bullet$	$\bullet$
ExternalTemp	-	-	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$
ExternalCO2	-	-	$\bullet$	-	$\bullet$	-	$\bullet$	$\bullet$
ExternalHum	-	-	$\bullet$	-	$\bullet$	-	$\bullet$	$\bullet$
ExternalLight	-	-	-	-	-	-	$\bullet$	$\bullet$
FuelPrice	-	-	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$
VeggiesMarketPrice	-	-	-	-	-	-	$\bullet$	$\bullet$
FlowersMarketPrice	-	-	-	$\bullet$	-	$\bullet$	$\bullet$	$\bullet$
VeggiesPurchasePrice	-	-	-	-	$\bullet$	-	$\bullet$	$\bullet$
FlowersPurchasePrice	-	-	-	-	$\bullet$	$\bullet$	$\bullet$	$\bullet$

3

Gaussian Process Emulators

3.1 Overview

ABMs are often used to simulate complex real world processes. Running these simulations is, in most cases, computationally taxing and cannot be carried out on the scale which we would like. Even in cases where the runtime is not a limiting issue for the ABM, generating thousands of runs is not an efficient method for performing inference on the system being modeled. To address this problem, we are able to utilize emulators. An emulator is a stochastic process that serves as a representation of a simulator (in this case our ABM), which incorporates full probabilistic specification based on beliefs and knowledge. Using an emulator to serve as a surrogate for an ABM can be particularly useful in instances when the original model has a run time of several hours, and there is a need for simulations to run in real time. Such approaches have been investigated in the context of weather and environmental modelling (Margvelashvili, 2011) as well as transportation (Rasouli and Timmermans, 2013). Emulation in a Bayesian framework has been discussed in detail, notably in Kennedy and O'Hagan (2001), Craig et al. (2001), Bliznyuk et al. (2008) and Liu and West (2009).

3.2 Emulation of high-dimensional computer output

Some of the most prominent work in the utilization of emulators to model simulation output are Kennedy and O'Hagan (2000) and Kennedy and O'Hagan (2001). Here, I present an expansion on this methodology presented by Higdon et al. (2008) (discussed in an earlier application in Higdon et al. (2004)), incorporating approaches developed by Santner et al. (2003).

Often, when the objective of an experiment is to understand and predict the behavior of complex systems and procedures, the subject of interest cannot be observed frequently enough to gather sufficient data to perform analyses. It is, however, possible to make use of computer models such as ABMs to simulate the process of interest. Naturally, some uncertainty will arise in the selection of certain inputs and parameters in our simulator, but this can be mitigated by utilizing actual observed data to guide the simulator and assist in performing inference.

To begin, suppose we have obtained n observations of the system of interest. For $i = 1, ..., n$, let the vector $\mathbf{x}_i$ represent the conditions under which the ith ovservation of the system is made and define its dimension to be p_x. Let $y(\mathbf{x}_i)$ denote the actual ith observation. (Note that the term 'conditions' will be problem-specific: In Higdon et al.'s study of implosion study, conditions specified the mass of the explosive used). The dimension of $\mathbf{x}_i$ can vary depending on the system and experiment. Then, we have a simple model:

$$y(\mathbf{x}_i) = \xi(\mathbf{x}_i) + \epsilon(\mathbf{x}_i) \tag{3.1}$$

where $\xi(\mathbf{x}_i)$ is the response of the system under conditions $\mathbf{x}_i$ and $\epsilon(\mathbf{x}_i)$ represents observation error. In many instances, systems will be well-enough understood that the errors can be treated as having a known distribution.

The ABM (simulator) will have certain calibration settings $\mathbf{t}$ which serve as inputs and affect the output. For ABMs, these calibrations will likely include, among other things, the rules which determine agent behavior. While our goal is to model system behavior and observations, the values of the calibration settings are not known for our n actual observations. In this case, $\boldsymbol{\theta}$ is used to represent the optimal, but unknown, values of these settings.

Letting $\eta(\mathbf{x}, \mathbf{t})$ represent the ABM output under conditions $\mathbf{x}$ and calibration values $\mathbf{t}$, the observed data $\mathbf{y} = (y(\mathbf{x}_1), ..., y(\mathbf{x}_n))^T$ can now be modeled statistically:

$$y(\mathbf{x}_i) = \eta(\mathbf{x}_i, \boldsymbol{\theta}) + \delta(\mathbf{x}_i) + \epsilon(\mathbf{x}_i) \tag{3.2}$$

where $\delta(\mathbf{x}_i)$ is a stochastic term to account for systematic discrepancies between the ABM $\eta(\mathbf{x}_i, \boldsymbol{\theta})$ and the physical process $y(\mathbf{x}_i)$. This additive decomposition is based on Kennedy and O'Hagan (2001).

Again, because in many applications an ABM will be complex and computationally demanding, only a limited number of simulations can be obtained. Suppose we carry out m ABM runs with conditions and calibration settings $\mathbf{x}_j^*, \mathbf{t}_j^*$, producing output $\boldsymbol{\eta} = (\eta(\mathbf{x}_1^*, \mathbf{t}_2^*, ..., \eta(\mathbf{x}_m^*, \mathbf{t}_m^*))^T$. The ABM output can then be used as additional data in setting up our emulator for analysis. Because of the complexity of the ABM, the function representing its output, η, is unknown. To address this uncertainty, we model η probabilistically in order to approximate ABM outputs for untried input values $(\mathbf{x}, \mathbf{t})$. Following O'Hagan (1978), Higdon et al. utilize a Gaussian Process model for $\eta(\mathbf{x}, \mathbf{t})$.

As a prior for $\eta(\mathbf{x}, \mathbf{t})$ Higdon et al. proposed a Gaussian Process with a constant mean function $\mu(\mathbf{x}, \mathbf{t})$ and a product covariance with power exponential form follwing Sacks et al. (1989). Recall from above that $p_x = \dim(\mathbf{x})$ and define $p_t = \dim(\mathbf{t})$ to be the dimension of the calibration settings. Thus, the covariance function will have the following form:

$$Cov((\mathbf{x},\mathbf{t}),(\mathbf{x}',\mathbf{t}')) = \frac{1}{\lambda_\eta}\prod_{j=1}^{p_x}\rho_{\eta j}^{4(x_j-x_j')^2}\prod_{k=1}^{p_t}(\rho_{\eta,p_x+k})^{4(t_j-t_j')^2}$$
$$= \frac{1}{\lambda_\eta}R((\mathbf{x},\mathbf{t}),(\mathbf{x}',\mathbf{t}');\boldsymbol{\rho}_\eta). \quad (3.3)$$

Here, λ_η controls the marginal precision of η and $\boldsymbol{\rho}_\eta$ is a $p_x + p_t$ vector which controls the dependence strength in each component direction of $\mathbf{x}$ and $\mathbf{t}$. The quantity $\rho_{\eta j}$ is the correlation between outputs which are evaluated at inputs differing only in the jth dimension by half their domain. Independent priors on $\mu(\mathbf{x},\mathbf{t})$, λ_η and $\boldsymbol{\rho}_\eta$ complete the prior model specification.

Continuing the specification of the model, a Gaussian process is used to model the discrepancy term $\delta(\mathbf{x})$ as well. A zero mean is specified for the Gaussian process as we expect our observations $y(\mathbf{x}_i)$ to be close to the simulation $\eta(\mathbf{x}_i,\mathbf{t}_i)$.

The covariance function has the form:

$$Cov(\mathbf{x},\mathbf{x}') = \frac{1}{\lambda_\delta}\prod_{j=1}^{p_x}\rho_{\delta j}^{4(x_j-x_j')^2}$$
$$= \frac{1}{\lambda_\delta}R((\mathbf{x},\mathbf{x}');\boldsymbol{\rho}_\delta). \quad (3.4)$$

Here, λ_δ controls the marginal precision of δ and $\boldsymbol{\rho}_\delta$ is a p_x vector which controls the dependence strength in each component direction of $\mathbf{x}$. The quantity $\rho_{\delta j}$ is the correlation between values of the discrepancy which are evaluated at inputs $\mathbf{x}$ differing only in the jth dimension by half their domain. To complete the prior model specification, independent priors are placed on λ_δ and $\boldsymbol{\rho}_\delta$ that depend on knowledge of how well the simulator models reality (how close our observations will be to the simulation).

Define the concatenation of $\mathbf{y}$ and $\boldsymbol{\eta}$ as the $m+n$-vector $\mathcal{D} = (\mathbf{y}^T, \boldsymbol{\eta}^T)^T$. Define $\boldsymbol{\Sigma}_y$ to be the $n \times n$ observation covariance matrix, $\boldsymbol{\Sigma}_\eta$ to be the $(m+n)\times(m+n)$

covariance matrix obtained from applying (3.3) to the input points $(\mathbf{x}_i, \boldsymbol{\theta})$ and $(\mathbf{x}_i^*, \mathbf{t}_i^*)$ corresponding to $\mathcal{D}$ and define $\boldsymbol{\Sigma}_\delta$ to be the $n \times n$ covariance matrix obtained from applying (3.4) to the input settings $\mathbf{x}_i$ corresponding to the observations $\mathbf{y}$.

Then, define

$$\boldsymbol{\Sigma}_{\mathcal{D}} = \boldsymbol{\Sigma}_\eta + \begin{bmatrix} \boldsymbol{\Sigma}_\delta + \boldsymbol{\Sigma}_y & \mathbf{0}_{n\times m} \\ \mathbf{0}_{m\times n} & \mathbf{0}_{m\times m} \end{bmatrix}. \tag{3.5}$$

This formulation leads to a multivariate Normal likelihood for $\mathcal{D}$:

$$\mathcal{L}(\mathcal{D}|\boldsymbol{\theta}, \mu, \lambda_\eta, \boldsymbol{\rho}_\eta, \lambda_\delta, \boldsymbol{\rho}_\delta, \boldsymbol{\Sigma}_y) \propto |\boldsymbol{\Sigma}_{\mathcal{D}}|^{-1/2} \exp\left(-\frac{1}{2}(\mathcal{D} - \mu * \mathbf{1}_{m+n})^T \boldsymbol{\Sigma}_{\mathcal{D}}^{-1} (\mathcal{D} - \mu * \mathbf{1}_{m+n})\right). \tag{3.6}$$

If we define a prior for $\boldsymbol{\theta}$ that is independent of the other parameters, then (3.6) leads to the following joint posterior:

$$\pi(\boldsymbol{\theta}, \mu, \lambda_\eta, \boldsymbol{\rho}_\eta, \lambda_\delta, \boldsymbol{\rho}_\delta|\mathcal{D}) \propto \mathcal{L}(\mathcal{D}|\boldsymbol{\theta}, \mu, \lambda_\eta, \boldsymbol{\rho}_\eta, \lambda_\delta, \boldsymbol{\rho}_\delta, \boldsymbol{\Sigma}_y) \times \pi(\mu) \times \pi(\lambda_\eta) \times \pi(\boldsymbol{\rho}_\eta) \times \pi(\lambda_\delta) \times \pi(\boldsymbol{\rho}_\delta) \times \pi(\boldsymbol{\theta}). \tag{3.7}$$

From this posterior, we can infer about quantities of interest that go into our simulation. Of primary interest are the mean function for the process, μ, and the calibration parameters $\boldsymbol{\theta}$.

It is of interest to use a model to describe simulator output at untried input values. To this end, we use the ouput from the m-runs of the simulator to construct a Gaussian Process model (emulator) to emulate the simulator at arbitrary input settings. To simplify this formulation, we can standardize the design space to the hypercube $(\mathbf{x}, \mathbf{t}) \in [0, 1]^{p_x+p_t}$.

3.2.1 ABM Output Model

To construct the emulator, we model the ABM output with a p_η dimensional basis representation:

$$\eta(\mathbf{x}, \mathbf{t}) = \sum_{i=1}^{p_\eta} \mathbf{k}_i w_i(\mathbf{x}, \mathbf{t}) + \boldsymbol{\epsilon}. \tag{3.8}$$

Here, $\{\mathbf{k}_1, ..., \mathbf{k}_{p_\eta}\}$ is a set of orthogonal n_η-dimensional basis vectors, $w_i(\mathbf{x}, \mathbf{t})$ are Gaussian processes over the input space and $\boldsymbol{\epsilon}$ is an n_η-dimensional error term. This formulation allows us to constuct p_η independent Gaussian process models instead of a single model mapping $[0, 1]^{p_x+p_t} \rightarrow \mathbb{R}^{n_\eta}$. This allows efficient representation of the output by means of Principal Components.

After standardizing simulation outputs, they are stored in an $n_\eta \times m$ matrix $\boldsymbol{\Xi}$. From the singular value decomposition of $\boldsymbol{\Xi}$, we obtain the basis vectors $\mathbf{K}_\eta = \left[\mathbf{k}_1, ..., \mathbf{k}_{p_\eta}\right]$. By scaling each $\mathbf{k}_i$, it allows each process $w_i(\mathbf{x}, \mathbf{t})$ to be modeled with zero mean and marginal variance near 1. While there is no well-established choice for the number of basis vectors p_η, it should be such that at least 99% of the variance in the m simulator runs is explained. (Higdon et al. suggested that p_η=5 is typically sufficient). Bayarri et al. (2007a) also proposed using wavelet basis elements for the decomposition of η in cases where functions of model output are irregular.

Following from (3.8), we repsent the principal component weights as

$$w_i(\mathbf{x}, \mathbf{t}) \sim GP(0, \lambda_{w_i}^{-1} R((\mathbf{x}, \mathbf{t}), (\mathbf{x}', \mathbf{t}'); \boldsymbol{\rho}_{w_i})) \tag{3.9}$$

with the covariance function given by (3.3), marginal precision λ_{wi} and correlation distances for each input dimension given by $\boldsymbol{\rho}_{wi}$.

Define the m-vector $\mathbf{w}_i = (w_i(\mathbf{x}_1^*, \mathbf{t}_1^*), w_i(\mathbf{x}_2^*, \mathbf{t}_2^*), ..., w_i(\mathbf{x}_m^*, \mathbf{t}_m^*))$ to be the restriction of w_i to the input settings at which the simulator was actually run. Define the $m * p_\eta$-dimensional vector $\mathbf{w}$ to be the concatenation of all of the $\mathbf{w}_i$. The vector $\mathbf{w}$ will have the prior distribution $N_{m*p_\eta}(0, \mathrm{diag}(\lambda_{wi}^{-1}\mathbf{R}((\mathbf{x}^*, \mathbf{t}^*); \boldsymbol{\rho}_{wi})))$. In similar fashion to the parametrization of η previoiusly, $\boldsymbol{\rho}_w$ contains the $p_\eta(p_x + p_t)$ spatial correlations. Define $\boldsymbol{\Sigma}_w = \text{diag}(\lambda_{wi}^{-1}\mathbf{R}(\mathbf{x}^*, \mathbf{t}^*; \boldsymbol{\rho}_{wi})$.

The λ_{wi} is given independent Gamma(a_w, b_w) priors and the ρ_{wik} are given independent Beta(a_{ρ_w}, b_{ρ_w}) priors. Having standardized the output and expecting the marginal variance of each $w_i(\mathbf{x}, \mathbf{t})$ to be approximately 1, we can set our prior expectation for λ_{wi} to be 1, setting $a_w = b_w = 5$.

The selection of the hyperparameters for ρ_{wik} depends on how many of the inputs are expected to actively influence the simulator response. Choosing $a_{\rho_w} = 1$ and $0 < b_{\rho_w} < 1$, puts prior mass near 1. Under the parametrization, if $\rho_{wik} = 1$ then input k is inactive for the ith principal component.

Referring back to (3.8), if we assume the error vector to be iid normal, this simplifies specification of the sampling model for the simulator output. Define the mn_η-dimensional vector $\boldsymbol{\eta}$ to be the concatenation of the m standardized output vectors (the columns of $\boldsymbol{\Xi}$). Then, given the precision of each error, λ_η, the sampling model for the ABM output is $\boldsymbol{\eta} \sim N_{m*n_\eta}(0, \mathbf{K}\boldsymbol{\Sigma}_w\mathbf{K}^T + \lambda_\eta^{-1}\mathbf{I}_{m*n_\eta})$. Here the matrix $\mathbf{K} = [\mathbf{I}_m \otimes \mathbf{k}_1; ...; \mathbf{I}_m \otimes \mathbf{k}_{p_\eta}]$ where $\mathbf{k}_i$ are the basis vectors from the singular value decomposition of $\boldsymbol{\Xi}$ as previously defined and $\boldsymbol{\Sigma}_w$ is also as defined above. The quantity λ_η has a specified Gamma(a_η, b_η) prior.

3.2.2 Discrepancy Model

In a similar fashion to the statistical specification of the representation of the ABM output, η, a model for the discrepancy function δ must also be specified. As above, using a basis representation with a Gaussian Process prior placed on the basis weights, δ can be represented as:

$$\delta(\mathbf{x}) = \sum_{k=1}^{p_\delta} \mathbf{d}_k v_k(\mathbf{x}) \tag{3.10}$$

where $\mathbf{d}_k$ are basis functions and $v_k(\mathbf{x})$ are their corresponding weights, each with GP priors. The $\mathbf{d}_k$ are specified based on what is known about the actual process and any knowledge of bias in the simulator. The choice of the value p_δ depends on the basis kernel width in component directions. In instances where little is known *a priori* about the form of δ, a Gaussian or Bessel function of the first kind are standard choices for the kernel.

Basis vectors are divided into F groups, denoted $G_1, G_2, \ldots, G_F$, in which each group has the set of coefficients $\mathbf{v}_i = (v_{i,1}(\mathbf{x}), \ldots, v_{i,|G_i|}(\mathbf{x}))^T$ for $i = 1, 2, ...F$. Each set of coefficients $\mathbf{v}_i$ is modeled with independent zero-mean Gaussian Process priors:

$$\mathbf{v}_i \sim GP(0_{|G_i|}, \lambda_{vi}^{-1}\mathbf{I}_{|G_i|} \otimes R((\mathbf{x}, \mathbf{x}'); \boldsymbol{\rho}_{vi})) \qquad i = 1, ..., F$$

where λ_{vi} is the common marginal percision of the elements of $\mathbf{v}_i(\mathbf{x})$, $\boldsymbol{\rho}_{vi}$ is a p_x vector controlling correlation strength along components of $\mathbf{x}$, and $R((\mathbf{x}, \mathbf{x}'); \boldsymbol{\rho}_{vi}))$ is the stationary GP correlation from (3.4). Setting $F = 1$ corresponds to all basis coefficients having common precision and correlation distance.

As with the specification of the model for the simulator output, the precisions λ_{vi} are assigned Gamma(a_{vi}, b_{vi}) priors and ρ_{vi} are assigned independent Beta(a_{ρ_v}, b_{ρ_v}) priors. Often, uninformative priors are used for these quantities.

The n experimentally observed data points $\mathbf{y}(\mathbf{x}_i)$, $i = 1, ..., n$, can be modeled as:

$$\mathbf{y}(\mathbf{x}_i) = \eta(\mathbf{x}_i, \boldsymbol{\theta}) + \delta(\mathbf{x}_i) + \mathbf{e}(\mathbf{x}_i)$$

where we define the number of observations contained in $\mathbf{y}(\mathbf{x}_i)$ as n_{y_i} and the errors are modeled as $\mathbf{e}(\mathbf{x}_i) \sim N_{y_i}(0, (\lambda_y \mathbf{W}_i)^{-1})$.

3.2.3 Emulator Design

Now, using the basis representations of the ABM and discrepancy, the experimental data can now be modeled as:

$$\mathbf{y}(\mathbf{x}_i) = \mathbf{K}_i \mathbf{w}(\mathbf{x}_i, \theta) + \mathbf{D}_i \mathbf{v}(\mathbf{x}_i) + \mathbf{e}(\mathbf{x}_i) \tag{3.11}$$

where $\mathbf{K}_i$ and $\mathbf{D}_i$ are the matrices of basis vectors of $\mathbf{k}_i$ and $\mathbf{d}_i$, respectively.

Define the matrix $\mathbf{B}$ to be $[\text{diag}(\mathbf{D}_1, ..., \mathbf{D}_n); \text{diag}(\mathbf{K}_1, ..., \mathbf{K}_n)]\text{diag}(\mathbf{P}_D^T, \mathbf{P}_K^T)$ where $\mathbf{P}_D$ and $\mathbf{P}_K$ are permutation matrices. Define $\mathbf{W}_y = \text{diag}(\mathbf{W}_1, ..., \mathbf{W}_n)$. Now, the sampling model for all of the experimental data can be expressed as:

$$\mathbf{y} \sim N_{n_y}(0, \mathbf{B}\text{diag}(\mathbf{\Sigma}_v, \mathbf{\Sigma}_u)\mathbf{B}^T + (\lambda_y \mathbf{W}_y)^{-1}).$$

The permutation matrices are needed in the specification of $\mathbf{B}$ because basis weights were separated in defining the $\mathbf{v}(\mathbf{x}_i)$ and $\mathbf{w}(\mathbf{x}_i, \theta)$. The matrix $\mathbf{\Sigma}_v = \lambda_{vi}^{-1}\mathbf{I}_{|G_i|} \otimes R((\mathbf{x}, \mathbf{x}'); \boldsymbol{\rho}_{vi})$ is the covariance matrix from the specification of v_i previously, and $\mathbf{\Sigma}_u$ is the covariance matrix specified in the GP specification of w_i with $\boldsymbol{\theta}$ in place of t (since the observed data are under the best, unobserved calibration settings $\boldsymbol{\theta}$ rather than specified settings, t). Since all of these inputs are assumed to be at calibration setting $\boldsymbol{\theta}$, the correlations will depend only on $\mathbf{x}$ in this case.

We assume a Gamma(a_y, b_y) prior for the observational error precision λ_y. Because the observation precision $\mathbf{W}_y$ is often fairly well-known, an informative prior is used for λ_y favoring values near 1.

Now, define the $n * p_\delta$-dimensional vector $\mathbf{v} = vec([\mathbf{v}(\mathbf{x}_1); ...; \mathbf{v}(\mathbf{x}_n)]^T)$ and define the $n * p_\eta$ vector

$$\mathbf{u}(\boldsymbol{\theta}) = vec([\mathbf{w}(\mathbf{x}_1, \boldsymbol{\theta}); ...; \mathbf{w}(\mathbf{x}_n, \boldsymbol{\theta})]^T)$$

to be the GP model for the ABM component of the observed experiment, at input setting $\mathbf{x}_i$ and unknown parameter setting θ.

Recalling that $\mathbf{w} = (\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_m)^T$ from the representation of the ABM output at input $\mathbf{x}$ and calibration setting $\mathbf{t}$, now define the $d_z = (n(p_\eta + p_\delta) + mp_\eta)$-dimensional vector $\mathbf{z} = (\mathbf{v}^T, \mathbf{u}^T(\boldsymbol{\theta}), \mathbf{w}^T)^T$. The vector $\mathbf{z}$ will have prior distribution:

$$\mathbf{z} \sim N_{d_z}(0, \mathbf{\Sigma}_z) \tag{3.12}$$

where the covariance matrix is $\mathbf{\Sigma}_z = \begin{bmatrix} \mathbf{\Sigma}_v & 0 & 0 \\ 0 & \mathbf{\Sigma}_u & \mathbf{\Sigma}_{u,w} \\ 0 & \mathbf{\Sigma}_{u,w}^T & \mathbf{\Sigma}_w \end{bmatrix}$. The zeros in the matrix result from the independence between $\mathbf{u}(\theta)$ and $\mathbf{w}$ and $\mathbf{v}$, respectively. The matrices Σ_u, Σ_v and Σ_w have been defined previously and

$$\Sigma_{u,w} = \mathrm{diag}(\lambda_{wi}^{-1}\mathbf{R}(\mathbf{x}, \theta), (\mathbf{x}^*, \mathbf{t}^*); \rho_{wi}); i = 1, ..., p_\eta)$$

where $\mathbf{R}(\mathbf{x}, \theta), (\mathbf{x}^*, \mathbf{t}^*); \rho_{wi})$ is the $n \times m$ correlation matrix between n experimental settings $(\mathbf{x}_1, \theta), ..., (\mathbf{x}_1, \theta)$ crossed with the m simulator input settings $(\mathbf{x}_1^*, \mathbf{t}_1^*), ..., (\mathbf{x}_m^*, \mathbf{t}_m^*)$.

Now, noting that $\mathbf{y} = \mathbf{Bz} + e(\mathbf{x})$ and $\eta = \mathbf{Kz} + \epsilon$, all of the data from both experiment and simulator output can be represented as:

$$\begin{bmatrix} \mathbf{y} \\ \boldsymbol{\eta} \end{bmatrix} = \begin{bmatrix} \mathbf{B} & 0 \\ 0 & \mathbf{K} \end{bmatrix} \mathbf{z} + \begin{bmatrix} \mathbf{e} \\ \epsilon \end{bmatrix}. \tag{3.13}$$

Since the errors $\mathbf{e}$ and ϵ are multivariate normal, the joint sampling distribution of all of the observations will also be multivariate normal.

Bringing together the specifications of the all of the calibration and model parameters, the joint posterior distribution will have the form:

$$\begin{aligned} \pi(\lambda_\eta, \lambda_w, \rho_w, \lambda_y, \lambda_v, \rho_v, \boldsymbol{\theta} | \mathbf{y}, \boldsymbol{\eta}) \propto & L(\mathbf{y}, \boldsymbol{\eta} | \lambda_\eta, \lambda_w, \rho_w, \lambda_y, \lambda_v, \rho_v, \theta) \times \pi(\lambda_\eta) \\ & \times \prod_{i=1}^{p_\eta} \pi(\lambda_{wi}) \times \prod_{i=1}^{p_\eta} \prod_{k=1}^{p_x+p_t} \pi(\rho_{wik}) \times \pi(\lambda_y) \\ & \times \prod_{i=1}^{F} \pi(\lambda_{vi}) \times \prod_{i=1}^{F} \prod_{k=1}^{p_x} \pi(\rho_{vik}) \times \pi(\boldsymbol{\theta}) \end{aligned} \tag{3.14}$$

where the priors are as previously specified and $\pi(\boldsymbol{\theta})$ denotes the prior distribution of $\boldsymbol{\theta}$, which, due to the uncertainty of these values a priori, is taken to be uniform on a p_t-dimensional rectangle.

In order to reduce the burden of computation of $L(\mathbf{y}, \boldsymbol{\eta} | \lambda_\eta, \lambda_w, \rho_w, \lambda_y, \lambda_v, \rho_v, \boldsymbol{\theta})$, we use (3.13) and the fact that the distributions of $\mathbf{z}$ and $\begin{bmatrix} \mathbf{e} \\ \epsilon \end{bmatrix}$ are known to obtain:

$$\begin{aligned} L(\mathbf{y}, \boldsymbol{\eta} | \lambda_\eta, \lambda_w, \rho_w, \lambda_y, \lambda_v, \rho_v, \boldsymbol{\theta}) \propto & L(\hat{\mathbf{z}} | \cdot) \times \lambda_\eta^{m(n_\eta - p_\eta)/2} e^{\left[-\frac{1}{2}\lambda_\eta \boldsymbol{\eta}^T (\mathbf{I} - \mathbf{K}(\mathbf{K}^T\mathbf{K})^{-1}\mathbf{K}^T)\boldsymbol{\eta}\right]} \\ & \times \lambda_y^{(n_y - \mathrm{rank}(\mathbf{B}))/2} e^{\left[-\frac{1}{2}\lambda_y \mathbf{y}^T (\mathbf{W}_y - \mathbf{W}_y\mathbf{B}(\mathbf{B}^T\mathbf{W}_y\mathbf{B})^{-1}\mathbf{B}^T\mathbf{W}_y)\mathbf{y}\right]} \end{aligned} \tag{3.15}$$

where $\hat{\mathbf{z}} = \text{vec}([(\mathbf{B}^T\mathbf{W}_y\mathbf{B})^{-1}\mathbf{B}^T\mathbf{W}_y\mathbf{y}; (\mathbf{K}^T\mathbf{K})^{-1}\mathbf{K}^T\boldsymbol{\eta}])$ and $L(\hat{\mathbf{z}}|\cdot)$ can be computed from the normality of $\hat{\mathbf{z}}$ as a linear combination of normals. This result follows from least squares estimation and multivariate normal theory.

So, $\hat{\mathbf{z}} \sim N(0, \boldsymbol{\Sigma}_{\hat{z}})$ where $\boldsymbol{\Sigma}_{\hat{z}} = \boldsymbol{\Sigma}_z + \begin{bmatrix} (\lambda_y\mathbf{B}^T\mathbf{W}_y\mathbf{B})^{-1} & 0 \\ 0 & (\lambda_\eta\mathbf{K}^T\mathbf{K})^{-1} \end{bmatrix}$. The joint likelihood for the observations derived in (3.15) can be incorporated into the form for the full joint posterior:

$$\begin{aligned}
\pi(\lambda_\eta, \lambda_w, \rho_w, \lambda_y, \lambda_v, \rho_v, \boldsymbol{\theta}|\mathbf{y}, \boldsymbol{\eta}) \propto |\boldsymbol{\Sigma}_{\hat{z}}^{-1/2}| \exp\{-\frac{1}{2}\hat{\mathbf{z}}^T\boldsymbol{\Sigma}_{\hat{z}}^{-1}\hat{\mathbf{z}}\} \times \lambda_\eta^{a'_\eta - 1} e^{-b'_\eta\lambda_\eta} \\
\times \prod_{i=1}^{p_\eta} \lambda_{wi}^{a_w-1} e^{-b_w\lambda_{wi}} \times \prod_{i=1}^{p_\eta}\prod_{k=1}^{p_x+p_t} \rho_{wik}^{a_{\rho w}-1}(1-\rho_{wik})^{b_{\rho w}-1} \\
\times \lambda_y^{a'_y-1} e^{-b'_y\lambda_y} \times \prod_{i=1}^{F} \lambda_{vi}^{a_v-1} e^{-b_v\lambda_{vi}} \\
\times \prod_{i=1}^{F}\prod_{k=1}^{p_x} \rho_{vik}^{a_{\rho v}-1}(1-\rho_{vik})^{b_{\rho v}-1} \times \pi(\theta) \qquad (3.16)
\end{aligned}$$

where

$a'_\eta = \frac{m(n_\eta - p_\eta)}{2}$,

$a'_y = a_y + \frac{n_y - rank(\mathbf{B})}{2}$,

$b'_\eta = b_\eta + \frac{1}{2}\boldsymbol{\eta}^T(\mathbf{I} - \mathbf{K}(\mathbf{K}^T\mathbf{K})^{-1}\mathbf{K}^T)\boldsymbol{\eta}$,

$b'_y = b_y + \frac{1}{2}\lambda_y\mathbf{y}^T(\mathbf{W}_y - \mathbf{W}_y\mathbf{B}(\mathbf{B}^T\mathbf{W}_y\mathbf{B})^{-1}\mathbf{B}^T\mathbf{W}_y)\mathbf{y}$.

This expression reduces matrix inversion from order $(n_y + mn_\eta)$ in (3.14) to $n(p_\delta + p_\eta) + mp_\eta$ in (3.16). So, given observations and computer model output, posterior draws can be produced using standard MCMC methods. For components of $\boldsymbol{\rho}_w, \boldsymbol{\rho}_v$ and $\boldsymbol{\theta}$ are updated via Metropolis methods and the precision parameters are sampled using Hastings updates.

3.2.4 Posterior Prediction

After making posterior draws based on (3.16), predictions can now be made for the calibrated simulator $\boldsymbol{\eta}(\mathbf{x}^*, \boldsymbol{\theta})$ and the discrepancy $\boldsymbol{\delta}(\mathbf{x}^*)$ can be generated for any input values $\mathbf{x}^*$. This directly yields predictions of the behavior of the system $\boldsymbol{\xi}(\mathbf{x}^*) = \boldsymbol{\eta}(\mathbf{x}^*, \boldsymbol{\theta}) + \boldsymbol{\delta}(\mathbf{x}^*)$.

Because of the forms for $\boldsymbol{\eta}(\mathbf{x}^*, \boldsymbol{\theta})$ and $\boldsymbol{\delta}(\mathbf{x}^*)$ in (3.8) and (3.10), only draws from $\mathbf{w}(\mathbf{x}^*, \boldsymbol{\theta})$ and $\mathbf{v}(\mathbf{x}^*)$ need to be produced given a posterior draw of $(\lambda_\eta, \boldsymbol{\lambda}_w, \boldsymbol{\rho}_w, \lambda_y, \boldsymbol{\lambda}_v, \boldsymbol{\rho}_v, \boldsymbol{\theta})$.

Following the drivation in (3.15), the basis coefficients can be drawn conditional on the reduced data, $\hat{\mathbf{z}}$, rather than the full data $\mathbf{y}$ and $\boldsymbol{\eta}$, reducing computational costs.

We have that $\left(\begin{bmatrix} \hat{\mathbf{z}} \\ \mathbf{v}(\mathbf{x}^*) \\ \mathbf{w}(\mathbf{x}^*, \boldsymbol{\theta}) \end{bmatrix} | \lambda_\eta, \lambda_w, \rho_w, \lambda_y, \lambda_v, \rho_v, \theta \right)$ has a multivariate normal distribution with mean $\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}$ and covariance matrix $\begin{bmatrix} \boldsymbol{\Sigma}_{\hat{z}} & \boldsymbol{\Sigma}_{\hat{z},v*} & \boldsymbol{\Sigma}_{\hat{z},w*} \\ \boldsymbol{\Sigma}_{v*,\hat{z}} & \mathrm{diag}(\lambda_{vi}^{-1}\mathbf{I}_{|G_i|}) & \mathbf{0} \\ \boldsymbol{\Sigma}_{w*,\hat{z}} & \mathbf{0} & \mathrm{diag}(\lambda_{wi}^{-1}) \end{bmatrix}$, where there is correlation between $\hat{\mathbf{z}}$ and $\mathbf{v}(\mathbf{x}^*)$ because of the correlation between $\mathbf{v}$ and $\mathbf{v}(\mathbf{x}^*)$ and the fact that $\mathbf{v}$ is part of the composition of $\hat{\mathbf{z}}$. Similarly, there is correlation between $\hat{\mathbf{z}}$ and $\mathbf{w}(\mathbf{x}^*, \boldsymbol{\theta})$ because $\hat{\mathbf{z}}$ is a linear combination of $\mathbf{y}$ and $\boldsymbol{\eta}$, both of which are made up of $\mathbf{w}(\mathbf{x}^*, \theta)$. From the joint Gaussian structure, the distributions of $\mathbf{v}(\mathbf{x}^*)$ and $\mathbf{w}(\mathbf{x}^*, \boldsymbol{\theta})$ are straightforward from normal theory.

In similar fashion to above, posterior predictions of the process $\boldsymbol{\eta}(\cdot, \cdot)$ can be made at any inputs $(\mathbf{x}^*, \mathbf{t}^*)$. The reduced data can be expressed as $\hat{\mathbf{w}} = (\mathbf{K}^T\mathbf{K})^{-1}\mathbf{K}^T\boldsymbol{\eta}$. Conditional on the parameters $(\lambda_\eta, \lambda_w, \rho_w)$, $\hat{\mathbf{w}}$ and predictions $\mathbf{w}(\mathbf{x}^*, \mathbf{t}^*)$ have a jointly Gaussian distribution

$$\begin{bmatrix} \hat{\mathbf{w}} \\ \mathbf{w}(\mathbf{x}^*, \mathbf{t}^*) \end{bmatrix} \sim N\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \boldsymbol{\Sigma}_{\hat{w}} & \boldsymbol{\Sigma}_{\hat{w},w*} \\ \boldsymbol{\Sigma}_{w*,\hat{w}} & \mathrm{diag}(\lambda_{wi}^{-1}) \end{bmatrix} \right) \tag{3.17}$$

and are straightforward to sample.

3.3 Treed Gaussian Process

A potential issue in the use of Gaussian Process emulators is that, for some models, the assumption that the Gaussian Process has the same covariance structure throughout the entire input space is too strong. For these instances primarily and to avoid other potential disadvantages, Gramacy and Lee (2008) proposed use of a treed Gaussian Process on a partitioned input space.

In this approach, the input space is partitioned into sets (through a variety of algorithms) and a separate stationary Gaussian Process model is fit within each set. The motivating problem for this approach was modeling the Langley glide-back booster's return to Earth from space. The idea is that, as the booster gets closer to the earth, gravitational and atmospheric conditions change, so the settings in which the booster is traveling are distinctly different at various points in its descent.

3.3.1 Specification

Let $\mathcal{X}$ represent the input space for a model. Using the same notation in the previous section, consider observations in a system of interest $y(\mathbf{x}_i)$ at input settings $\mathbf{x}_i$. The observations are again modeled as $y(\mathbf{x}_i) = \eta(\mathbf{x}_i, \boldsymbol{\theta}) + \delta(\mathbf{x}_i) + \epsilon(\mathbf{x}_i)$. In settings (often spatial problems) where conditions vary significantly across the input space, a partition model is a way to provide the necessary flexibility. This method divides up the input space and fits separate base models to the data independently within each region.

Treed partition models often use binary splits on the value of a single variable to divide the input space (e.g. split the input space $\mathcal{X}$ based on the first dimension, creating $P_1 = \{x : x_1 < .5\}$ and $P_2 = \{x : x_1 \geqslant .5\}$), and are done recursively. This leads to partition boundaries which are parallel to coordinate axes which gives simple, ordered partition regions and allows for generalization to non-binary splits since multiple splits may be made on the same variable. Each partition, a leaf of the tree, has an independent model applied to it. The classification and regression tree (CART) method of Breiman et al. (1984) is a frequently used treed partition model that fits a constant surface on each leaf of the tree. Chipman et al. (1998) proposed a Bayesian approach to CART where a meaningful prior is specified for the size of the tree. Chipman et al. used a tree-generating process for the prior which stipulated a minimum amount of data within each leaf to infer on parameters. This method begins with all data in one region, and the region will split with probability $a(1 + q_\ell)^{-b}$ where q_ℓ is the depth of the region (node) and the a and b parameters are chosen to provide the desired spread of trees. The prior for the splitting process comes from choosing the splitting dimension d_s from a discrete uniform on $[1, ..., d_x]$ where d_x is the dimension of an input x, and then choosing a splitting location s_{d_s} uniformly on the range of the of x_{d_s}.

3.3.2 Treed model

A tree $\mathcal{T}$ partitions the input space $\mathcal{X}$ into R nonoverlapping regions $\{r_k\}_{k=1}^{R}$ by recursion. Define n_k to be the number of observation within a region r_k. Let $\mathbf{X}_k$ be a $n_k \times p_x$ matrix of the inputs in region r_k, and let $\mathbf{Y}_k$ be a $n_k \times p_y$ matrix of the outputs (or observations) in region r_k. Each row of $\mathbf{X}_k$, $\mathbf{x}_{k,j}$, j=1,...,n_k, is the p_x dimensional set of inputs for the corresponding row of $\mathbf{Y}_k$, $\mathbf{y}_{k,j}$, j = 1,...,n_k. Define D_k to be the data pairs $D_k = (\mathbf{X}_k, \mathbf{Y}_k)$ within region r_k. Let $p_0 = p_x + 1$ be the dimension of an element of the input space plus an intercept and define the $n_k \times p_0$ matrix $\mathbf{F}_k = (\mathbf{1}_{n_k}, \mathbf{X}_k)$. Within each region r_k, the GP model is generated in a hierarchical fashion by:

$$\mathbf{Y}_k|\boldsymbol{\beta}_k, \sigma_k^2, \mathbf{K}_k \sim N_{n_k}(\mathbf{F}_k, \boldsymbol{\beta}_k, \sigma_k^2\mathbf{K}_k)$$
$$\boldsymbol{\beta}_0 \sim N_{p_0}(\boldsymbol{\mu}, \mathbf{B})$$
$$\boldsymbol{\beta}_k|\sigma_k^2, \tau_k^2, \mathbf{W}, \boldsymbol{\beta}_0 \sim N_{p_0}(\boldsymbol{\beta}_0, \sigma_k^2\tau_k^2\mathbf{W})$$
$$\tau_k^2 \sim IG(\alpha_\tau/2, q_\tau/2)$$
$$\sigma_k^2 \sim IG(\alpha_\sigma/2, q_\sigma/2)$$
$$\mathbf{W}^{-1} \sim Wish((\rho\mathbf{V})^{-2}, \rho)$$

where $\mathbf{K}_k$ is the correlation matrix consistent with previously specified correlation structure. Gramacy and Lee treated the hyperparameters $\boldsymbol{\mu}, \mathbf{B}, \rho, \alpha_\tau, q_\tau, \alpha_\sigma, q_\sigma$ as known, since inference on them is not of interest in this case.

In the preceding hierarchical specification, there is no assurance that the process near the boundary of adjacent regions will be continuous across partitions. This is useful in allowing one to fit a discontinuous surface. In applications where the the processes should be continuous across partitions, smoothness can be induced through model averaging.

3.3.3 Estimation

Within each region, the data D_k are used to update the Gaussian Process parameters $\boldsymbol{\theta}_k = \{\boldsymbol{\beta}_k, \sigma_k^2, \mathbf{K}_k, \tau_k^2\}$ for $k = 1, ..., R$. The upper level parameters which are not region-specific, i.e., $\boldsymbol{\theta}_0 = \{\mathbf{W}, \boldsymbol{\beta}_0\}$, are also updated. Conditional on the tree from partitions, $\mathcal{T}$, the full set of parameters is $\boldsymbol{\theta} = \boldsymbol{\theta}_0 \cup \bigcup_{k=1}^{R} \boldsymbol{\theta}_k$. Samples from the posterior distribution of $\boldsymbol{\theta}$ are drawn using MCMC by first generating $\boldsymbol{\theta}_k|\boldsymbol{\theta}_0$ for $k = 1, ..., R$ conditional on some initial set $\boldsymbol{\theta}_0$ and then generating $\boldsymbol{\theta}_0|\bigcup_{k=1}^{R} \boldsymbol{\theta}_k$.

Conditional on a tree $\mathcal{T}$, the parameters for the Gaussian Process can be sampled by Gibbs sampling as a result of their conjugacy. Following from the hierarchical specification, the regression parameters $\boldsymbol{\beta}_0$ and $\boldsymbol{\beta}_k$ have the following full conditional distributions:

$$\boldsymbol{\beta}_0|\mathbf{Y}_1,...,\mathbf{Y}_R,\tau_1^2,...,\tau_R^2,\sigma_1^2,...,\sigma_R^2,\tilde{\boldsymbol{\beta}}_1,...,\tilde{\boldsymbol{\beta}}_R\mathbf{W} \sim N_m(\tilde{\boldsymbol{\beta}}_0,\mathbf{V}_{\tilde{\beta}_0})$$

$$\boldsymbol{\beta}_k|\mathbf{Y}_k,\tau_k^2,\sigma_k^2,\mathbf{W},\boldsymbol{\beta}_0 \sim N_m(\tilde{\boldsymbol{\beta}}_k,\sigma_k^2\mathbf{V}_{\tilde{\beta}_k})$$

where

$$\begin{aligned}
\tilde{\boldsymbol{\beta}}_0 &= \mathbf{V}_{\tilde{\beta}_0}(\mathbf{B}^{-1}\boldsymbol{\mu} + \mathbf{W}^{-1}\sum_{k=1}^{R}\boldsymbol{\beta}_k(\sigma_k\tau_k)^{-2}) \\
\mathbf{V}_{\tilde{\beta}_0} &= (\mathbf{B}^{-1} + \mathbf{W}^{-1}\sum_{k=1}^{R}(\sigma_k\tau_k)^{-2})^{-1} \\
\tilde{\boldsymbol{\beta}}_k &= \mathbf{V}_{\tilde{\beta}_k}(\mathbf{F}_k^T\mathbf{K}_k^{-1}\mathbf{Y}_k + \mathbf{W}^{-1}\boldsymbol{\beta}_0/\tau_k^2) \\
\mathbf{V}_{\tilde{\beta}_k} &= (\mathbf{F}_k^T\mathbf{K}_k^{-1}\mathbf{F}_k + \mathbf{W}^{-1}/\tau_k^2)^{-1}.
\end{aligned}$$

The regional linear variance τ_k^2 will have inverse gamma full conditional distribution:

$$\tau_k^2|\mathbf{Y}_k,\boldsymbol{\beta}_0,\boldsymbol{\beta}_k,\mathbf{W},\sigma_k^2 \sim IG((\alpha_\tau + m)/2,(q_\tau + (\boldsymbol{\beta}_k - \boldsymbol{\beta}_0)^T\mathbf{W}^{-1}(\boldsymbol{\beta}_k - \boldsymbol{\beta}_0)/\sigma_k^2)/2).$$

The regional linear covariance matrix $\mathbf{W}$ will have an inverse Wishart full conditional distribution:

$$\mathbf{W}^{-1}|\mathbf{Y}_1,...,\mathbf{Y}_R,\boldsymbol{\beta}_0,\boldsymbol{\beta}_1,...,\boldsymbol{\beta}_R,\sigma_1^2,...,\sigma_R^2,\tau_1^2,...,\tau_R^2 \sim W_m((\rho\mathbf{V} + \mathbf{V}_{\hat{W}})^{-1},\rho + R)$$

where

$$\mathbf{V}_{\hat{W}} = \sum_{k=1}^{R}\frac{1}{(\sigma_k\tau_k)^2}(\boldsymbol{\beta}_k - \boldsymbol{\beta}_0)(\boldsymbol{\beta}_k - \boldsymbol{\beta}_0)^T.$$

Obtaining a joint posterior distribution for $\mathbf{K}_k, \boldsymbol{\beta}_k$, and σ_k^2 is straightforward from the specification of $\mathbf{Y}_k$. Analytically integrating out $\boldsymbol{\beta}_k$ and σ_k^2 from this joint posterior provides a marginal posterior for $\mathbf{K}_k$ which, as illustrated by Berger et al. (2001), improves mixing of the Markov chain. This marginal density in (3.18), while not of the form of a familiar distribution, allows sampling of parameters of $\mathbf{K}_k$ through a Metropolis-Hastings algorithm. The marginal posterior is:

$$\pi(\mathbf{K}_k|\mathbf{Y}_k, \boldsymbol{\beta}_0, \mathbf{W}, \tau_k^2) = \left(\frac{|\mathbf{V}_{\tilde{\boldsymbol{\beta}}_k}|(2\pi)^{-n_k}}{|\mathbf{K}_k||\mathbf{W}|\tau_k^{2(m)}} \right)^{1/2}$$
$$\times \frac{(q_\sigma/2)^{\alpha_\sigma/2}\Gamma[(1/2)(\alpha_\sigma + n_k)]}{[(1/2)(q_\sigma + \psi_k)]^{(\alpha_\sigma+n_k)}\Gamma[\alpha_\sigma/2]} \times \pi(\mathbf{K}_k) \tag{3.18}$$

where

$$\psi_k = \mathbf{Y}_k^T \mathbf{K}_k^{-1} \mathbf{Y}_k + \boldsymbol{\beta}_0^T \mathbf{W}^{-1} \boldsymbol{\beta}_0 / \tau_k^2 - \tilde{\boldsymbol{\beta}}_k^T \mathbf{V}_{\tilde{\beta}_k}^{-1} \tilde{\boldsymbol{\beta}}_k.$$

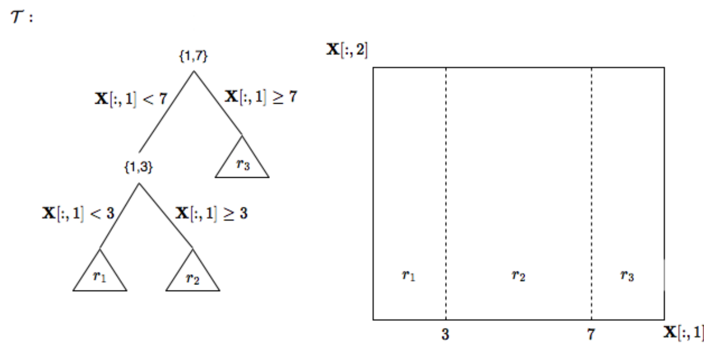
Finally, integrating out $\boldsymbol{\beta}_k$ from the joint posterior, as above, gives the marginal full condtional posterior for σ_k^2:

$$\sigma_k^2|\mathbf{Y}_k, \boldsymbol{\beta}_0, \mathbf{W}, \tau_k^2, \mathbf{K}_k \sim IG((\alpha_\sigma + n_k)/2, (q_\sigma + \psi_k)/2).$$

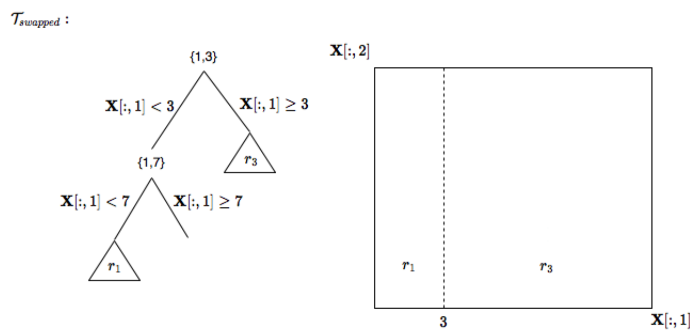
In order to integrate out dependence on the structure of the tree, reversible-jump MCMC is used. Gramacy and Lee use the tree operations of Chipman et al., change, swap, grow, prune, and they propose adding a rotate operation. For an existing split point $\{d_s, s_{d_s}\}$ in the input space, the *change* operation proposes shifting the value of s_{d_s} to the next greater or lesser split point $s_{d_s}^+$ or $s_{d_s}^-$ in the d_s dimension of $\mathbf{X}$. This is accomplished by sampling a value s' uniformly from the existing set of split points $\{d_s, s_{d_s}\}_{k=1}^{[R/2]} \times \{+, -\}$, causing the Metropolis-Hastings acceptance ratio for the operation to simplify to a likelihood ratio, since the parameters $\boldsymbol{\theta}_{r_k}$ in regions r_k lying below the split point $\{d_s, s'\}$, are held fixed. The *swap* operation proposes switching the order in which an adjacent parent-child node pair splits up the inputs. A parent-child pair is randomly selected from the tree and the splitting rules are swapped. Swaps when parent-child nodes split on the same variable can be problematic since the operation will force the child region below both to become empty. To avoid this, Gramacy and Lee proposed using the *rotate* operation from binary search trees. The rotate operation maintains splitting rules, but simply "rotates" the tree in a way that adjusts the configuration in a way similar to the swap operation, but keeps all existing nodes, thus eliminating the potential for creating an empty region, as shown in Figure 3.1. This operation also has the advantage that it encourages better mixing of the Markov chain because of the dynamic set of nodes it provides for pruning, which helps the chain avoid becoming stuck in local minima. The relevant part of the Matropolis-Hastings acceptance ratio for the rotate operation is the prior for $\mathcal{T}$ which prefers trees of less depth. For a given tree $\mathcal{T}$, let $I = \{I_i, I_l\}$ be the set of (internal and leaf) nodes which increase in depth after rotation and let $D = \{D_i, D_l\}$ be the set of (internal and leaf) nodes that decrease in depth after rotation. Then the Metropolis-Hastings acceptance ratio for a proposed tree $\mathcal{T}^*$ from rotation is:

$$
\begin{aligned}
\frac{p(\mathcal{T}^*)}{p(\mathcal{T})} &= \frac{\prod_{\ell \in I_i} a(2+q_\ell)^{-b} \prod_{\ell \in I_l} [1 - a(2+q_\ell)^{-b}]}{\prod_{\ell \in I_i} a(1+q_\eta)^{-b} \prod_{\ell \in I_l} [1 - a(2+q_\ell)^{-b}]} \\
&\times \frac{\prod_{\ell \in D_i} a q_\ell^{-b} \prod_{\eta \in D_l} [1 - a q_\ell^{-b}]}{\prod_{\ell \in D_i} a(1+q_\ell)^{-b} \prod_{\ell \in D_l} [1 - a(1+q_\ell)^{-b}]}
\end{aligned}
$$

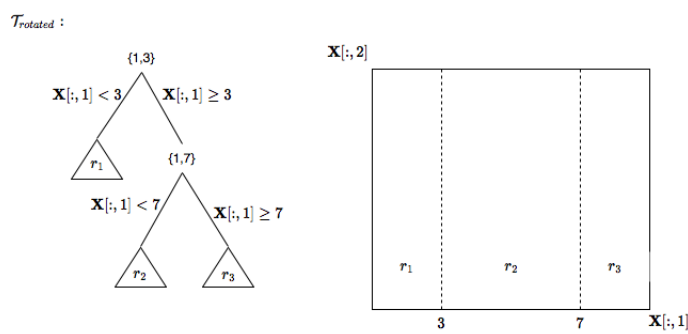
The *grow* and *prune* operations are the most complex because they add and remove, respectively, partations, which changes the dimension of the parameter space, hence the need for a reversible-jump algorithm. The grow operation begins by uniformly selecting a leaf node. When a new region r^* is added, new parameters $\{K(\cdot,\cdot), \tau^2\}_{r*}$ must be proposed and when a region is removed, the parameters must be absorbed by the parent node or discarded. The linear model parameters $\{\beta, \sigma^2\}$ are integrated out of the Metropolis-Hasting acceptance ratio. One of the children that is produced in the grow operation is uniformly chosen to inherit the correlation function of its parent $K(\cdot,\cdot)$. The other child then draws its correlation function from the prior, which results in the Jacobian term in the MCMC being 1. The prune operation follows a similar pattern to grow. It begins by uniformly selecting a parent of a pair of leaf nodes. Then, parameters from the correlation function are randomly



(a)



(b)



(c)

FIGURE 3.1: Diagram (left) and graphical (right) representations of arbitrary splits on the first dimension of a 2-dimensional input space, $\mathbf{X}$, along with the corresponding swap and rotate operations. 3.1(a) shows the tree, 3.1(b) shows how the swap operation leaves an empty node, and 3.1(c) shows the rotate operation.

selected for the consolidated (parent) node from one of the children.

After acceptance of a grow or prune operation, the variance for the new (or consolidated) region σ^2_{r*} can be drawn from its marginal posterior, followed by draws for β_{r*} and then the remainder of the parameters required for the region.

Let $\{\mathbf{X}_{r*}, \mathbf{Y}_{r*}\}$ be the data in the new parent node ℓ at depth q_η and let $\{\mathbf{X}_i, \mathbf{Y}_i\}$ $i = 1, 2$ be the data for the child at depth $q_\ell + 1$ resulting from a new split $\{d_{s*}, s_{d_{s*}}\}$. Let $\mathcal{P}$ and $\mathcal{G}$ be the set of nodes of $\mathcal{T}$ that are prunable and growable, respectively. After a grow operation takes place at node ℓ (resulting in the tree $\mathcal{T}'$) let $\mathcal{P}'$ be the set of prunable nodes of $\mathcal{T}'$; if the parent of node of ℓ is prunable in $\mathcal{T}$ (i.e. $\eta_p \in \mathcal{P}$ where ℓ_p denotes the parent node of ℓ), then $|\mathcal{P}'| = |\mathcal{P}|$. If $\eta_p \notin \mathcal{P}$, then $|\mathcal{P}'| = |\mathcal{P}| + 1$ since ℓ itself is now prunable and its parent was not. The Metropolis-Hastings acceptance ratio for the grow operation is

$$\frac{|\mathcal{G}|}{|\mathcal{P}'|} \frac{a(1+q_\ell)^{-b}(1-a(2+q_\ell)^{-b})^2}{1-a(1+q_\ell)^{-b}} \times \frac{p(\mathbf{K}_1|\mathbf{Y}_1, \boldsymbol{\beta}_0, \tau_1^2, \mathbf{W})p(\mathbf{K}_2|\mathbf{Y}_2, \boldsymbol{\beta}_0, \tau_2^2, \mathbf{W})}{p(\mathbf{K}_{r*}|\mathbf{Y}_{r*}, \boldsymbol{\beta}_0, \tau_{r*}^2, \mathbf{W})\pi(\mathbf{K}_2)}$$

where, as noted before, $\mathbf{K}_1$ is chosen randomly to receive the parameterization of $\mathbf{K}$, its parent, and the new parameters for $\mathbf{K}_2$ are proposed according to the prior π as in (3.18). The prune operation has an analogous acceptance ratio, where the parameters for $K(\cdot, \cdot)$ for the consolidated node ℓ are randomly chosen from one of the children being absorbed.

3.3.4 Prediction

Under Gramacy and Lee's treed Gaussian Process model, prediction differs slightly from the method specified by Higdon et al. due to the region-specific parameters. Prediction under the treed Gaussian Process model follows the theory of Hjort and More (1994). Let $\mathbf{x}^*$ be input values in a given region of the input space. Conditional on the structure of the regional covariance $\mathbf{\Sigma}_k$, a predicted observation value $y(\mathbf{x}^* \in$

r_k) has a normal distribution with mean $y(\hat{\mathbf{x}}^*)$ equal to

$$E[y(\mathbf{x}^*)|D_k, \mathbf{x}^*] = \mathbf{f}^T(\mathbf{x}^*)\tilde{\boldsymbol{\beta}}_k + \nu_k(\mathbf{x}^*)^T \boldsymbol{\Sigma}_k^{-1}(\mathbf{Y}_k - \mathbf{F}_k\tilde{\boldsymbol{\beta}}_k)$$

and variance $\sigma(\hat{\mathbf{x}}^*)$ equal to

$$var(\mathbf{y}(\mathbf{x}^*)|D_k, \mathbf{x}^*) = \sigma_k^2[\zeta(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{q}_k^T(\mathbf{x}^*)\mathbf{C}_k^{-1}\mathbf{q}_k]$$

where

$$\begin{aligned}
\mathbf{C}_k^{-1} &= (\boldsymbol{\Sigma}_k + \tau_k^2 \mathbf{F}_k \mathbf{W} \mathbf{F}_k^T)^{-1} \\
\mathbf{q}_k(\mathbf{x}^*) &= \nu_k(\mathbf{x}^*) + \tau_k^2 \mathbf{F}_k \mathbf{W} \mathbf{f}(\mathbf{x}^*) \\
\zeta(\mathbf{x}^*, \mathbf{x}') &= \boldsymbol{\Sigma}_k(\mathbf{x}^*, \mathbf{x}') + \tau_k^2 \mathbf{f}_k^T(\mathbf{x}^*) \mathbf{W} \mathbf{f}(\mathbf{x}') \\
\mathbf{f}^T(\mathbf{x}^*) &= (1, \mathbf{x}^{*T}) \in \mathbb{R}^{1\times m}
\end{aligned}$$

and $\nu_k(\mathbf{x}^*)$ is a vector of length n_k $\nu_{k,j} = \boldsymbol{\Sigma}_k(\mathbf{x}^*, \mathbf{x}_j)$ for all $\mathbf{x}_j \in \mathbf{X}_k$.

As with the approach in Higdon et al. (2008) presented previously, Gramacy and Lee (2008) suggest translating and scaling the input $\mathbf{X}$ so that it lies in an d_x-dimensional unit hypercube.

3.4 Emulator Diagnostics

3.4.1 Overview

Regarding the specification of the Gaussian Process emulator, it is possible that the emulator will predict simulator output poorly. This could be due to the assumption of a Gaussian process being inappropriate or poor choices of parameters obtained from the training data. Bastos and O'Hagan (2009) dealt with the issue of problems in emulator predictions and methods to fix these problems.

Using a more general parametrization than was presented in this overview on emulation, let $\eta(\mathbf{x})$ represent the output of a computer model where $\mathbf{x}$ are the inputs.

This is, as before, treated as a Gaussian Process, with a mean function $m(x)$ and covariance function given by $V(x, x') = \sigma^2 C(x, x'; \psi)$, where σ^2 is an unknown scaling parameter and $C(x, x'; \psi)$ is a known correlation function, where ψ is a vector of unknown correlation parameters. The mean function is $m(x) = h(x)^T \beta$ where the function h maps from the the input space of x to $\mathbb{R}^q$ where the dimension of the range, q, is not necessarily the same as the dimension of the input space, and β is a vector of unknown coefficients. Here, the non-zero mean is essentially combining the previously described simulator and discrepancy measure into one function. By incorporating observed simulator outputs at training data points, the mean and covariance of η can be updated to obtain posteriors.

3.4.2 Simulator Validation

In problems involving use of emulators, there is growing a growing focus on validation of the computer models that are being emulated. The idea being that, for the emulator to be useful, it must be giving results that closely model reality, hence the simulator it is modeling should be checked to see how well its results compare to observed data. This validation issue is important to ABMs due to the complexity of the sets of inputs within most realistic ABMs. While some methods for assessing goodness-of-fit of ABMs exist, and will be discussed later, Bayarri et al. (2007b) proposed a general six-step process for computer model validation which can easily be applied to ABMs.

The objective of this approach is to assess simulator output by producing tolerance bounds for the output that correspond to the true process being modeled, and then testing whether the true process lies within these bounds. This assessment takes into account sources of uncertainty and model bias, and, hence is application-specific. The formulation of the tolerance bounds will be discussed at the end of the validation process.

Step 1: Input/Uncertainty Map

An input/uncertainty (I/U) map is a way of organizing information about model input and their related uncertainties. This map has four main features:

1. Model inputs which are potentially important to simulator output are listed;
2. Each input is ranked by its importance;
3. The uncertainty associated with each input is given using; the range of possible values or a distribution;
4. The current status of each input, including how it treated in the model, is given.

The I/U map is dynamic and, in particular, features 2, 3 and 4 may change throughout the validation process. Bayarri proposed using a 5-point scale, in which a rating of 1 represents minimal impact on prediction error and a rating of 5 represents significant potential impact, to rank the importance of inputs. The determination of significance of impact is ad-hoc, but this scale becomes particularly important in identifying important inputs in applications when the list of possible inputs is very large. The initial importance rating of each input (at the beginning of the validation process) is based on experience and/or expert advice for the particular application. This is to give a general idea of the influence of each input on the simulator output and the range of input uncertainty.

Inputs about which very little is known should be further investigated during the process, but it may not be possible to do this effectively until the other, better-understood, inputs have been taken into account. The numerical accuracy of the model should be addressed before this validation process begins and, in cases of a lack of convergence, the resulting error becomes a part of the model bias.

Step 2: Determine Evaluation Criteria

Model evaluation should be based on some specific criteria defined on model output. Specific diagnostics which could be used for this evaluation are discussed in detail in the context of emulator diagnostics later in this section. Additionally, the relevant domain of input variables over the model to be evaluated should be specified in order to ensure the model is giving "good" results at inputs relevant to the real process being modeled. Combining multiple evaluation criteria in assessing overall model performance is of use, leading to subsequent analysis on the appropriate scope of applications of the model as well as the range of reliable predictions of the model.

Step 3: Data Collection

Data from field experiments and computer experiments are a key element of the validation process. Field data, while crucial, are often difficult to obtain since it represents the actual process of interest. The data collected serve as a supplement to historical data on the process of interest. Once collected, the data will allow for comparison of model output to the emulator to assess bias and uncertainty in model predictions, as well as sensitivity of the model to inputs. The collected data additionally aids in identifying problematic components of the model being validated. A final purpose for the data is for use in development of approximations to computationally expensive model code, which will be discussed subsequently.

In order for sufficient data to accomplish all of these objectives, multiple stages of experiments must be conducted. Such experiments should cover important ranges of the input space, which can be done effectively, as suggested in Higdon et al.'s approach, by scaling inputs to a unit hypercube.

Step 4: Model Approximation

The computer models used in the applications of interest are often computationally expensive, and running their code directly throughout the validation process is not feasible. To address this, an approximation to the model is used, in the form of a surrogate model. The usual surrogate model, as described previously, is a Gaussian Process emulator. Denote the output from the computer model at input $\mathbf{x}$ and calibration parameters $\boldsymbol{\theta}$ by $y^M(\mathbf{x}, \boldsymbol{\theta})$. Then the approximation of $y^M(\mathbf{x}, \boldsymbol{\theta})$ will be $\eta(\mathbf{x}, \boldsymbol{\theta})$, the Gaussian process at the given inputs. Formulation of the emulator follows the process laid out previously, so details are omitted here. In this context, however, the emulator is being used as a tool in the validation process for the simulator, and is not the focal point of the development.

The simulator approximation can be used to obtain estimates of important model parameters to be used later in the validation process. Bayarri found that plug-in MLE extimates of the paramters (typically representing input settings or calibration parameters) gave similar results to a fully Bayesian analysis, and suggested using MLE estimates for the parameters together with a Bayesian analysis of the validation and prediction process.

Step 5: Analysis of Model Output

Since the computer model output $y^M(\mathbf{x}, \boldsymbol{\theta})$ is an approximation of reality, there is the possibility of some bias, as mentioned earlier. In the notation that Higdon et al. used in the discussion of emulator specification, let $\delta(\mathbf{x})$ be the bias function, representing the discrepancy between the computer model and the true process. Denoting behavior of the true process as $y^R(\mathbf{x})$ gives the following relation:

$$y^R(\mathbf{x}) = y^M(\mathbf{x}, \boldsymbol{\theta}) + \delta(\mathbf{x}). \tag{3.19}$$

Field data at a given set of inputs $\mathbf{x}_i$, $y^F(\mathbf{x}_i)$ is considered to be a more accurate representation of reality, with some measurement error, giving the following relation:

$$y^F(\mathbf{x}_i) = y^R(\mathbf{x}_i) + \epsilon_i^F \tag{3.20}$$

where the ϵ_i^F are iid normal with mean zero (indicating no bias in field measurements) and precision λ^F.

To complete a Bayesian model, priors on the calibration paramters $\boldsymbol{\theta}$, field data precision λ^F, and the discrepancy/basis function $\delta(\mathbf{x})$ must be specified. The prior for $\boldsymbol{\theta}$, $\pi(\boldsymbol{\theta})$, is specified in the I/U map, as is usually taken to be uniform on some range. Bayarri et al. propose to let $\pi(\lambda^F)$ be exponential and, as in the earlier specification, the prior for $\delta(\mathbf{x})$ is a Gaussian Process.

In cases where the computer model runs reasonably fast and $y^M(\mathbf{x}, \boldsymbol{\theta})$ can be computed quickly, Bayesian analysis can proceed directly, otherwise the emulator $\eta(\mathbf{x}, \boldsymbol{\theta})$ must be incorporated into the analysis. In the latter case, the parameters for the Gaussian Process need to be added to the list of unknowns for complete analysis, or the plug-in MLE estimates can be used.

The bias function $\delta(\mathbf{x})$ will be assumed to have either a mean of 0, as before, or an unknown constant mean μ^b. For the Bayesian analysis in the case where $y^M(\mathbf{x}, \boldsymbol{\theta})$ can be computed quickly, (3.19) and (3.20) combine to give

$$y^F(\mathbf{x}) = y^M(\mathbf{x}, \boldsymbol{\theta}) + \delta(\mathbf{x}) + \epsilon^F$$

indicating a multivariate normal density for the collection of field data, $f(\mathbf{y}^F|\boldsymbol{\theta}, \lambda^F, \delta)$. Denoting the prior distribution for $\boldsymbol{\theta}$, λ^F and $\delta(\mathbf{x})$ by $\pi(\boldsymbol{\theta}, \lambda^F, \delta(\mathbf{x}))$, the posterior of these quantities, given the field data, is

$$\pi(\boldsymbol{\theta}, \lambda^F, \delta(\mathbf{x})) \propto f(\mathbf{y}^F|\boldsymbol{\theta}, \lambda^F, \delta) \times \pi(\boldsymbol{\theta}, \lambda^F, \delta(\mathbf{x})).$$

The posterior distribution will be determined by MCMC results.

The complete analysis requires evaluation of y^M at all generated inputs/settings $\mathbf{x}, \boldsymbol{\theta}$, which is not is not feasible when the computer model is expensive, in which case the emulator η should be used, making use of the latin hypercube space-filling technique mentioned earlier. The use of the approximation η introduces additional uncertainty into predictions.

In the interest of achieving a stable MCMC algorithm, Bayarri et al. (2007b) proposed an approach called *modular MLE*. First, Gaussian process parameters are determined only from data based on the computer model, not field experiment data. Secondly, fix the Gaussian Process hyperparamteres at their MLEs and leave only precisions and calibration paramteres random. Both of these reduce computational burden, but also result in an analysis that is no longer fully Bayesian. Bayarri et al. found that predictions from this analysis to be close to those from a fully Bayesian analysis.

The MCMC analysis produces draws from the posterior distributions of $\boldsymbol{\theta}$, λ^F, $\delta(\mathbf{x})$, and y^M. The posterior distributions of relevant quantities can be estimated from these posterior samples.

The MCMC draws can be used to produce predictions with corresponding uncertainty for the estimates and inputs, which allows assessment of the accuracy of predictions and, hence utility of the computer model. To predict the behavior of the true process at a new set of inputs $\mathbf{x}^*$, all that is required are draws from the posterior predictive distribution of $\mathbf{y}^{R*}$, $\pi(\mathbf{y}^{R*}|\mathbf{y}^F, \mathbf{y}^M)$. By (3.19), such draws are obtained from draws from the joint posterior predictive distribution of $\mathbf{y}^M$ and δ, denoted by $\mathbf{y}^{M(i)}$ and $\delta^{(i)}$, at the new set of inputs. This prediction for the true process $\hat{\mathbf{y}}^{R*}$ is equal to the estimate of the posterior predictive mean of $\mathbf{y}^{R*}$,

$$\hat{\mathbf{y}}^{R*} = \frac{1}{N}\sum_{i=1}^{N}[\mathbf{y}^{M(i)} + \delta^{(i)}],$$

at the new set of inputs and, in the case where the model involves calibration parameters θ, use an estimate, $\hat{\theta}$, based on previous data.

The covariance matrix corresponding to this predictor can be estimated by

$$\hat{cov}(\hat{\mathbf{y}}^{R*}) = \frac{1}{N}\sum_{i=1}^{N}[\hat{\mathbf{y}}^{R*} - (\mathbf{y}^{M(i)} + \delta^{(i)})] \times [\hat{\mathbf{y}}^{R*} - (\mathbf{y}^{M(i)} + \delta^{(i)})]^T.$$

In regard to obtaining tolerance bounds for model output as discussed earlier, the objective is, for a given probability γ, to find a τ such that the prediction is within τ of the true $y^R(\mathbf{x})$. So, obtaining N predictions of the simulator, an estimate of τ can be obtained by finding the value for which $\gamma \times 100\%$ of samples satisfy

$$|[\hat{\mathbf{y}}^{R*} - (\mathbf{y}^{M(i)} + \delta^{(i)})| < \tau$$

This method can be generalized to find asymmetric tolerance bounds by finding (τ_1, τ_2) such that $\gamma \times 100\%$ of samples satisfy

$$\mathbf{y}^{M(i)} + \delta^{(i)} - \tau_1 < \hat{\mathbf{y}}^{R*} < \mathbf{y}^{M(i)} + \delta^{(i)} + \tau_2$$

subject to component-wise minimization of $\tau_1 + \tau_2$.

Step 6: Feedback and Feed-forward

Steps 4 and 5 produce analysis that can be used to update the I/U map. These steps give feedback on inputs whose uncertainty needs to be reduced, regions of the model needing closer examination and possible revisions of model evaluation criteria. The feed-forward component is making use of the analysis to predict the accuracy of new models related to the model being developed but for which no field data are available.

3.4.3 Diagnostics for Linear Models

The process for validating a Gaussian Process emulator is analogous to that of validating a linear model, by using residuals. In order to obtain residuals for an emulator, a new data set (separate from the values in the training data) must be used.

Let $\mathbf{X}^v = \{\mathbf{x}_1^v, ..., \mathbf{x}_m^v\}$ represent an unobserved set of inputs, called the validation input data. The m-dimensional output from the simulator at these validation inputs is $\mathbf{y}^v = \eta(\mathbf{X}^v) = (\eta(\mathbf{x}_1^v), ..., \eta(\mathbf{x}_m^v))$. In order to ensure that the emulator accurately represents the simulator throughout the input space, the validation input data should be selected to cover the region of the input space over which the emulator will be used.

A general diagnostic $D(\cdot)$ is a function of the validation data output, $\mathbf{y}^v$, which is used to compare $D(\mathbf{y}^v)$ to the reference distribution $D(\eta(\mathbf{X}^v))$. Values of $D(\mathbf{y}^v)$ which fall in appropriately chosen regions with low probability indicate possible conflict between the emulator and the simulator. If no values of the diagnostic fall into regions suggesting conflict, then this suggests that the emulator accurately represents the simulator.

Individual prediction errors for the validation data are the differences between the observed simulator output y_i^v and the predictive mean output $E[\eta(x_i^v)|\mathbf{y}]$ at the same inputs.

The standardized prediction errors can be used as a diagnostic:

$$D_i^I(\mathbf{y}^v) = \frac{y_i^v - E[\eta(x_i^v)|\mathbf{y}]}{\sqrt{V[\eta(x_i^v)|\mathbf{y}]}}.$$

If the emulator properly represents the simulator, then the validation output has approximate mean $E[\eta(x_i^v)|\mathbf{y}]$ and an estimate of its variance is $V[\eta(x_i^v)|\mathbf{y}]$, in which case D_i^I has a Student t distribution with $n-1$ degrees of freedom. In most cases, there is enough training data that the degrees of freedom of this distribution are sufficiently large to consider D_i^I to have standard normal distributions. In light of this normal approximation, values of $|D_i^I(\mathbf{y}^v)| > 2$ indicate conflict between the emulator and simulator. If a single isolated error is obtained, further training data can be obtained in the input space near this value to investigate emulator agreement with the simulator agreement at this location.

A high number of large values of D_i^I indicate a more serious problem with the emulator. Large errors of the same sign in a particular region suggest a problem with the mean function $m(\cdot)$ or perhaps, suggest that the stationary model used may not be the correct for this region, in which case a treed Gaussian Process, as discussed earlier, can be implemented to give better fit.

If there are large errors at validation input values which are close to training data inputs, the correlation structure may be poorly chosen, causing predictions to be over-influenced by nearby training data.

Mahalanobis Distance

It is often desirable to be able to present a single summary diagnostic. Under the assumption of independence of outputs, a χ^2 distribution could be obtained by summing $D_i^I(\mathbf{y}^v)^2$. However, the assumption of independence is too strong to make since, for example, a simulator which is a smooth function would cause outputs from input values close together in the input space to be similar.

A summary diagnostic which allows for correlated outputs is the Mahalanobis distance between the emulator predictions and simulator output at input $\mathbf{x}^v$:

$$D_{MD}(\mathbf{y}^v) = (\mathbf{y}^v - E[\eta(\mathbf{x}^v)|\mathbf{y}])^T \times (V[\eta(\mathbf{x}^v)|\mathbf{y})^{-1} \times (\mathbf{y}^v - E[\eta(\mathbf{x}^v)|\mathbf{y}])$$

. Extreme values of $D_{MD}(\mathbf{y}^v)$, both large and small, indicate conflict between the emulator and simulator. In this case, individual errors should be investigated for any patterns to assess underlying issues and possible problems in specific regions of the input space.

Under the assumptions of Gaussian Processes, $D_{MD}(\eta(\mathbf{x}^v))$ is proportional to a random variable with an $F_{m,n-q}$ distribution. In order to explore the errors, first take decompose $V[\eta(\mathbf{x}^v)|\mathbf{y}]$ to the form $V[\eta(\mathbf{x}^v)|\mathbf{y}] = GG^T$, so a Cholesky or eigen-decomposition will suffice. This yields a standard deviation matrix G. After taking this decomposition, a new diagnostic can be constructed

$$D_G(\mathbf{y}^v) = G^{-1}(\mathbf{y}^v - E[\eta(\mathbf{x}^v)|\mathbf{y}])$$

which is an m-vector of transformed errors which have been scaled to be uncorrelated with variances of 1. In the case where the outputs can be assumed to be normal, each of the transformed errors will have a Student t distribution with $(n-q)$ degrees of freedom where q is the dimension of the coefficients β. Note that $D_G(\mathbf{y}^v)^T D_G(\mathbf{y}^v) = D_{MD}(\mathbf{y}^v)$. Hence, the m elements of $D_G(\mathbf{y}^v)$ can be used to look for patterns of extreme values in particular regions of the input space.

For the particular decomposition to use, Bastos and O'Hagan recommend using a Pivoted Cholesky decompostion, which combines the benefits of both the eigendecomposition as well as the Cholesky decomposition. Under this decomposition, the data are permuted so that the first element has the largest marginal variance, the second element has the largest predictive variance conditional on the first variance, and so on. This corresponds to $P(\mathbf{y}^v - \eta(\mathbf{x}^v))$ where P is a permutation matrix giving the desired arrangement of the data. Then, if the standard Cholesky decomposition of $V[\eta(\mathbf{x}^v)|\mathbf{y}] = AA^T$, then the pivoted cholesky decomposition gives $PV[\eta(\mathbf{x}^v)|\mathbf{y}]P^T = PAA^TP^T = (PA)(PA)^T = GG^T$. Hence, the square root matrix G now has the form $G = PA$. Now, denote the elements of the vector $D_G(\mathbf{y}^v) = D_i^{PC}(\mathbf{y}^v)$ for $i = 1, ..., m$ as Pivoted Cholesky errors.

A group of extreme values for $D_i^{PC}(\mathbf{y}^v)$ at the beginning of the vector suggest heterogeneity, perhaps requiring use of a treed approach, whereas a large number of extreme errors in the latter part of the vector suggests a problem with the correlation structure. Because each $D_i^{PC}(\mathbf{y}^v)$ corresponds to a particular validation input point, this allows straightforward examination of individual errors.

Graphical Methods

The diagnostics developed above can be further used for studying the quality of emulator predictions using graphical displays. One choice of graphical diagnostic is to plot standardized individual prediction errors $D_i^I(\mathbf{y}^v)$ against the emulator's predictions. Patterns in this graphic suggest a problem with the mean function. Systematic errors of the same sign in specific regions of the output suggest a misspecified mean function or poor coefficient estimation. Heteroscedasticity suggests a violation of the assumption of stationarity, in which case a treed approach should be applied to see if fitting separate stationary processes gives better results. Individual errors of large absolute value suggest that predictive variance is too small, whereas individual errors close to 0 suggest excessively large variance. Plotting $D_i^T(\mathbf{y}^v)$ against the index would give similar intrepretation.

Another graphical display that provides a useful diagnostic uses $D^{PC}(\mathbf{y}^v)$. Recall that the elements of $D^{PC}(\mathbf{y}^v)$ have a Student t distribution with $n-q$ degrees of freedom. So, a QQ plot using this distribution is a useful diagnostic. Points lying close to the 45-degree line support the normality assumption of simulator outputs, and clusters of points away from the 45-degree line suggest a problem with the predictive variance. Any curvature in the QQ plot suggests non-normality of simulator outputs (meaning that a Gaussian Process is not a good choice to model them), and outliers suggest regional fitting problems, which can be addressed using a Treed Gaussian process approach.

A final graphic to use as a diagnostic is a plot of $D_i^I(\mathbf{y}^v)$ against the validation input values. This plot gives another means of assessing behavior regionally within the input space.

Other Diagnostics

Other diagnostics can be used to compare the simulator and emulator model. Based on the formulation of $\eta(\cdot)$, a $100\alpha\%$ credible interval, $CI_i(\alpha)$ can be formed for the output corresponding to any validation input. The diagnostic

$$D_{CI}(\mathbf{y}^v) = \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}(\mathbf{y}_i^v \in CI_i(\alpha))$$

represents the proportion of validation outputs that lie in their corresponding marginal credible interval. The diagnostic $D_{CI}(\mathbf{y}^v)$ would be expected to be close to α, but because of dependence between inputs, the distribution of D_{CI} is not binomial. The only way to compute the reference distribution for $D_{CI}(\eta(\mathbf{x}^v))$ is by Monte Carlo simulation. By drawing a large number of draws from the predictive distribution of $\eta(\mathbf{x}^v)|\mathbf{y}$, and computing $D_{CI}(\cdot)$ for each sample, this gives the empirical distribution of $D_{CI}(\cdot)$ which serves as a good approximation of the distribution $D_{CI}(\eta(\mathbf{x}^v))$.

3.5 First-Order Emulators

Hooten et al. (2011) proposed a somewhat simpler method of emulation, departing from the approach Higdon et al. (2008) put forth. The approach of approach Hooten et al., referred to as "first-order" emulation, is relevant to models where the parameters $\boldsymbol{\theta}$ have some physical or biological meaning, as opposed to (Higdon et al.'s) "second-order" emulators where these corresponding parameters are used for model calibration and have no physical meaning. In light of the fact that $\boldsymbol{\theta}$ have practical meaning, it is of interest in these models to perfrom inference on these parameters.

The idea behind this approach is to take K simulations of a computer experiment, denoted by $\mathbf{y}^{(j)}$ for $j = 1, ..., K$ where $\mathbf{y}^{(j)}$ is an n-dimensional vector. Let these K simulated outputs form the columns of a matrix $\mathbf{Y} = (\mathbf{y}^{(1)}, ..., \mathbf{y}^{(K)})$. Then, taking its singular-value decomposition, the output can be expressed as $\mathbf{Y} = \mathbf{UDV}^T$, $\mathbf{Y} \in \mathbb{R}^{n \times K}$, $\mathbf{U} \in \mathbb{R}^{n \times n}$, $\mathbf{D} \in \mathbb{R}^{n \times K}$, and $\mathbf{V} \in \mathbb{R}^{K \times K}$. Then, using this decomposition as well as the experimental parameters $\boldsymbol{\theta}$, a multidimensional, non-linear surface $\mathbf{v}(\boldsymbol{\theta})$ which spans the space of $\boldsymbol{\theta}$ can be considered, which then would allow for interpolation at other values of experimental parameters $\boldsymbol{\theta}^*$ in the support of $\boldsymbol{\theta}$. Hooten et al. aim to evaluate emulators based on first-order properties of $\mathbf{v}$. This approach is advantageous in many situations in that it is simpler to implement than second-order emulators.

3.5.1 Overview

Consider an approximation of actual observed data $\mathbf{y}$ based on the expression for computer model output $\mathbf{Y}$, $\mathbf{Y} = \mathbf{UDV}^T$:

$$\mathbf{y} = \mathbf{UDv}(\boldsymbol{\theta}) + \boldsymbol{\epsilon}. \tag{3.21}$$

Let the $p \times K$ dimensional matrix $\boldsymbol{\Theta} = (\boldsymbol{\theta}^{(1)}, ..., \boldsymbol{\theta}^{(K)})^T$ represent the set of experimental parameters for all of the K experimental output, and $\mathbf{Y}$ ($\boldsymbol{\theta}^{(i)}$ is the p-dimensional set of experimental parameters of experimental output $\mathbf{y}^{(i)}$). A predictive model can be developed for the surface $\mathbf{v}$ based on $\mathbf{V}$ and $\boldsymbol{\Theta}$ from the computer model. Following the approach of Higdon et al. (2008), Hooten et al. use a predictive model $\mathbf{v} \sim g(\boldsymbol{\Theta}, \boldsymbol{\beta})$ where $\boldsymbol{\Theta}$ are covariates and $\boldsymbol{\beta}$ are nuisance parameters. The model g can vary between applications, but should be capable of informing the predictive distribution of $\mathbf{v}$ for any value of $\boldsymbol{\theta}$.

3.5.2 Linear first-order emulators

Consider the simple case where the model g is linear and Gaussian. Letting the K-vector $\mathbf{v}_i$ represent the ith column of V, then $\mathbf{v}_i = \boldsymbol{\Theta}\boldsymbol{\beta}_i + \boldsymbol{\xi}_i$ where $\boldsymbol{\beta}_i$ is a $p \times 1$ vector and $\boldsymbol{\xi}_i \sim N(0, \tau^2 I_K)$. Without applying Bayesian methods, a least squares solution for $\boldsymbol{\beta}_i$ can be obtained as $\hat{\boldsymbol{\beta}}_i = (\boldsymbol{\Theta}^T\boldsymbol{\Theta})^{-1}\boldsymbol{\Theta}^T\mathbf{v}_i$. Using this solution, a prediction for $\mathbf{v}$, $\hat{\mathbf{v}}_i$, can be obtained at any parameter setting $\hat{\boldsymbol{\theta}}^*$ from $\hat{\mathbf{v}}_i = \hat{\boldsymbol{\theta}}^{*T}\hat{\boldsymbol{\beta}}_i$.

The coefficients for all K sets of nuisance parameters for the predictive model $\hat{\boldsymbol{\beta}}_1, ..., \hat{\boldsymbol{\beta}}_K$ can be combined into a single matrix $\hat{\mathbf{B}} = (\hat{\boldsymbol{\beta}}_1, ..., \hat{\boldsymbol{\beta}}_K)^T$ which will be equivalent to $(\boldsymbol{\Theta}^T\boldsymbol{\Theta})^{-1}\boldsymbol{\Theta}^T\mathbf{V}$. This, then, allows the model (3.21) to be expressed as:

$$\mathbf{y} = \mathbf{UD}\hat{\mathbf{B}}^T\boldsymbol{\theta} + \boldsymbol{\epsilon} \tag{3.22}$$

using $\hat{\mathbf{B}}^T\boldsymbol{\theta}$ as the expectation of $\mathbf{v}(\boldsymbol{\theta})$. Using this parametrization and the least squares estimate form of $\hat{\mathbf{B}} = (\boldsymbol{\Theta}^T\boldsymbol{\Theta})^{-1}\boldsymbol{\Theta}^T\mathbf{V}$ and the model for the computer output

$\mathbf{Y} = \mathbf{UDV}^T$, model (3.22) can be re-expressed as:

$$\begin{aligned}
\mathbf{y} &= \mathbf{UD}((\boldsymbol{\Theta}^T\boldsymbol{\Theta})^{-1}\boldsymbol{\Theta}^T\mathbf{V})^T\boldsymbol{\theta} + \boldsymbol{\epsilon} \\
&= \underbrace{\mathbf{UDV}^T}\boldsymbol{\Theta}(\boldsymbol{\Theta}^T\boldsymbol{\Theta})^{-1}\boldsymbol{\theta} + \boldsymbol{\epsilon} \\
&= \mathbf{Y}\boldsymbol{\Theta}(\boldsymbol{\Theta}^T\boldsymbol{\Theta})^{-1}\boldsymbol{\theta} + \boldsymbol{\epsilon} \qquad (3.23)
\end{aligned}$$

which illustrates that the mean of the observed response $\mathbf{y}$ is a weighted average of computer model output, $\mathbf{Y}$, where the weights are $(\boldsymbol{\Theta}^T\boldsymbol{\Theta})^{-1}\boldsymbol{\theta}$ where $\boldsymbol{\theta}$ must be estimated.

Depending on the structure of the matrices in the singular-value decomposition, it may be possible to truncate $\mathbf{UD}$ and $\mathbf{V}$ so that one keeps only the first $q < K$ columns of $\mathbf{UD}$, consequently using only the first q rows of $\mathbf{V}^T$. This will result in $\boldsymbol{\beta}$ consisting of q coefficients and the matrix of nuisance parameters $\tilde{\mathbf{B}}$ beecomes $q \times p$. So, the data could be modeled in similar fashion as before $\mathbf{y} = \widetilde{\mathbf{UD}}\tilde{\mathbf{B}}^T\boldsymbol{\theta} + \boldsymbol{\epsilon}$ where $\widetilde{\mathbf{UD}}$ is the truncation of $\mathbf{UD}$.

In the case where $\mathbf{B}$ is known or well-estimated, obtaining estimates of $\boldsymbol{\theta}$ is straightforward.

When the linear assumption for modeling $\mathbf{V}$ is valid, the model for observed data can be expressed as a combination of the previous formulations:

$$\begin{aligned}
\mathbf{y} &= \mathbf{UD}v(\boldsymbol{\theta}) + \boldsymbol{\epsilon} \\
&= \mathbf{UD}(\mathbf{B}^T\boldsymbol{\theta} + \boldsymbol{\xi}) + \boldsymbol{\epsilon} \\
&= \mathbf{UDB}^T\boldsymbol{\theta} + \mathbf{UD}\boldsymbol{\xi} + \boldsymbol{\epsilon}
\end{aligned}$$

where the variance of $\boldsymbol{\xi}$, τ^2, is known since it results from computer model output. The proportion of variance from the estimate of $\mathbf{v}$, $\nu = \frac{\tau^2}{\sigma^2+\tau^2}$ can be used as a measure of the quality of the predictive model for $\mathbf{v}$, where a smaller value of ν indicates the a close approximation to the computer model.

3.5.3 Non-linear first-order emulators

Now, assume a more general link between $\mathbf{v}$ and $\boldsymbol{\theta}$. So, let $\mathbf{v} \sim g(\boldsymbol{\Theta}, \boldsymbol{\beta})$ where g is a generic model. Now, the model for $\mathbf{y}$ is now:

$$\mathbf{y} = \mathbf{UDv} + \boldsymbol{\epsilon}$$

$$\mathbf{v} \sim g(\boldsymbol{\Theta}, \boldsymbol{\beta})$$

The main source of complication for this non-linear form over the linear form is that inference on $\boldsymbol{\theta}$ is less straightforward. Bayesian methods are useful in analysis.

The desired posterior distribution $\pi(\boldsymbol{\theta}, \sigma^2, \mathbf{v}|\mathbf{y})$ is proportional to the compositional form $\pi(\mathbf{y}|\mathbf{v}, \sigma^2)\pi(\mathbf{v}|\boldsymbol{\Theta}, \boldsymbol{\beta})\pi(\boldsymbol{\theta})\pi(\sigma^2)\pi(\boldsymbol{\beta})$. Draws from the posterior are made via MCMC. To aid in identifiability and to allow flexibility in the choice of g, a two-stage implementation can be used where first the model $\mathbf{V} \sim g(\boldsymbol{\Theta}, \boldsymbol{\beta})$ is fit using computer output and then used for predictions of $\mathbf{v}^*$. Then, the goal is to infer $\boldsymbol{\theta}$ and σ^2 conditional on the observed $\mathbf{y}$.

So, the posterior of interest is now:

$$\pi(\boldsymbol{\theta}, \sigma^2, \mathbf{v}|\mathbf{y}) \propto \int \pi(\mathbf{y}|\mathbf{v}, \sigma^2)\pi(\mathbf{v}|\boldsymbol{\theta})\pi(\boldsymbol{\Theta})\pi(\sigma^2)d\mathbf{v}$$

where $\pi(\mathbf{v}|\boldsymbol{\theta})$ is the predictive model for $\mathbf{v}$. This integration is simplified by the use of composition sampling in MCMC by means of Metropolis-Hastings. So, given a proposed value of $\boldsymbol{\theta}$, $\boldsymbol{\theta}^*$, it is sufficient to draw from $\pi(\mathbf{v}|\boldsymbol{\theta}^*)$. That draw will be used in the MH ratio to accept/reject $\boldsymbol{\theta}^*$. An inverse-gamma prior for σ^2 will result in a conjugate full-conditional distribution.

3.5.4 Implementation

In general cases for the link function, a variety of approaches can be used to approximate $g(\boldsymbol{\theta}, \boldsymbol{\beta})$. Hooten et al. use the nonparametric random forest approach following Breiman (2001) to link $\mathbf{V}$ to $\boldsymbol{\theta}$. One would fit a separate random forest model $g_i(\boldsymbol{\theta}, \boldsymbol{\beta})$ to each right singular vector $\mathbf{v}_i$. This nonlinear predictive model is then used to yield a prediction $\hat{\mathbf{v}}^*$ given some set of input parameters $\boldsymbol{\theta}$. Because of the sparsity of the model structure in the random forest method, Hooten et al. propose obtaining residuals $\hat{\mathbf{r}}^*$ over the entire set training parameters $\boldsymbol{\Theta}$ and then selecting a bootstrap sample of the residuals, $\mathbf{r}^*$. Then, one adds the boostrap residuals to the random forest predictions to obtain a quasi-realization $\mathbf{v}^* = \hat{\mathbf{v}}^* + \mathbf{r}^*$. This approach has the advantage of being based on average predictions, which have been shown to have both low bias and low variance (cf. Hastie et al., 2009). The residuals $\hat{\mathbf{r}}^*$ in this case are not the usual residuals representing the difference between an observation and fitted values, but actually represent true predictive errors. Because the predictions $\hat{\mathbf{v}}^*$ at new parameter settings $\boldsymbol{\theta}^*$ were not included in any bootstrap samples, they are based on average of a large set of trees, which reduces the variance. These detials combine to make the quasi-realizations $\mathbf{v}^*$ reasonably close to samples from the predictive distribution while making minimal assumptions about the form of that predictive distribution.

A second approach to address the uncertainty in a nonlinear model is to specify a hierarchical model for $\mathbf{v}$ where $\mathbf{v} \sim f(\hat{\mathbf{v}}^*(\boldsymbol{\theta}^*, \sigma_v^2)$ and f is some density, and then specify a distribution for $\boldsymbol{\theta}$. This approach is more complex since samples for the vectors must be drawn separately from the parameters, so the previously discussed quasi-realization method is favored in most instances.

The last step in the implementation is to specify the emulator model: $\mathbf{y} = \widetilde{\mathbf{UD}}\mathbf{v}(\boldsymbol{\theta}) + \boldsymbol{\epsilon}$ with the errors $\boldsymbol{\epsilon} \sim N_p(\mathbf{0}, \sigma^2\mathbf{I})$. The quantities $\boldsymbol{\theta}$ and σ^2 are given priors, where the prior for $\boldsymbol{\theta}$, $N_p(\mathbf{0}, \sigma_\theta^2\mathbf{I})_L^U$, is a normal distribution truncated to the region [U , L] in $\mathbb{R}^p$ (and can be unbounded to give a traditional normal distribution) and $\log(\sigma) \sim N(0, \sigma_\sigma^2)$. As in previous methods, fitting this model in a Bayesian manner requires sampling from the full conditional distributions of $\boldsymbol{\theta}$ and σ^2 via MCMC.

Due to the complexity of the random forest models and the fact that the priors are not conjugate, a Metropolis-Hastings update can be applied. Because of the use of Metropilis-Hastings, there may be need for some adjustments in the MCMC algorithm in terms of the proposal distributions to give desirable acceptance rates. Once enough draws have been generated from the algorithm, then these can be used to stimate the value of the parameter $\boldsymbol{\theta}$.

As the discussion of the implementation of the nonlinear case shows, first-order emulators may in fact not be faster than second order emulators. The distinct advantage that first-order emulators have is that they are simpler to formulate and easier to implement.

3.6 Discussion

Gaussian Processes are a field which has been extensively studied in statistics. The variety of implementations of Gaussian Process emulation as well as its utility in likelihood-free contexts make it a natural approach for statistical inference using ABMs. The discontinuity inherent within many ABMs in terms of the sets of agent rules and corresponding outputs make treed approaches particularly useful. In applications involving physical or biological systems, first-order emulators present a useful approach to infer parameters of practical significance and place less focus on model calibration. Overall, the ability to perform uncertainty quantification for ABMs is an important topic, and emulators offer a means to do so.

4

Approximate Bayesian Computation

4.1 Overview

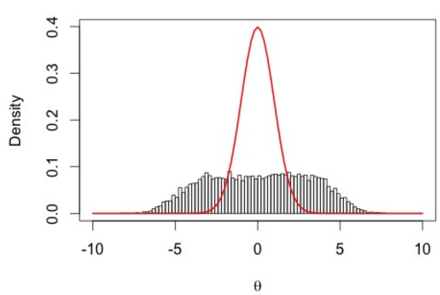
Another approach for performing Bayesian inference utilizing ABMs is Approximate Bayesian Computation (ABC) (Pritchard et al., 1999). The objective of ABC is to make inferences about a parameter (or set of parameters) θ based on a observed data, x_0 when likelihood functions are intractable. The ABC method compares simulated data to observed data and uses this as a criterion for inference. ABC has been used in a variety of applications including genetics (Tanaka et al., 2006), ecology (Beaumont, 2006) and population evolution (Drovandi and Pettitt, 2011).

The most basic ABC algorithm uses rejection sampling and proceeds as follows:

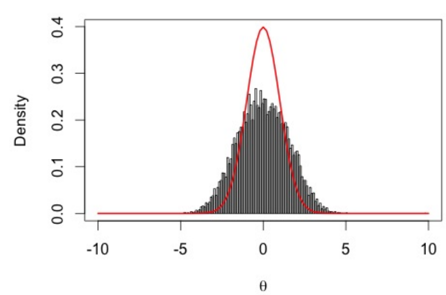
1. Sample a value θ' from the prior distribution, $\pi(\theta)$
2. Generate a set of data x' from $f(\cdot|\theta')$
3. Measure the distance between the generated data and the observed data, $\rho(x', x_0)$. If $\rho(x', x_0) \leqslant \epsilon$, accept θ', otherwise reject this value.
4. Return to 1.

Under this approach, for a given value of ϵ, this procedure represents sampling from $\pi(\theta|\rho(x', x) < \epsilon)$. In the limiting cases, $\epsilon = 0$ would represent draws from the true posterior distribution of $\pi(\theta|x)$, and $\epsilon = \infty$ would represent draws from the prior, since no sampled values would be rejected.

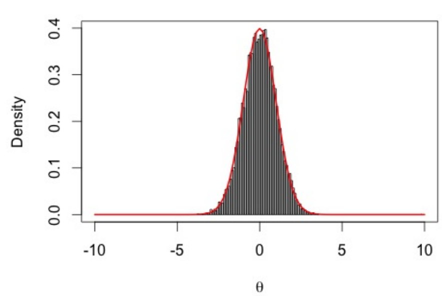
To illustrate this method, I present ABC applied to an example where the true posterior distribution is known. In this example, the distribution of the data is $x|\theta \sim \mathcal{N}(\theta, 1)$ and the prior for θ is Uniform(-10,10). Given an observation $x_0 = 0$, the target posterior is known to be $\theta|x_0 \sim \mathcal{N}(0, 1)$ truncated to (-10,10). The distance measure for this example is $\rho(x', x_0) = |x' - x_0|$. Figure 4.1 shows how the value of ϵ affects the quality of the approximation to the true posterior.



(a)



(b)



(c)

FIGURE 4.1: Histograms of the approximate posterior from ABC in the normal example with $\epsilon = 5$ (a), 2 (b), and 0.1 (c). In each plot, the red line represents the true posterior density.

This basic ABC approach is often altered by using some summary statistic, $\mathcal{S}$, instead of using the full data x. Ideally, $\mathcal{S}$ would be a sufficient statistic since this would simultaneously simplify computation and, because of sufficiency, $\pi(\theta|x) = \pi(\theta|\mathcal{S})$. However, defining such a statistic can be difficult since the likelihood function is not available and it could be unclear which aspects of the data are relevant to the model. This issue has been discussed by Fearnhead and Prangle (2012), among others. A drawback of this algorithm is that, if the prior distribution is very different from the posterior distribution, the acceptance rate will be low.

4.2 ABC MCMC

To avoid low acceptance rates of the ABC rejection sampler, an ABC method based on MCMC was proposed in Marjoram et al. (2003). The ABC MCMC algorithm proceeds as follows:

1. Initialize θ_i, $i = 0$.
2. Propose θ' according to a proposal distribution $q(\theta|\theta_i)$.
3. Simulate a data set x' from $f(x|\theta')$.
4. If $\rho(x', x_0) \leqslant \epsilon$, proceed to step 5, otherwise set $\theta_{i+1} = \theta_i$ and go to step 6.
5. Set $\theta_{i+1} = \theta'$ with probability

   $\alpha = \min(1, \frac{\pi(\theta')q(\theta_i|\theta')}{\pi(\theta_i)q(\theta'|\theta_i)})$

   and set $\theta_{i+1} = \theta_i$ with probability $1 - \alpha$
6. Set $i = i + 1$ and go to step 2.

A Markov chain with the stationary distribution $\pi(\theta|\rho(x', x_0) \leqslant \epsilon)$ is produced from the algorithm. Hence, ABC MCMC is guaranteed to converge to the target approximate posterior distribution. A symmetric proposal distribution q will result in the acceptance probability α depending only on the prior distribution π. Additionally, a uniform prior results in $\alpha = 1$.

The ABC MCMC approach has the potential disadvantge that the correlated nature of the samples combined with a potentially low probability of acceptance could result in very long chains and having the chains get stuck in low probability regions for several iterations.

4.3 ABC Sequential Monte Carlo

An extension of this approach which avoids some of the disadvantages of the rejection method and MCMC method is ABC using Sequential Monte Carlo (SMC) (Sisson et al., 2007; Toni et al., 2009). An adaptive version of the algorithm was also proposed in DelMoral et al. (2012). ABC SMC with sequential importance sampling uses parameter values (particles) $\{\theta^{(1)}, ..., \theta^{(N)}\}$ sampled from the prior, propagates them through a sequence of distributions based on a monotone decreasing sequence of tolerances, $\epsilon_1, ..., \epsilon_T$, to obtain a sample from a target distribution. This algorithm proceeds as follows:

1. Initialize tolerances $\epsilon_1, ..., \epsilon_T$

   Set population indicator $t = 0$

2. (a) Set particle indicator to $i = 1$

   (b) If $t = 0$, sample θ'' independently from the prior $\pi(\theta)$

   Else, sample θ' from the the previous population $\{\theta^{(i)}_{t-1}\}$ with weights w_{t-1} and perturb the particle to obtain $\theta'' \sim K_t(\theta|\theta')$ where K_t is the perturbation kernel.

   If $\pi(\theta'') = 0$, return to step 2(b)

Simulate a candidate data set $x' \sim f(x|\theta'')$.

If $\rho(x', x_0) \geqslant \epsilon_t$, return to step 2(b).

(c) Set $\theta_t^{(i)} = \theta''$ and calculate weight for particle $\theta_t^{(i)}$ by

$$w_t^{(t)} = \begin{cases} 1, & \text{if t=0} \\ \frac{\pi(\theta_t^{(i)})}{\sum_{j=1}^N w_{t-1}^{(j)} K_t(\theta_{t-1}^{(j)}, \theta_t^{(i)})}, & \text{if t} > 0 \end{cases}$$

If $i < N$, set $i = i + 1$, go to step 2(b).

3. Normalize the weights.

If $t < T$, set $t = t + 1$ and return to step 2(a).

The perturbation kernel is often chosen as a random walk (either uniform or Gaussian). Also note that the case where $T = 1$ corresponds to the ABC rejection algorithm.

The ABC SMC approach provides information about model sensitivity to different parameters through observing the shape of intermediate and posterior distributions. Models are more sensitive to parameters that are inferred quickly and have narrow credible intervals than those inferred later and have a more diffuse posterior. Bonassi (2013) proposes an extension of this method which uses adaptive weights.

Additionally, ABC SMC is demonstrated in a model selection framework which is consistent with standard Bayesian model selection concepts and the use of Bayes factors, comprehensively discussed in Kass and Raftery (1995). The ABC SMC model selection approach includes a discrete parameter $m \in \{1, ..., M\}$ as an indicator for each of the collection of models being considered and denotes model-specific parameters as $\theta(m) = (\theta(m)^{(1)}, ..., \theta(m)^{(k_m)}$ where k_m denotes the number of parameters in model m.

This algorithm proceeds as follows:

1. Initialize tolerances $\epsilon_1, ..., \epsilon_T$

Set population indicator $t = 0$

2. (a) Set particle indicator to $i = 1$

   (b) Sample m' from $\pi(m)$ If $t = 0$, sample θ'' independently from the prior $\pi(\theta(m'))$

   If $t > 0$, sample θ' from the previous population $\{\theta(m')_{t-1}\}$ with weights $w(m')_{t-1}$.

   Perturb the particle θ' to obtain $\theta'' \sim K_t(\theta|\theta')$.

   If $\pi(\theta'') = 0$, return to step 2(b)

   Simulate a candidate data set $x' \sim f(x|\theta'', m')$.

   If $\rho(x', x_0) \geqslant \epsilon_t$, return to step 2(b).

   (c) Set $m_t^{(i)} = m'$ and add θ'' to the population of particles $\{\theta(m')_t\}$ calculating its weight as

   $$w_t^{(t)} = \begin{cases} 1, & \text{if t=0} \\ \frac{\pi(\theta'')}{\sum_{j=1}^{N} w_{t-1}^{(j)} K_t(\theta_{t-1}^{(j)}, \theta'')}, & \text{if t} > 0 \end{cases}$$

   If $i < N$, set $i = i + 1$, go to step 2(b).

3. For every m, normalize the weights.

   If $t < T$, set $t = t + 1$ and return to step 2(a).

Outputs from this algorithm are approximations of the marginal posterior distribution of the model parameter $P(m|x)$ and the marginal posterior distributions of parameters $P(\theta_i|x, m)$, $m = 1, ..., M$, $i = 1, ..., k_m$. Note that it is possible for a model to die out (i.e. have no particles left belonging to that particular model) if it offers a poor description of the data, which results in sampling of particles only from the remaining models.

In addition to avoiding the curse of dimensionality, which is more prominent in the basic ABC rejection scheme, the SMC approach is more robust for a broader class of models. The ABC SMC approach is particularly useful for ABMs because of the model selection aspect. The model selection approach allows ABMs not only to be compared with other ABMs, but also with different types of models. In allowing for a model-specific parameter set in the algorithm, the Bayesian appoach of model selection involved in this approach allows the comparison of distinct models and does not require them to be nested. Because the algorithm allows the comparison of multiple models at once, there is less drawback to considering a broad collection of models. Additionally, the model selection algorithm implicitly penalizes models for having large numbers of parameters by decreasing the probability of accepting a perturbed particle. This feature of the algorithm encourages parsimonious models, which is an aspect of ABMs which has not received a great deal of attention in practice.

4.4 ABC Regression Adjustment

Another modification of ABC involves regression adjustment to allow higher acceptance rates. These approaches allow for a larger threshold ϵ and then correct the accepted draws of the parameter to account for the discrepancy between the simulated and observed data. Regression-adjusted ABC methods are appealing since a higher acceptance rate and larger ϵ mean that fewer data generation iterations (i.e. model simulations) are required, which is particularly advantageous for larger and more complex ABMs.

Early versions of the ABC algorithm with a regression adjustment in Beaumont et al. (2002) assume the conditional density of interest can be described by a regression model of the following form:

$$\theta_i = \alpha + (\mathbf{x}_i - \mathbf{x}_0)^T \beta + \varepsilon_i$$

for $i = 1, ..., m$, some intercept α and vector of regression coefficients β, and the ε_i

are uncorrelated with zero mean and common variance. When $\mathbf{x}_i = \mathbf{x}_0$, the θ_i are drawn from the posterior with $\mathbb{E}[\theta|\mathbf{x} = \mathbf{x}_i] = \alpha$.

Based on the assumed regression equation, the least squares estimate of $(\hat{\alpha}, \hat{\beta})$ is $(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{\theta}$, where $\boldsymbol{\theta}$ is the m-vector of parameters and $\mathbf{X}$ is an $m \times (p+1)$ matrix where row i is of the form $(1,\ x_{i1} - x_{01},\ \cdots,\ x_{ip} - x_{0p})$.

It follows that $\theta_i^* = \theta_i - (\mathbf{x}_i - \mathbf{x}_0)^T\hat{\beta}$ will form an approximate random sample from $P(\theta|\mathbf{x}_i = \mathbf{x}_0)$ and $\hat{\alpha}$ can be interpreted as a point estimate of θ.

4.4.1 Local-linear regression

While complete linearity may be an implausible assumption, the approach may apply locally in some neighborhood of $\mathbf{x}_0$.

For a local-linear regression approach, the local parameter estimates must now minimize

$$\sum_{i=1}^{m}\{\theta_i - \alpha - (\mathbf{x}_i - \mathbf{x}_0)^T\beta\}^2 K_\delta\rho(\mathbf{x}_i - \mathbf{x}_0)$$

where K_δ is a kernel function. The local parameter estimates are now

$$(\hat{\alpha}, \hat{\beta}) = (\mathbf{X}^T W\mathbf{X})^{-1}\mathbf{X}^T W\boldsymbol{\theta},$$

where W is a diagonal matrix with ith diagomal element $K_\delta\rho(\mathbf{x}_i - \mathbf{x}_0)$.

The posterior mean estimate for θ is $\hat{\alpha} = \frac{\sum_i^m \theta_i^* K_\delta\rho(\mathbf{x}_i - \mathbf{x}_0)}{K_\delta\rho(\mathbf{x}_i - \mathbf{x}_0)}$. Note that using the indicator kernel would result in local-constant regression which would correspond to a rejection method estimate.

4.4.2 Non-linear regression

A modification to the above regression approach proposed by Blum and Francois (2010), which introduces a nonlinear regression model for parameter estimation. The proposed regression model has the form

$$\theta_i = \eta(\mathbf{x}_i) + \sigma(\mathbf{x}_i) \times \epsilon_i,\ i = 1, ..., m$$

where the ϵ_i are independent with zero mean and common variance, $\eta(\mathbf{x}_i) = \mathbb{E}[\theta|\mathbf{x} = \mathbf{x}_i]$ and $\sigma(\mathbf{x}_i)$ denotes the variance $\mathrm{Var}[\theta|\mathbf{x} = \mathbf{x}_i]$.

An estimate for the expectation $\hat{\eta}(\mathbf{x}_i)$ can be obtained from fitting a non-linear regression model, while the variance is estimated from another regression model for the log residuals

$$\log(\theta_i - \hat{\eta}(\mathbf{x}_i))^2 = \log(\sigma^2(\mathbf{x}_i)) + \epsilon_i.$$

The non-linear extension adds flexiblity to the regression adjustment approach by removing the assumption of linearity even locally. The complex dynamics of ABMs can be such that local behavior is highly non-linear and this approach expands the class of models to which ABC methods can be applied.

4.5 Reinforcement Learning

A particularly relevant application of ABC for ABMs is ABC Reinforcement Learning (Dimitrakakis and Tziortziotis, 2013). The reinforcement learning problem is pertinent to ABMs because it deals with an agent, whose actions $a_t \in \mathcal{A}$ are determined by some policy π, acting in an unknown environment μ. The environment reacts to agents' actions with a sequence of observations $x_t \in \mathcal{X}$ and scalar-valued rewards r_t. The agent-environment interaction could depend on the complete history $h \in \mathcal{H}$, where $\mathcal{H} = (\mathcal{X}, \mathcal{A}, \mathbb{R})$ is the set of all action-state-reward sequences.

The agent's objective is to maximize its utility, $U = \sum_{t=1}^{\infty} \gamma^{t-1} r_t$, which is a discounted sum of the rewards obtained, with $\gamma \in [0, 1]$. The optimal agent policy will maximize the expectation of U. Reinforcement learning is an extension of ABMs in this sense, since there is the notion of long-term utility rather than a one-step-ahead consideration. While reinforcement learning looks at a single agent's actions, it could be incorporated into a more traditional ABM with multiple agents, with limited rewards and utility divided among all agents based on their behaviors and interactions.

Using ABC for reinforcement learning, the procedure generates sample models $\mu^{(k)}$ from a prior distribution ξ and then creates a history $h^{(k)}$ from each sampled model. If $h^{(k)}$ is sufficiently close to the true history h (or if $f(h^{(k)})$ is sufficiently close to $f(h)$ for a sufficient statistic $f : \mathcal{H} \to \mathcal{W}$ where $\mathcal{W}$ is a vector space) then $\mu^{(k)}$ is accepted as a sample from the posterior. Due to the importance of utility in reinforcement learning problems, Dimitrakakis and Tziortziotis (2013) suggested a utility-based statistic for determining whether generated histories are close enough to observed histories.

Given a history h containing N trajectories in the environment, with the ith trajectory resulting in a utility of $U^{(i)}$, the mean estimate is $\hat{\mathbb{E}}^{N}_{\pi,\mu}[U] = \frac{1}{N}U^{(i)}$. After generating history $h^{(k)}$ with N^* trajectories from the sampled $\mu^{(k)}$, the mean estimate $\hat{\mathbb{E}}^{N*}_{\pi,\mu^{(k)}}[U]$ can be obtained. By the Hoeffding inequality, the difference between the true and sampled mean utilities $|\mathbb{E}^{N}_{\pi,\mu}[U] - \mathbb{E}^{N*}_{\pi,\mu^{(k)}}[U]|$ is bounded below by

$$|\hat{\mathbb{E}}^{N}_{\pi,\mu}[U] - \hat{\mathbb{E}}^{N*}_{\pi,\mu^{(k)}}[U]| - U_{\max}\sqrt{\frac{\log(2/\delta)(N+N^*)}{2NN^*}}$$

with probability at least $1-\delta$. Here, $U_{\max}$ is the range of the utility function. This lower bound is then used as $\|f(h) - f(h^{(k)})\|$. The only parameters of this statistic are the probability δ and the number of trajectories in the sample, N^*. The final component in the reinforcement problem is, for a sampled model $\hat{\mu}$, to select the optimal policy by means of dynamic programming.

4.6 Discussion

ABC methods allow flexible inference using ABMs, especially in the choice of what statistic to use when comparing simulated and observed data. While sufficiency gives a better approximation to the true posterior distribution of interest, other choices of statistic allow for prioritization of certain aspects of the model. The continuing development of ABC algorithms demonstrates the utility of the approach for inference in a wide range of problems.

Because of the complexity of most ABMs, summary statistics will rarely be sufficient and the selection of summary statistics can be difficult. Some approaches involving the consideration of dimension reduction and retention of information have been examined to address these issues (Aeschbacher et al., 2012).

A potential limitation in implementing ABC in this context is that the run times of many ABMs of interest are long enough that this approach can be time consuming. Unlike the emulator approach, which requires only a limited number of simulations at various settings, ABC can require thousands of ABM simulations which is not always feasible. Although some measures can be taken to expedite the process (such as model simplification to decrease run time or running ABM simulations in parallel), the time necessary to generate a sufficient number of samples to approximate the posteriors of interest can be limiting.

5

ABM Application: MSM Community

5.1 Overview

A primary use for ABMs is in simulating the development of human social networks. The flexibility of the inputs allows for a wide range of human behaviors and interaction to be modeled in a straightforward way.

Here, we examine a network of men who have sex with men (MSM) in southern India. This community has been studied closely due to its dangerously high HIV prevalence rate. Obtaining a better understanding of this community can allow us to introduce interventions to mitigate this epidemic. The objective of our analysis is to determine the latent structure driving sex ties within the community, thus informing the spread of HIV. We then use ABMs to investigate this latent structure and examine the effects of medical interventions on the spread of HIV.

5.2 MSM Network Data

In this study, MSM network data were acquired using Cell-phone Assisted Network Detection and Identification (CANDID) and time-location cluster sampling within well-characterized sex-venues in Southern India (Schneider et al., 2011a). Cell phone contact lists were merged utilizing phone numbers as unique identifiers. Recruitment of MSM study respondents continued until the chance of a new respondent already being part of the network exceeded 95%. This process resulted in a network census of MSM (n=245) and an augmented MSM digital communication network (n=4843 nodes, 6624 edges, 5.2) of which both social (n=2658 nodes, 3957 edges) and sex (n=2605 nodes, 2667 edges) sub-networks were also analyzed. The network-based model predictions of sex behavior incorporate the reports of other network members on the individual of interest as well as structural features of network members such as centrality.

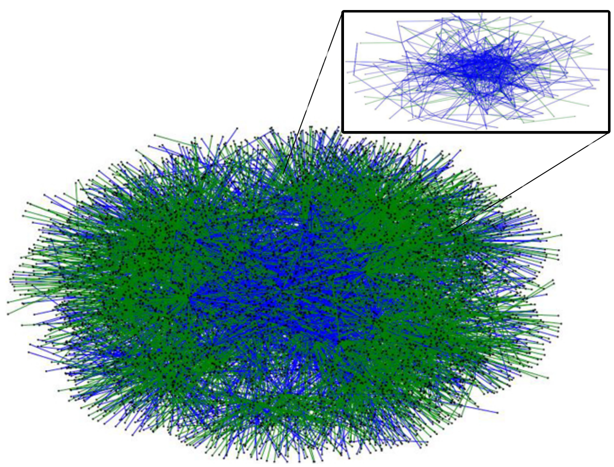


FIGURE 5.1: MSM Network Visualization. Inset represents edges restricted to the 245 egos.

The data were obtained by interviewing 245 individuals (egos) and obtaining information about their acquaintences who were also MSM (alters). The information provided by the egos about the alters includes the nature of the relationship (sex partner or social edge), age, marital status, religion, caste, whether the individual is a sex worker, whether they met at a 'hotspot' (an area known to have high prostitution activity), and sex position.

For the 245 egos who were interviewed, we also have self-report data identifying age, marital status, religion, caste, whether the individual is a sex worker and preferred sex position, as well as results of HIV tests and tests for syphilis and herpes.

5.3 Network Latent Structure

In order to explore latent structure within the network, we utilized a Mixed Membership Block Model (MMSB) (Airoldi et al., 2008). This procedure seeks to determine a number K of latent blocks within the network and associate each node within the blocks by a membership probability vector. The model posits that the network graph $\mathcal{G} = (\mathcal{N}, \mathcal{R})$ is generated in the following way:

- $\pi_p \sim Dir(\alpha)$ a K-dimensional mixed membership vector for all nodes p
- $z_{p \to q} \sim Mulit(\pi_p)$ a K-dimensional membership indicator for all pairs p and q
- $z_{p \leftarrow q} \sim Mulit(\pi_q)$ a K-dimensional membership indicator for all pairs p and q
- $G(p, q) \sim Bern(z_{p \to q} B z_{p \leftarrow p}^T)$ for B a K-by-K matrix of Bernoulli rates determining ties within/across blocks.

Additionally, we can introduce a sparsity parameter, $\rho \in [0, 1]$, which down-weights the probability of an interaction to $(1 - \rho)(z_{p \to q}^T B z_{q \to p})$. This parameter accounts for non-interactions not explained by the block model itself (perhaps due to infrequency of interaction between two nodes). Large value of ρ weighs interactions more than non-interactions in determining estimates of $\{\pi_{1:N}, Z_{\to}, Z_{\leftarrow}, \alpha, B\}$, the mixed membership probability vector, matrix of membership indicators for senders and receivers, Dirichlet hyperparameter and matrix of Bernoulli rates, respectively.

We seek to obtain the posterior distributions of π_p, $z_{p \to q}$ and $z_{q \to p}$ for all p, q and to estimate Dirichlet hyperparameter α, the $K \times K$ matrix of Bernoulli rates B and the sparsity parameter ρ.

The joint probability of the data R and the latent variables $\{\vec{\pi}_{1:N}, Z_{\rightarrow}, Z_{\leftarrow}\}$ is

$$p(R, \vec{\pi}_{1:N}, Z_{\rightarrow}, Z_{\leftarrow}|\vec{\alpha}, B) = \prod_{p,q} P(R(p,q)|\vec{z}_{p\rightarrow q}, \vec{z}_{q\rightarrow p}, B) P(\vec{z}_{p\rightarrow q}|\vec{\pi}_p) P(\vec{z}_{q\rightarrow p}|\vec{\pi}_q) \prod_p P(\vec{\pi}_p|\vec{\alpha})$$

The normalizing constant for the posterior distribution of the latent variables is the integral

$$p(R|\vec{\alpha}, B) = \int_{\vec{\pi}_{1:N}} \prod_{p,q} \sum_{\vec{z}_{p\rightarrow q}, \vec{z}_{q\rightarrow p}} p(R, \vec{\pi}_{1:N}, Z_{\rightarrow}, Z_{\leftarrow}|\vec{\alpha}, B),$$

which is intractable, since it requires integrating over all $\vec{\pi}_p$. We can use mean-field variational methods (Teh et al., 2008) to approximate the posterior.

Using a set of variational free parameters $\Delta = \{\vec{\gamma}_{1:N}, \Phi_{\rightarrow}, \Phi_{\leftarrow}\}$, we can introduce a distribution q of the latent variables, fully factorized:

$$q(\vec{\pi}_{1:N}, Z_{\rightarrow}, Z_{\leftarrow}|\vec{\gamma}_{1:N}, \Phi_{\rightarrow}, \Phi_{\leftarrow}) = \prod_p q_1(\vec{\pi}_p|\vec{\gamma}_p) \prod_{p,q} \Big(q_2(\vec{z}_{p\rightarrow q}|\vec{\phi}_{p\rightarrow q}) q_2(\vec{z}_{q\rightarrow p}|\vec{\phi}_{q\rightarrow p}) \Big)$$

where q_1 is a Dirichlet and q_2 is a multinomial.

By Jensen's Inequality, we can establish a lower bound for the log of the intractable integral $p(R|\vec{\alpha}, B)$:

$$\log(p(R|\vec{\alpha}, B)) \geqslant \mathbb{E}_q[\log(p(R, \vec{\pi}_{1:N}, Z_{\rightarrow}, Z_{\leftarrow}|\vec{\alpha}, B))] - \mathbb{E}_q[\log(q(\vec{\pi}_{1:N}, Z_{\rightarrow}, Z_{\leftarrow}))].$$

adjusting the variational free parameters can tighten the lower bound on $\log(p(R|\vec{\alpha}, B))$ (Jordan and Wainwright, 2003), which minimizes the Kullback-Leibeler divergence between q and the true posterior of interest.

We can express the KL-divergence between q and the true posterior $p(\vec{\pi}_{1:N}, Z_{\rightarrow}, Z_{\leftarrow}|R)$ as:

$$
\begin{aligned}
D_{KL}(q\|p) &= \int_{\vec{\pi}_{1:N}} \prod_{p,q} \sum_{\vec{z}_{p\to q},\vec{z}_{q\to p}} q(\vec{\pi}_{1:N}, Z_{\to}, Z_{\leftarrow}) \log\left(\frac{q(\vec{\pi}_{1:N}, Z_{\to}, Z_{\leftarrow})}{p(\vec{\pi}_{1:N}, Z_{\to}, Z_{\leftarrow}|R,\alpha,B)}\right) \\
&= \mathbb{E}_q[\log\left(\frac{q(\vec{\pi}_{1:N}, Z_{\to}, Z_{\leftarrow})}{p(\vec{\pi}_{1:N}, Z_{\to}, Z_{\leftarrow}|R,\alpha,B)}\right)] \\
&= \mathbb{E}_q[\log\left(\frac{q(\vec{\pi}_{1:N}, Z_{\to}, Z_{\leftarrow})}{p(\vec{\pi}_{1:N}, Z_{\to}, Z_{\leftarrow},R|\alpha,B)}\right) + \log(p(R|\alpha,B))] \\
&= \mathbb{E}_q[\log(q(\vec{\pi}_{1:N}, Z_{\to}, Z_{\leftarrow})) - \log(p(\vec{\pi}_{1:N}, Z_{\to}, Z_{\leftarrow},R|\alpha,B))] + \log(p(R|\alpha,B))
\end{aligned}
$$

so

$$
\begin{aligned}
\log(p(R|\alpha,B)) &= D_{KL}(q\|p) - \mathbb{E}_q[\log(q(\vec{\pi}_{1:N}, Z_{\to}, Z_{\leftarrow})) - \log(p(\vec{\pi}_{1:N}, Z_{\to}, Z_{\leftarrow},R|\alpha,B))] \\
&= D_{KL}(q\|p) + \mathbb{E}_q[\log(p(\vec{\pi}_{1:N}, Z_{\to}, Z_{\leftarrow},R|\alpha,B))] - \mathbb{E}_q[\log(q(\vec{\pi}_{1:N}, Z_{\to}, Z_{\leftarrow}))] \\
&\geqslant \mathbb{E}_q[\log(p(\vec{\pi}_{1:N}, Z_{\to}, Z_{\leftarrow},R|\alpha,B))] - \mathbb{E}_q[\log(q(\vec{\pi}_{1:N}, Z_{\to}, Z_{\leftarrow}))]
\end{aligned}
$$

We use a variational EM algorithm to arrive at optimal values of the variational free parameters and hyperparameters.

- Variational E step: Update $\{\vec{\gamma}_{1:N}, \Phi_{\to}, \Phi_{\leftarrow}\}$ for fixed values of $\vec{\alpha}$ and B

$$
\begin{aligned}
\hat{\phi}_{p\to q,g} &\propto e^{\mathbb{E}_q[\log(\pi_{p,g})]} \prod_h \left(B(g,h)^{R(p,q)}(1-B(g,h))^{1-R(p,q)}\right)^{\phi_{q\to p,h}} \\
\hat{\phi}_{q\to p,h} &\propto e^{\mathbb{E}_q[\log(\pi_{q,h})]} \prod_g \left(B(g,h)^{R(p,q)}(1-B(g,h))^{1-R(p,q)}\right)^{\phi_{p\to q,g}} \\
\hat{\gamma}_{p,k} &= \alpha_k + \sum_q \phi_{p\to q,k} + \sum_q \phi_{q\to p,k}
\end{aligned}
$$

- Variational M step: Obtain updated empirical Bayes estimates of $\vec{\alpha}$ and B based on updated variational free parameter values

$$
\hat{B}(g,h) = \frac{\sum_{p,q} R(p,q)\phi_{p\to q,g}\phi_{q\to p,h}}{\sum_{p,q}\phi_{p\to q,g}\phi_{q\to p,h}}; \qquad (g,h) \in [1,K]\times[1,K]
$$

Obtain the estimate for $\vec{\alpha}$, using a Newton-Raphson method (Blei and Jordan, 2003):

$$\nabla\mathcal{L}_\alpha = \frac{\partial\mathcal{L}_{\vec{\alpha}}}{\partial\alpha_k} = N * \left(\psi(\sum_k \alpha_k) - \psi(\alpha_k)\right) + \sum_p \left(\psi(\gamma_{p,k}) - \psi(\sum_k \gamma_{p,k})\right)$$

$$H(\mathcal{L}_\alpha) = \frac{\partial\mathcal{L}_{\vec{\alpha}}}{\partial\alpha_{k_1}\alpha_{k_2}} = N * \left(\mathbb{I}_{(k_1=k_2)}\psi'(\alpha_{k_1}) - \psi'(\sum_k \alpha_k)\right)$$

Obtain the estimate for ρ,

$$\hat{\rho} = \frac{\sum_{p,q}(1 - R(p,q)) * (\sum_{g,h}\phi_{p\to q,g}\phi_{q\to p,h})}{\sum_{p,q}\sum_{g,h}\phi_{p\to q,g}\phi_{q\to p,h}}$$

The mean-field fully factorized q is

$$q(\vec{\pi}_{1:N}, Z_\to, Z_\leftarrow|\vec{\gamma}_{1:N}, \Phi_\to, \Phi_\leftarrow) = \prod_p q_1(\vec{\pi}_p|\vec{\gamma}_p) \prod_{p,q}\left(q_2(\vec{z}_{p\to q}|\vec{\phi}_{p\to q})q_2(\vec{z}_{q\to p}|\vec{\phi}_{q\to p})\right),$$

which assumes $\pi_{1:N}$, $Z_\to$ and $Z_\leftarrow$ are independent, which they are not (recall $\vec{z}_{p\to q} \sim$ Multinomial (π_p)). To maintain the dependence, a nested variational algorithm is used. In the algorithm, $\vec{\phi}_{p\to q}, \vec{\phi}_{p\leftarrow q}$ are updated until convergence, to keep this block optimized given all of the other parameters and to maintain dependence between $\vec{\phi}$ and $\vec{\gamma}$, which induces dependence between $\vec{z}_{p\to q}$ and $\vec{\pi}_p$. Figure 5.2 describes the variational inference algorithm.

From examining the number of groups, we see that K=2 yields the highest log-likelihood, which is consistent with the results of the previous stochastic block model results.

The estimate for the sparsity parameter is $\rho = 0.93$, the estimate of $\vec{\alpha}$ is (0.2813, 0.5182), and the estimate for the Bernoulli rate matrix B is shown in Table 5.1.

1. initialize $\vec{\gamma}^{0}_{pk} = \frac{2N}{K}$ for all p, k
2. **repeat**
3.   **for** $p = 1$ to N
4.   **for** $q = 1$ to N
5.   get **variational** $\vec{\phi}^{t+1}_{p\rightarrow q}$ and $\vec{\phi}^{t+1}_{p\leftarrow q} = f\left(R(p,q), \vec{\gamma}^{t}_{p}, \vec{\gamma}^{t}_{q}, B^{t} \right)$
6.   partially update $\vec{\gamma}^{t+1}_{p}, \vec{\gamma}^{t+1}_{q}$ and B^{t+1}
7. **until** convergence

5.1. initialize $\phi^{0}_{p\rightarrow q,g} = \phi^{0}_{p\leftarrow q,h} = \frac{1}{K}$ for all g, h
5.2. **repeat**
5.3.   **for** $g = 1$ to K
5.4.   update $\phi^{s+1}_{p\rightarrow q} \propto f_1\left(\vec{\phi}^{s}_{p\leftarrow q}, \vec{\gamma}_{p}, B \right)$
5.5.   normalize $\vec{\phi}^{s+1}_{p\rightarrow q}$ to sum to 1
5.6.   **for** $h = 1$ to K
5.7.   update $\phi^{s+1}_{p\leftarrow q} \propto f_2\left(\vec{\phi}^{s}_{p\rightarrow q}, \vec{\gamma}_{q}, B \right)$
5.8.   normalize $\vec{\phi}^{s+1}_{p\leftarrow q}$ to sum to 1
5.9. **until** convergence

FIGURE 5.2: Variational inference algorithm for variational free parameters, with lower panel corresponding to the nested algorithm for inference for $(\phi_{q\rightarrow p},\, \phi_{q\leftarrow p})$

Table 5.1: Estimated rates of interaction within and across blocks from the MMSB.

	$\mathcal{B}_1$	$\mathcal{B}_2$
$\mathcal{B}_1$	0.3516	0.4135
$\mathcal{B}_2$	0.4135	0.5613

To determine which attribute(s) are contributing to the latent structure, we examined individuals with high posterior membership probabilities. The posterior membership probability vectors for the 245 egos are shown in Figure 5.3.

Block membership assignments matched most closely to self-reported marital status (80.2% correct identification, marital status data agreed with self-report 78.1%). Blocking assignments compared to self-reported marital status are shown in Table 5.2.

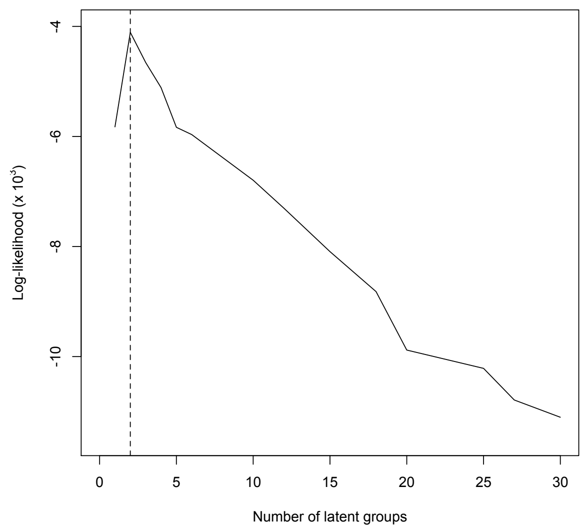


FIGURE 5.3: Approximate log-likelihood of MMSB by number of latent blocks

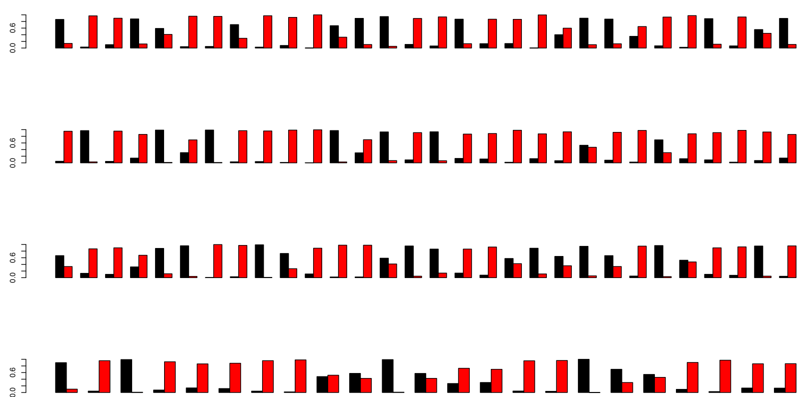


FIGURE 5.4: Posterior membership vectors $\vec{\pi}_p$ for egos

Table 5.2: MMSB assignments compared to marital status

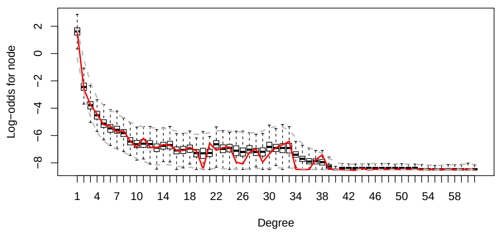
Block Assignment	Self-reported Married	Self-reported Unmarried
1	48	25
2	14	119

Block 1 matches most closely with individuals who are self-reported as married and block 2 matches with unmarried individuals. The model suggests that ties are formed most often between unmarried individuals, with ties between married and unmarried men occurring slightly less often, and ties between married men occurring least often.

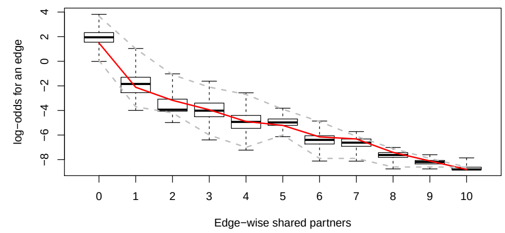
5.4 ABM for MSM Network

After conducting the MMSB analysis and investigating the latent structure within the network, we developed an ABM to validate the results and simulate the evolution of the network under different conditions. This model was developed in NetLogo (Wilensky, 1999) and analyzed in R using the *RNetLogo* package (Thiele et al., 2012).

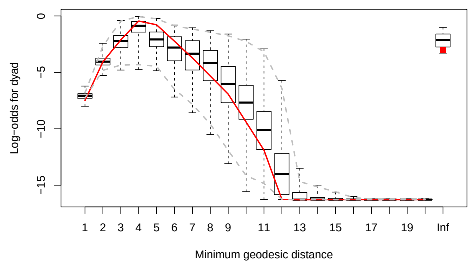
A network of 4843 agents was generated, with attributes matching those found in the network. For individuals with differing reported attributes and lacking self-report data, a mixed effects model (Westveld and Hoff, 2012) was used to determine an individual attribute assignment. We included a marriage effect in the agent rule sets, using the interaction probabilities based on marital status found in the MMSB to determine tie formation. Deaths are modeled implicitly, with HIV positive individuals ceasing to form new edges within the network based on a mortality rate estimated from WHO Global Health Observatory Data (2011), but not being removed from the network count. Once all of the final agent attributes were assigned, the ABM was run 500 times, each simulation representing two years of activity, which was roughly the time period representing the maximum relationship length in the network.



(a)



(b)



(c)

FIGURE 5.5: Comparisons of goodness-of-fit measures of the actual network (red line) to 500 simulations of the ABM. Figure 5.5(a) shows the degree distribution of the network, Figure 5.5(b) shows the distribution of edgewise-shared partnerships and Figure 5.5(c) shows the distribution of minimum geodesic distances. The vertical axis for all of the plots are on the log-odds scale.

To determine how well the ABM fit the network itself, we compared degree distributions, edge-wise shared partner distribution and minimum geodesic distance distribution from the ABM simulations to the network data (Hunter et al., 2008). These specified statistics for assessing of goodness-of-fit prioritize number of partners in the network (degree), the number of mutual partners a pair of partners has (edge-wise shared partners), and centrality (minimum geodesic distance) as important structural aspects of the network. Wasserman and Faust (1994) established centrality as a particularly significant concept in social network analysis in terms of diffusion across networks. Additionally, in the context of HIV in the community, all of these statistics are relevant to the speed of the spread of the virus across the network. The goodness-of-fit measures are presented in Figure 5.5.

5.5 Gaussian Process Emulator

After the validation of the model to the network data, we develop a Gaussian Process emulator of the ABM to efficiently simulate the network under various scenarios. Although this particular network is not particularly computationally intensive, it lends itself to this procedure because of the available data and the variety of settings at which we can investigate the network's development. Our development of the emulator follows the approach of Higdon et al. (2008) discussed in section 2.1.

The inputs of interest in the model are the probability of condom use for a pair of individuals, the proportion of individuals who are versatile MSM type, the proportion of individuals who are sex workers and the number of individuals in the network. The primary goal of this emulator is to predict (with uncertainty) the trajectory of the HIV epidemic in this community. Estimates of the probability of HIV transmission by MSM type were based on Jin et al. (2010).

The ABM simulations represented three years of network activity, between 2008 and 2010. This period was chosen because, in addition to the estimated 2008 HIV prevalence from the network data, we have annual estimates through 2010 of changing rates of HIV prevalence within the MSM community in the same region as the original MSM data based on Armbruster et al. (2013). A total of 50 simulations were generated at input settings obtained from a Latin-hypercube space filling design (Tang, 1993; Ye et al., 2000; Leary et al., 2003). We used the inputs of the probability of condom use, the proportion of versatile individuals in the community and the proportion of individuals who are sex workers as parameters $\mathbf{t} = (t_1, t_2, t_3)$ while the size of the network served as the user-specified input x_1. Probability of condom use is an important value to infer in the network, as condoms are the most effective means for preventing the transmission of HIV and there are issues of reporting bias in individual condom use patterns (Thomas et al., 2011). The proportion of versatile individuals in the community is also an important quantity for inference because, although versatiles a loosely defined group, they enable the formation of triangles (3-cycles) within the network, increasing the likelihood and rate of HIV spread (Pitts et al., 2011). The proportion of individuals who are sex workers is useful to infer about because these individuals are of higher degree in the network and the number of sex workers in a community is significant in explaining HIV prevalence (Talbott, 2007; Lorway et al., 2009). All three of these quantities are not known with certainty for this MSM community, due in large part to reporting bias (Schneider et al., 2011b). The size of the network is the main quantity the can be controlled when gathering data in the field for a study of this type. Varying the number of individuals affects some estimates and confidence intervals, as well as raising questions about the applicability of extrapolation. In this particular application, estimates for the main quantities of interest restricted to the 245 interviewed individuals were consistent with those same quantities for the entire network, so our concerns in changing the network size in simulations are mitigated. Considerations of this nature should be made on a case-by-case basis.

Measurements were recorded at 52 equally spaced time points throughout each simulation. The simulated output, along with the summary from the network data, are presented in Figure 5.5.

5.5.1 Model Formulation

We represent the output of the ABM, $\eta(\mathbf{x}, \mathbf{t})$, using a p_η-dimensional principal component basis representation shown in Equation (3.8). For this model, $p_\eta = 2$ principal components were sufficient, as they explain 99.6% of the variation in the simulations. The basis functions are shown in Figure 5.7(a).

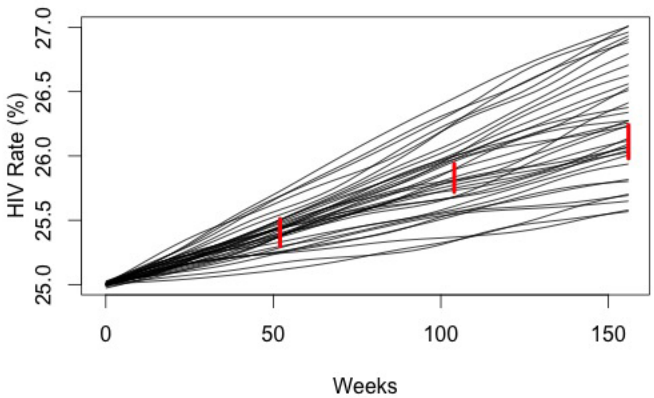
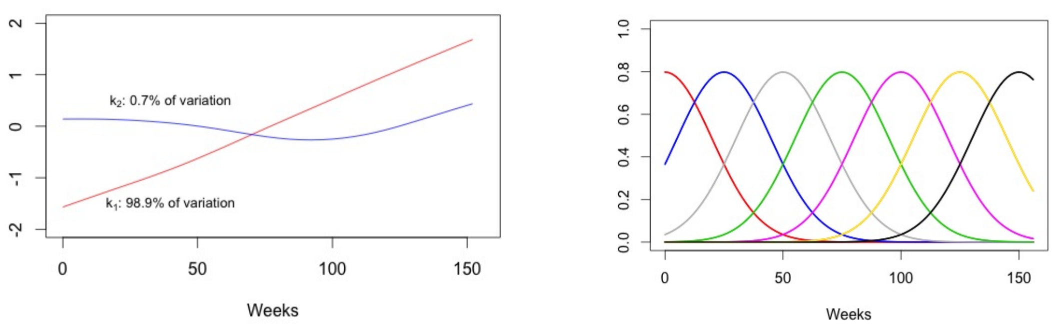


FIGURE 5.6: Output from 50 simulations of the MSM network ABM. Red bars represent 95% confidence intervals for community HIV rate based on Armbruster et al. (2013).



(a) (b)

FIGURE 5.7: Principal component basis for MSM network ABM (a) and kernel-based discrepancy basis (b).

We incorporate a bias function $\delta(\mathbf{x})$ to account for discrepancy between the ABM and the actual function of the network. We use a p_δ-dimensional basis representation for $\delta(\mathbf{x})$ as shown in Equation (3.10). The basis functions here determine the discrepancy throughout the simulation. Here, we used a Gaussian kernel for the basis functions, as we have limited knowlege of the form of the discrepancy. For this model, $p_\delta = 7$ basis functions are used to determine $\delta(\mathbf{x})$, allowing for smooth a discrepancy over time, with the spacing between kernels chosen based on Higdon (1998). The discrepancy basis functions are shown in Figure 5.7(b).

5.5.2 Posterior Sampling and Prediction

We have one observed trajectory of HIV prevalence in the community, $\mathbf{y}(x_1)$, shown by the red dashed line in Figure 5.5. Using the basis representation in Equation (3.11) to represent the observed trajectory, we specify the joint sampling model of $\mathbf{y}$ and η according to according to Equations (3.12) and (3.13). Draws from the posterior distributions of the model parameters were made via Metropolis-Hastings based on the posterior specified in Equation (3.16). The estimated posterior distributions for θ are shown in Figure 5.5.2.

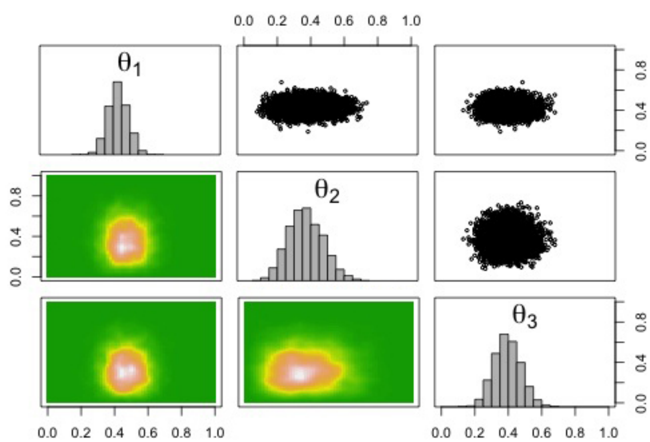


FIGURE 5.8: Two-dimensional marginals for the posterior distribution of the θ parameters.

The posterior for θ_1, representing the probability of condom use, is the most constrained, which is not surprising as it has the most interpretable influence on HIV spread in practice and in the ABM. The posterior mean of 0.429 is consistent with estimated condom use rates among MSM in a study by Dandona et al. (2005).

The posterior for θ_2, representing the proportion of individuals who are versatile MSM type shows the most spread, which is to be expected since, as mentioned previously, versatile is not a well-defined category. There have been proposals for a formal definition of versatile behavior from a quantitative standpoint (Heard and Schneider, 2013, for example), and a more generally accepted criterion would further inform inference in problems of this type.

Given posterior samples, we can then construct posterior realizations for the ABM output, the discrepancy, and make predictions of the trajectory of the HIV rate within a community of specified size x_1^* by generating predictions of the ABM output and the discrepancy function. Posterior mean estimates for η, δ and their sum, ξ, representing the behavior of the MSM network, are shown in Figure 5.5.2. The discrepancy function δ shows a smooth slightly downward curve, resulting in a plateau of the HIV rate in the community in ξ, which is consistent with decreasing rate of HIV growth presented by Armbruster et al. (2013).

Higdon et al. (2008) noted that, in the case where the discrepancy is large, it can affect the posterior distribution of θ. While that is not the case in our example, some of the issues of reporting bias and cultural identity among the MSM community likely limited the precision of the inference for the parameters in our analysis. Consideration should also be given to extrapolation of results. The issue of varying network size was mentioned in section 5.5, but, if considering extending these results, one should also consider the composition of the network being examined. Economic and cultural issues play a large role in behaviors of individuals within MSM communities and should be carefully taken into account in looking to make extrapolative predictions. Bayarri et al. (2007b) presents further discussion of details related to extrapolation.

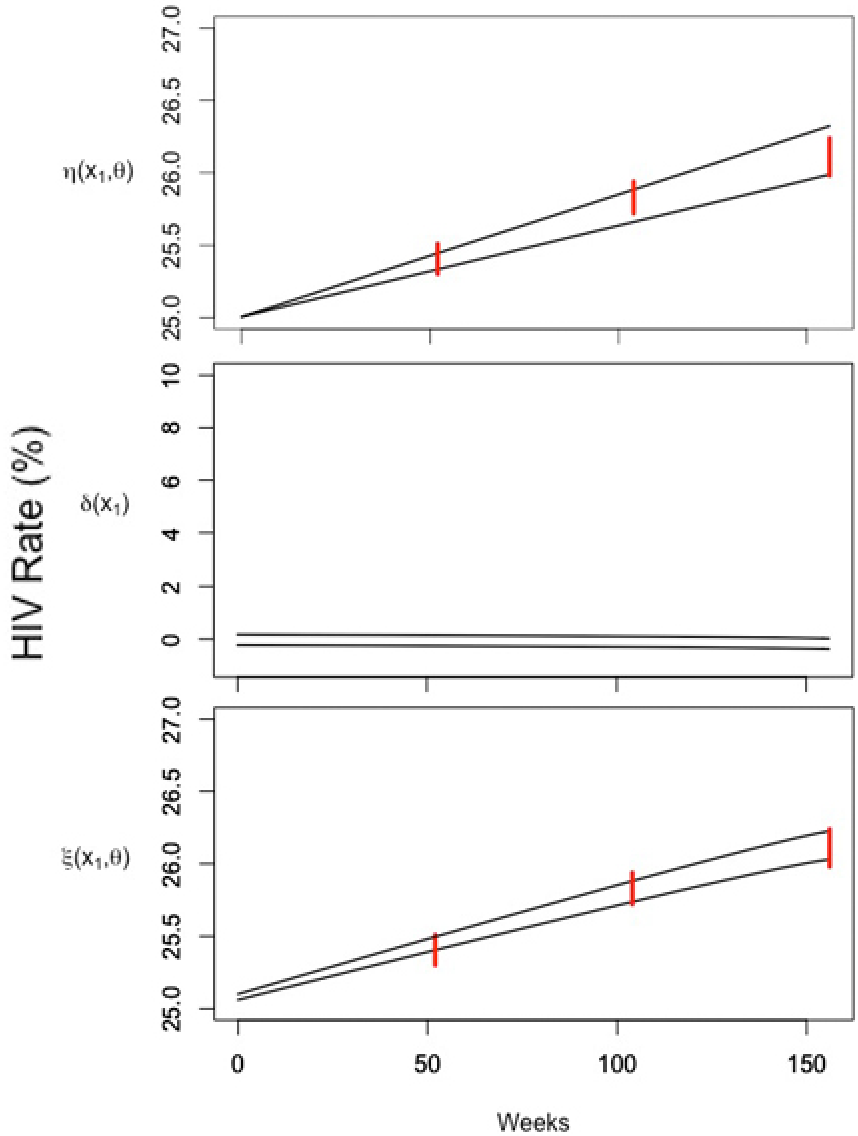


FIGURE 5.9: Top: posterior 95% credible interval for the calibrated ABM, $\eta(x_1, \theta)$. Middle: posterior 95% credible interval for the discrepancy function, $\delta(x_1)$. Bottom: posterior 95% credible interval for prediction of the network trajectory, $\xi(x_1, \theta) = \eta(x_1, \theta) + \delta(x_1)$.

5.6 Discussion

This application demonstrates the appeal in leveraging emulator techniques in an ABM setting. The initial analysis of the network, primarily the MMSB, provided valuable insight into the important variables that should go into the development of an ABM for this system. In a case such as this, when there is a clear quantity of interest in the ABM (in this case HIV rate), identifying the relevant input settings which account for the most uncertainty makes the development of an emulator for this application fairly straightforward. The network data and additional information available fit well into the Higdon et al. (2008) framework.

A limitation of this simulation is that, while deaths are modeled implicitly, no new individuals enter the network. Incorporating a more dynamic population into this model would make it more generalizable and allow for the incorporation of more interventions. Additionally, network evolution models for a fixed node set, such as described in Banks and Carley (1996) could be incorporated to expand this analysis.

An important component of the ABM/emulator approach for examples of this type is that one can infer important quantities in hard-to-reach populations, where information is often limited. One should be mindful of the potential issues with extrapolation discussed in section 5.5.

6

ABM Application: Heroin Market

6.1 Overview

An application for ABM techniques described earlier involves a model for a heroin market based on ethnographic research in Hoffer (2005). The model is a simulation of the Larimier open-air heroin market in Denver, Colorado during 1992-1996 as part of the Illicit Drug Market Simulation (IDMS) project. The of the development of the model have been discussed in detail prevoiusly in Hoffer et al. (2009). The model was implemented in Repast (North et al., 2013).

6.2 Model Background

The heroin market model consisted of six types of agents: customers, street dealers, street brokers, private dealers, police and the homeless. The full version of the model involves complex processes relating to customers' drug concentration levels and a detailed decision hierarchy which drives their behavior. Because the customer agents have the most extensive set of rules, we focus our model reduction on customer behavior processes in the physical market.

In the simulations for the model, we fixed the number of agents as follows: 200 customers, 20 street dealers, 25 street brokers, 25 private dealers, 100 homeless and 1 police officer.

6.3 Model Reduction

The objective of model simplification process was two-fold: first, to determine the optimal time step for the model and then to develop statistical approximations to the complex decision agents execute on the market. While both of these approaches should significantly reduce the run time for the model, they have different implications.

As a means of measuring the change in the model through the reduction process, we identified five key summary statistics:

1. The probability of a customer obtaining heroin.
2. The probability a customer gets arrested.
3. The amount of time a customer spends on the market.
4. The purchase mode of customer (street dealer/street broker).
5. The probability a customer is invited to a private dealer.

These five quantities are the most significant in terms of the behavior of customer agents on the market.

6.3.1 Time step determination

The initial version of the full model had a time step of one minute. At each tick of the model, one minute of time is simulated, and each agent makes a decision about their behavior, conditional on their previous state.

In order to determine the optimal time step, we ran 100 simulations of the model at increasing time step values and recorded the summary statistics. Averages of the summary statistics at each time step are shown in table 5.1.

Table 6.1: Averages of quantities of interest for the heroin market model by time step. The averages show clear divergence from the 1 minute values as the time step increases.

	1 min	2 min	3 min	5 min	10 min	60 min
Probability obtain drug	0.53	0.44	0.4	0.38	0.33	0.053
Probability of arrest	0.00029	0.00019	0.00024	0.00018	0.00014	0.000026
Average time in market (minutes)	93.2	100.8	102.9	103.3	105.5	169.7
Percent private dealer	57.9%	64.2%	64.7%	67.5%	66.2%	85.5%
Percent street dealer	23.6%	19.7%	20.8%	19.7%	23.4%	10.6%
Percent street broker	18.5%	16.1%	14.5%	12.8%	10.4%	3.9%
Probability of private dealer invite	0.069	0.058	0.051	0.042	0.029	0.0013

Based on table 5.1, we see a clear change in the summary statistics as the time step changes. Below are plots of the summary statistics' averages versus log time step.

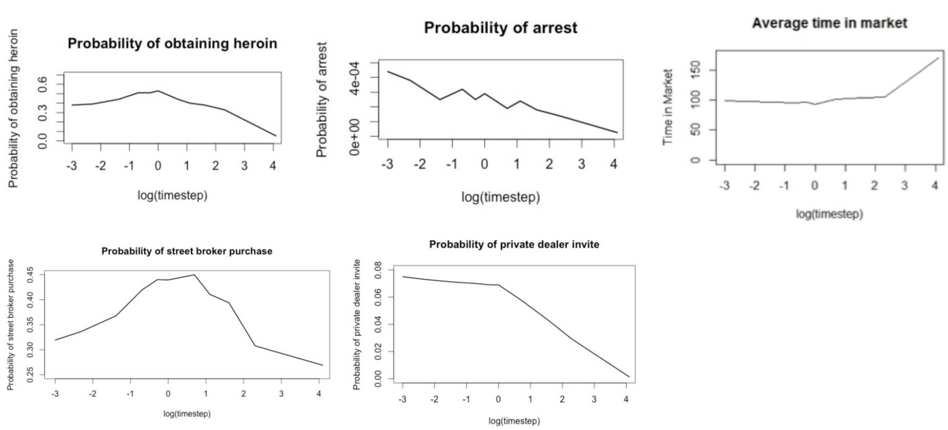


Figure 6.1: Plots of quantities of interest for the heroin market versus log time step.

The Figure 5.1 shows that, even at 2 minutes, certain key statistics depart from their values under the 1 minute time step. Based on these trends, we chose to maintain the original 1 minute time step.

6.3.2 Decision approximation

While on the physical market, customers go through a complex sequence of decisions that drive their behavior and interactions. Our goal is to simplify this entire process by developing statistical approximations to the decision process (Zou et al., 2012).

The main focus of our model reduction was on replacing the complex decision processes of customer agents with a probabilistic method of governing their behavior. In the reduced model, customer's behavior will be determined by draws from conditional distributions based on simulation results.

The implications of this approach are two-fold: first, by replacing the customer decision process with draws from appropriate distributions, we eliminate the need for certain other types of agents within the model, and secondly, we must establish a hierarchy of draws from those distributions, as some quantities need to be conditioned on others.

The decision hierarchy will proceed in the following way:

1. Draw a customer's probability of obtaining the drug p_o. A customer's market trip outcome $x \in \{0, 1\}$ for failure or success will be determined by a Bernoulli trial with the drawn rate p_o.

2. Conditional on π, draw a time for the customer to spend on the market, t_m.

3. Conditional on $x = 1$, draw a probability of arrest p_a. (Note that if the $x = 0$, $p_a = 0$).

4. Draw the probability of purchase form a street broker p_{sb} to determine purchase mode of either street dealer or street broker.

5. Conditional on $x = 1$, draw the amount of drug that will be purchased a_p (this will be determined by the customer's addiction level).

6. Conditional on a successful street broker purchase, draw the probability of an invitation to a private dealer p_i.

An alternative decision hierarchy uses logistic regression instead of beta distributions to draw the probability of obtaining heroin p_o and the probability of an invition to a private dealer p_i.

6.4 Model Comparison

The quantities of interest used to compare model output are average customer money, average customer drug inventory, average customer addiction and average customer drug concentration. A simple approach for model comparison is to see if the mean or median quantities from the reduced models fall within some tolerance of the full model (Bayarri et al., 2007b). A somewhat more sophistocated method for model comparison uses multiple hypothesis tests. For each of the four quantities of interest, the mean of the full model will be compared to the reduced model at each time point and apply a multiple testing correction procedure to obtain adjusted p-values testing the hypothesis that the full and reduced model outputs are (approximately) the same (Cox and Lee, 2008). Figures corresponding to the latter approach are presnted in Appendix B. The utility of both of these approaches is that they identify whether the models are producing similar output. They fail to aid in determining the cause of model discordance.

6.5 ABC applied to reduced model

ABC methods encompass the utility of the model comparison approaches described above in addition to providing insight into which parameters affect our quantities of interest.

Our objective is to determine optimal parameter settings in the reduced model to accurately reproduce output from the full model.

To continue our analysis of the models for the heroin market, we can apply ABC methods to the models. For the purpose of studying this market, we restrict the set of parameters of interest. We use a subset six statistics that were used to develop the reduced models to serve as our Θ. Let $\Theta = (p_o, p_a, p_d)$ representing the probability of a customer obtaining heroin on a trip to the market, the probability of a customer getting arrested in the market, and the probability of a customer being invited to a private dealer. Consistent with the development of a reduced ABM for the market, we let average customer drug inventory $\mathbf{c}_d$, average customer money $\mathbf{c}_m$ and average customer addiction $\mathbf{c}_a$ serve as the summary statistic $\mathbf{S}_y$ for model output.

An issue with the flexibility of this model is that, in the framework of ABC, there are parameters which may change throughtout the simulation. For the sake of this model, fixed parameter values for an entire simulation will give a constrained parameter range compared to dynamic values.

Using an ABC SMC approach, we establish uniform prior distributions for Θ, $p_o, p_a, p_d \sim U(0, 1)$. We use a uniform perturbation kernel for all three parameters, $K_t = \sigma U(-1, 1)$ with $\sigma = 0.05$.

We simulated 500 trajectories from the full model and used the means of the three summary statistics at each of the 365 days of the simulation for comparison with simulated data. The data for each of the summary statistics were then scaled to fall in [0,1].

Let the distance $\rho(S_{y_0}, S_y)$ between the output of the full model $\mathbf{S}_{y_0} = \{\mathbf{c}_{d_0}, \mathbf{c}_{m_0}, \mathbf{c}_{a_0}\}$ and the reduced model $\mathbf{S}_y = \{\mathbf{c}_d, \mathbf{c}_m, \mathbf{c}_a\}$ be the scaled sum of absolute errors,

$$\frac{1}{365} \sum_{i=1}^{365} |c_{d_0}[i] - c_d[i]| + |c_{m_0}[i] - c_m[i]| + |c_{a_0}[i] - c_a[i]|$$

.

We generated $N = 5000$ particles in each population, To ensure gradual transition between populations, we chose $T = 5$ populations. Because the maximum distance of a full model simulation from the mean was 0.183, this was chosen as the smallest tolerance, giving us $\epsilon = (0.549, 0.4575, 0.366, 0.2745, 0.183)$. The sequence of inferred distributions for all three parameters are shown in Figure 6.2.

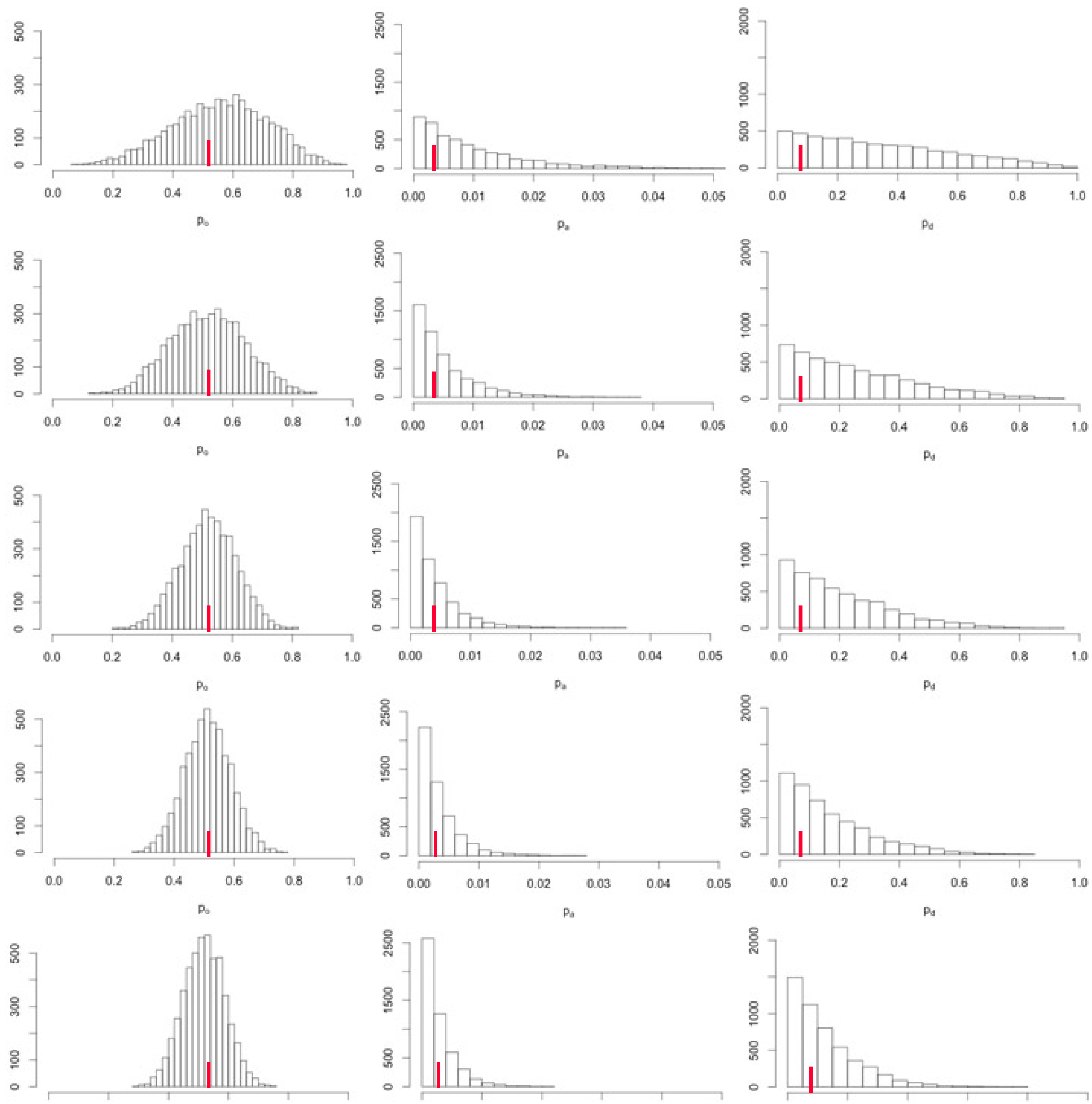


FIGURE 6.2: Histograms of populations 1 (top row) through 5 (bottom row; approximation of posterior distribution) of $\Theta = (p_o, p_a, p_d)$ from ABC SMC approach with $T = 5$ populations. Red lines indicate means from simulations of the full model.

As can be seen in Figure 6.2, the posterior for p_a showed the least variation, which is to be expected as customers are arrested infrequently (street dealers are the most frequently arrested agents). The posterior for p_o was centered around the average customer success rate in the market, 0.53. The posterior for p_d showed the most variation, but this is due to the fact that customers money, drug inventory and addiction levels follow similar patterns whether they purchase heroin in the market or from private dealers.

The analyses were also carried out with different distance functions, including sum of squared error and sum of absolute error without having scaled the summary statistics. Additionally, we explored the 'vanilla' ABC rejection algorithm and ABC MCMC for this application. The resulting approximate posterior distributions were very similar in all cases.

An alternative approach would be to apply ABC at each time step of the model and obtain a posterior distribution of the quantities of interest at each time step, across multiple simulations. This approach would yield statistics more consistent with those obtained in the model simplification process, but would be much more expensive.

6.6 Model Complexity

As discussed in chapter 2, understanding the relative complexity of models is an inportant concept. In this example, we have multiple models of a system and some intuition as to which is more complex based on the strategy for developing the models.

Based on a naive approach, the full model is more complex than the reduced model since the full model takes longer than the reduced model to run a simulation of a fixed period of time. The average run-time for a simulation of one year of market activity for the full model took an average of 32.3 minutes, compared to an average of 18.3 minutes for the reduced model. This represents a 43.2% reduction in run-time.

Average run-times showed similar reduction as the simulation length varied. This simple assessment is consistent with our intuition about the relative complexity of the two models. Additionally, the program for the reduced model is shorter than the full model, showing the reduced model to be less complex from a Kolmogorov-based complexity comparison.

Turning to more rigorous measures of complexity, we can examine the complexity of the models using compression algorithms. Using the convention that a more complex model will have a higher compression ratio, we examine the full and reduced models.

Using a Lempel-Ziv compression algorithm, the full model was found to have a compression ratio of 1.64 (2263620 KB uncompressed; 1384021 KB compressed), while the reduced model had a compression ratio of 1.42 (1910329 KB uncompressed; 1344286 KB compressed). Based on a Huffman coding algorithm, the full model was found to have a compression ratio of 1.32 (2263620 KB uncompressed; 1714864 KB compressed), while the reduced model had a compression ratio of 1.14 (1910329 KB uncompressed; 1675727 KB compressed). Under both algorithms, the compression-ratio complexity measure is consistent with our previous complexity assessment of the two models.

Table 6.2: Compression-ratio complexity measures for full and reduced versions of the heroin market ABM.

	Compression Ratio (LZ)	Compression Ratio (Huffman)
Full Model	1.64	1.32
Reduced Model	1.42	1.14

6.7 Discussion

In developing the reduced version of the heroin market ABM, while the output space has been preserved, the reduced model has adjusted the input space, as well as the some of the processes of the full model. In light of this, investigation of the equivalence of the full and reduced models in this application relates to the discussion of model equivalence in the case of partial overlap in the input spaces, as presented in Section 2.3.1. That case, as well as the case in which one input space (the full model) contains the other input space (the reduced model) are important concepts related to ABM equivalence assessment in the model reduction process.

The results of ABC methods in this instance highlight their utility for inference in an ABM setting. While, for this particular model, some distributions had already been obtained through the model reduction process, the ABC approach obtained distributions based on certain features of the market that experts deemed important to reproduce. In many cases with ABMs, there are a number of features of the system being simulated that are known and identified as important, and which can be incorporated into $S(y)$ in the ABC methods.

Analysis of the relative complexity of the full and reduced models reveals, in this instance, that our intuitive understanding of model complexity is consistent with our findings based on two approaches of assessing complexity. While this area is in need of further exploration, our initial results are promising. Combining ABM complexity analysis with comparisons of model equivalence can provide a framework for model selection which encourages parsimony. While there has been some discussion of the various programming languages in which one can develop ABMs, this is a topic of particular importance when considering model complexity. Models written in R or NetLogo are more straightforward for copmlexity assessment than models written in Repast, which have several associated files for each model, many of which need to be incorporated into the analysis of model complexity.

Further exploration of the topic will include other possible measures of model complexity, as well as investigation of cases where the naive run time assessment, the Kolmogorov-based complexity assessment and the compression-ratio assessment do not agree on the relative complexity of a set of models. Additionally, a scaling factor or function which can relate the complexity measures to one another is a concept which would be fairly straightforward to examine and develop.

Appendix A

Supplemental Figures for Meningitis Model

Below are figures displaying states affected by the meningitis outbreak and state-by-state case counts from the CDC.

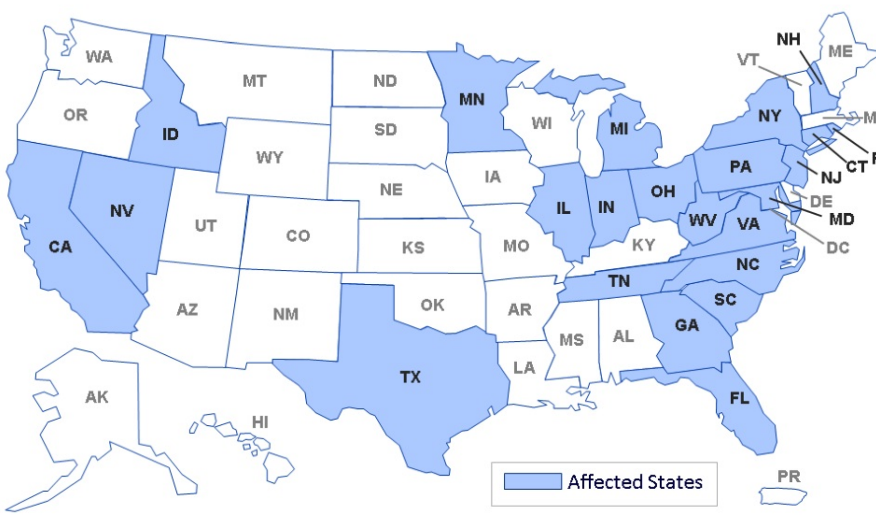


FIGURE A.1: Map of states with facilities which received contaminated Methylprednisolone Acetate (PF)

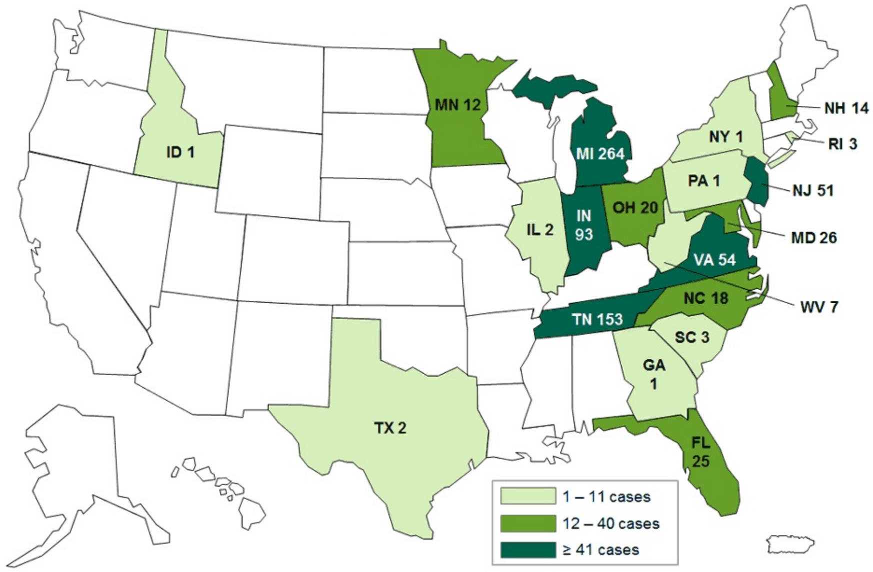


FIGURE A.2: Map of case counts by state for the 2012-2013 fungal meningitis outbreak

Appendix B

Additional IDMS Model Figures

Below are additional figures from the IDMS heroin market ABM presented in chapter 5.

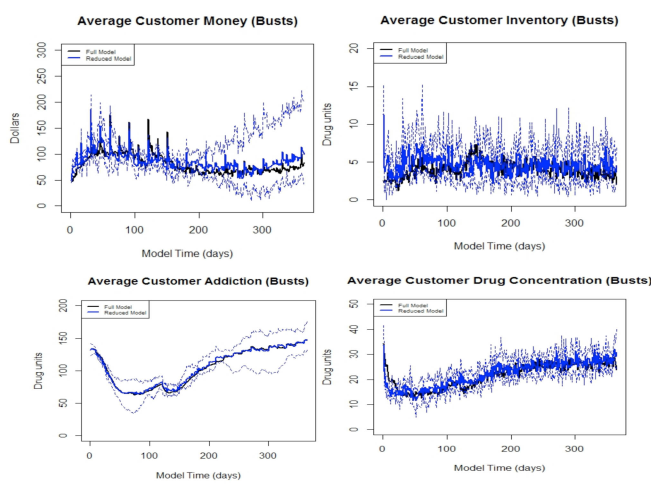


FIGURE B.1: Comparison of summary statistics for full and reduced versions of the IDMS model. Dashed lines represent 2.5% and 97.5% output quantiles.

FIGURE B.2: Adjusted p-values on negative-log scale for each of the four summary statistics comparing the reduced model to the full model.

Bibliography

Aeschbacher, S., Beaumant, M., and Futschik, A. (2012), "A Novel Approach for Choosing Summary Statistics in Approximate Bayesian Computation," *Genetics*, 192, 1027–1047.

Airoldi, E. M., Blei, D., Fienberg, S., and Xing, E. (2008), "Mixed Membership Stochastic Blockmodels," *Journal of Machine Learning Research*, 9, 1981.

Alexander, J. W. (1922), "A proof and extension of the Jordan-Brouwer separation theorem," *Transactions of the American Mathematical Society*, 23, 333–349.

Anderson, R. D. (1967), "On topological infinite deficiency," *Michigan Mathematics Journal*, 14, 365–383.

Armbruster, B., Roy, S., Kapur, A., and Schneider, J. A. (2013), "Sex Role Segregation and Mixing among Men Who Have Sex with Men: Implications for Biomedical HIV Prevention," *PloS one*, 8, e70043.

Axtell, R., Axelrod, R., Epxtein, J. M., and Cohen, M. (1996), "Aligning simulation models: A case study and results," *Computational and Mathematical Organization Theory*.

Banks, D. L. and Carley, K. M. (1996), "Models for network evolution," *Journal of Mathematical Sociology*, 21, 173–196.

Bastos, L. S. and O'Hagan, A. (2009), "Diagnostics for Gaussian Process Emulators," *Technometrics*, 51, 425–438.

Bayarri, M. J., Berger, J. O., Cafeo, J. A., Garcia-Dotano, J., Liu, F., Palomo, J., Parthasarathy, R. J., Paulo, R., and Sacks, J. (2007a), "Computer model validation with functional output," *The Annals of Statistics*, 35, 1874–1906.

Bayarri, M. J., Berger, J. O., Paulo, R., Sacks, J., Cafeo, J. A., Cavendish, J., Lin, C. H., and Tu, J. (2007b), "A Framework for Validation of Computer Models," *Technometrics*, 49, 138–154.

Beaumont, M. A. (2006), "Approximate Bayesian computation in evolution and ecology," *Annual Review of Ecology, Evolution, and Systematics*, 41, 379–406.

Beaumont, M. A., Zhang, W., and Balding, D. (2002), "Approximate Bayesian computation in population genetics," *Genetics*, 162, 2025.

Beaumont, M. A., Cornuet, J., Marin, J., and Robert, C. (2009), "Adaptive Approximate Bayesian computation," *Biometrika*, 96, 983–990.

Berger, J. O. (2003), "Could Fisher, Jeffreys and Neyman have agreed on testing?" *Statistical Science*, 18, 1–32.

Berger, J. O., de Olivera, V., and Sans, B. (2001), "Objective Bayesian Analysis of Spatially Correlated Data," *Journal of the American Statistical Association*, 96, 1365–1374.

Bharathy, G. and Silverman, B. (2010), "Validating Agent Based Social Systems Models," *Proceedings of the 2010 Winter Simulation Conference*.

Blei, D. and Jordan, M. (2003), "Latent Dirichlet allocation," *Journal of Machine Learning*, 3, 993–1022.

Bliznyuk, N., Ruppert, D., Shoemaker, C. A., Regis, R., Wild, S., and Mugunthan, P. (2008), "Bayesian Calibration of Computationally Expensive Models Using Optimization and Radial Basis Function Approximation," *Journal of Computational an Graphical Statistics*, 17, 270–294.

Blum, M. G. B. and Francois, O. (2010), "Non-linear regression models for Approximate Bayesian Computation," *Statistics and Computing*, 20, 63–73.

Bonabeau, E. (2002), "Agent-based modeling: Methods and techniques for simulating human systems," *Proceedings of the National Academy of Sciences of the United States of America*, 99, 7280–7287.

Bonassi, F. (2013), "Approximate Bayesian Computation for Complex Dynamic Systems," Ph.D. thesis, Duke University.

Brauer, F. and Castillo-Chavez, C. (2012), *Mathematical Models in Population Biology and Epidemiology*, Springer, New York.

Breiman, L. (2001), "Random Forests," *Machine Learning*, 45, 5–32.

Breiman, L., Friedman, J. H., Olshen, R., and Stone, C. (1984), *Classification and Regresion Trees*, Chapman and Hall, Belmont, CA.

Brouwer, L. E. J. (1912), "Zur Invarianz des n-dimensionalen," *Gebiets Mathematische Annalen*, 72.

Cangelosi, R. and Goriely, A. (2007), "Component retention in principal component analysis with application to cDNA microarray data," *Biology Direct*, 2, 1–21.

Centers for Disease Control & Prevention (2013a), "CDC - Meningitis - Fungal Meningitis," www.cdc.gov/meningitis/fungal.html, Accessed: 2013-12-27.

Centers for Disease Control & Prevention (2013b), "Multistate Fungal Meningitis Outbreak Investigation," www.cdc.gov/hai/outbreaks/meningitis.html, Accessed: 2013-10-27.

Chapman, T. A. (1972), "Canonical Extnesions of Homeomorphisms," *General Topology and its Applications*, 2, 227–247.

Chattoe, E. (2003), *Agent-Based Computational Demography: Using Simulation to Improve Our Understanding of Demographic Behaviour*, chap. The Role of Agent-Based Modelling in Demographic Explanation, Physicia-Verlag, Heidelberg.

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

Chwif, L., Barretto, M. R. P., and Paul, R. J. (2000), "On simulation model complexity," *Simulation Conference, 2000. Prodeecings Winter*, 1, 449–455.

Cox, D. D. and Lee, J. S. (2008), "Pointwise testing with functional data using the Westfall-Young randomization method," *Biometrika*, 95, 621–634.

Craig, P. S., Goldstein, M., Rougier, J. C., and Seheult, A. H. (2001), "Bayesian Forecasting for Complex Systems using Computer Simulators," *Journal of the American Statistical Association*, 96, 717–729.

Dandona, L., Dandona, R., Gutierrez, J. P., Kumar, G. A., McPherson, S., Bertozzi, S. M., and Asci-FFP (2005), "Sex behaviour of men who have sex with men and risk of HIV in Andhra Pradesh, India," *Aids*, 19, 611–619.

DelMoral, P., Doucet, A., and Jasra, A. (2012), "An adaptive sequential Monte Carlo method for approximate Bayesian computation," *Statistics and Computing*, 22, 1009–1020.

Diekmann, O. and Heesterbeek, J. A. P. (2000), *Mathematical epidemiology of infectious diseases*, John Wiley & Sons, New York.

Dimitrakakis, C. and Tziortziotis, N. (2013), "ABC Reinforcement Learning," .

Drovandi, C. C. and Pettitt, A. N. (2011), "Estimation of parameters for macroparasite evolution using Approximate Bayesian Computation," *Biometrics*, 67, 225–233.

Epstein, J. and Axtell, R. (1996), *Growing artificial societies: social science from the bottom up*, The Brookings Institution, Washington, DC.

Evans, S. and Bush, S. F. (2001), "Symbol Compression Ratio for String Compression and Estimation of Kolmogorov Complexity," submitted to 2002 IEEE International Symposium on Information Theory, to be held June.

Fagiolo, G., , Birchenall, C., and Windrum, P. (2007), "Special Issue on 'Empirical Validation in Agent-Based Models'," *Computational Economics*, 30.

Fearnhead, P. and Prangle, D. (2012), "Constructin gsummary statistics for approximate Bayesian comuptation: Semi-aucomatic approximate Bayesian computation (with discussion)," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74, 419–474.

Forzani, L. (2006), "Principal Component Analysis: A conditional point of view," Tech. rep., University of Minnesota.

Gammerman, A. and Vovk, V. (1999), "Kolmogorov complexity: sources, theory and applications," *The Computer Journal*, 42.

Gardner, M. (1970), "Mathematical Games: The fantastic combinations of John Conway's new solitaire game 'life'," *Scientific American*, 223, 120–123.

Gramacy, R. and Lee, H. K. (2008), "Bayesian Treed Gaussian Process Models With an Application to Computer Modeling," *Journal of the American Statistical Association*, 103, 1119–1130.

Grimm, V., Berger, U., Bastiansen, F., Eliassen, S., Ginot, V., Giske, J., Goss-Custard, J., Grand, T., Heinz, S. K., Huse, G., Huth, A., Jepsen, J. U., Jorgensen, C., Mooij, W. M., Muller, B., Pe'er, G., Piou, C., Railsback, S. F., Robbins, A. M., Robbins, M. M., Rossmanith, E., Ruger, N., Strand, E., Souissi, S., Stillman, R. A., Vabo, R., Visser, U., and Deangelis, D. L. (2006), "A standard protocol for describing individual-based and agent-based models," *Ecological Modelling*, 198, 115–126.

Grimm, V., Berger, U., DeAngelis, D. L., Polhill, J. G., Giske, J., and Railsback, S. F. (2010), "The ODD protocol: a review and first update," *Ecological Modelling*, 221, 2760–2768.

Hastie, T., Tibshriani, R., and Friedman, J. (2009), *The Elements of Statistical Learning: Data Mining, Inference and Prediction, 2nd Edition*, Springer, New York.

Heard, D. and Schneider, J. (2013), "Bayesian Network Analysis: HIV Risk in Southern Indian Community," Presented at the 10th Internaional Conference on Health Policy Statistics in Chicago, IL.

Higdon, D. (1998), "A process-evolution approach to modeling temperatures in the north Atlantic Ocean," *Journal of Environmental and Ecological Statistics*, 5, 173–190.

Higdon, D., Kennedy, M., Cavendish, J. C., Cafeo, J. A., and Ryne, R. D. (2004), "Combining field data and computer simulations for calibration and prediction," *SIAM Journal on Scientific Computing*, 26, 448–466.

Higdon, D., Gattiker, J., Williams, B., and Rightley, M. (2008), "Computer Model Calibration Using High-Dimensional Output," *Journal of the American Statistical Association*, 103, 570–583.

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

Hoffer, L. D. (2005), *Junkie Business: The Evolution and Operation of a Heroin Dealing Network (Case Studies on Contemporary Social Issues)*, Wedsworth, Beverly, MA.

Hoffer, L. D., Bobashev, G., and Morris, R. J. (2009), "Researching a Local Heroin Market as a Complex Adaptive System," *American Journal of Community Psychology*, 44.

Hooten, M. B. and Wikle, C. K. (2010), "Statistical Agent-Based Models for Discrete Spation-Temporal Systems," *Journal of the American Statistical Association*, 105, 236–248.

Hooten, M. B., Leeds, W. B., Fletcher, J., and Wikle, C. K. (2011), "Assessing First-Order Emulator Inference for Physical Parameters in Nonlinear Mechanistic Models," *Journal of Agricultural, Biological and Environmental Sciences*, 16, 475–494.

Hufman, D. A. (1952), "A Method for the Construction of Minimum Redundancy Codes," *Proceedings of the IRE*, 40, 1098–1191.

Hunter, D. R., Goodreau, S. M., and Handcock, M. S. (2008), "Goodness of Fit of Social Network Models," *Journal of the American Statistical Association*, 103, 248–258.

Jackson, J. (1991), *A User's Guide to Principal Components*, John Wiley & Sons, New York.

Jin, F., Jansson, J., Law, M., Prestage, G. P., Zablotska, I., Imrie, J. C., Kippax, S. C., Kaldor, J. M., Grulich, A. E., and Wilson, D. P. (2010), "Per-contact probability of HIV transmission in homosexual men in Sydney in the era of HAART," *AIDS*, 24, 907–913.

Jolliffe, I. (2005), *Principal component Analysis*, John Wiley & Sons, New York.

Jolliffe, I. T. (1982), "A note on the use of principal components in regression," *Journal of the Royal Statistical Society, Series C*, 31, 300–303.

Jordan, M. and Wainwright, M. J. (2003), "Graphical models, exponential families and variational inference," Tech. Rep. 649, Department of Statistics, University of California, Berkeley.

Kasmire, J., Beek, J. V. D., and Vavier, M. (2013), "Optimising Emergence (Version 1)," *CoMSES Computational Model Library*, Retrieved from: http://www.openabm.org/model/3824/version/1.

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

Kemeny, J. (1955), "Man Viewed as a Machine," *Scientific American*, 192, 58–67.

Kennedy, M. C. and O'Hagan, A. (2000), "Predicting the Output from a Complex Computer Code When Fast Approximations Are Available," *Biometrika*, 87, 1–13.

Kennedy, M. C. and O'Hagan, A. (2001), "Bayesian Calibration of Computer Models," *Journal of the Royal Statistical Society*, 68, 425–464.

Khalatur, P. G., Novikov, V. V., and Khokhlov, A. R. (2003), "Conformation-dependent evolution of copolymer sequences," *Physical Review E*, 67.

King, J. R. and Jackson, D. A. (1999), "Variable Selection in Large Environmental Data Sets Using Principal Components Analysis," *Environmetrics*, 10, 66–77.

Kleijinen, J. P. C. (1999), "Validation of models: Statistical techniques and data availability," *Proceedings of the 1999 Winter Simulation Conference*.

Krzanowski, W. J. (1987), "Selection of Variables to Preserve Multivariate Data Structure, Using Principal Components," *Journal of the Royal Statistical Society, Series C*, 36, 22–33.

Leary, S., Bhaskar, A., and Keane, A. (2003), "Optimal orthogonal-array-based Latin hypercubes," *Journal of Applied Statistics*, 30, 585–598.

Li, M. and Vitaanyi, P. M. B. (2008), *An introduction to Kolmogorov complexity and its applications*, Springer, New York.

Liu, F. and West, M. (2009), "A Dynamic Modelling Strategy for Bayesian Computer Model Emulation," *Bayesian Analysis*, 4, 393–412.

Lopes, D. (2011), "Development and Implementation of Bayesian Computer Model Emulators," Ph.D. thesis, Duke University.

Lorek, H. and Sonnenschein, M. (1999), "Modelling and simulation software to support individual-oriented ecological modelling," *Ecological Modelling*, 115, 199–216.

Lorway, R., Reza-Paul, S., and Pasha, A. (2009), "On Becoming a Male Sex Worker in Mysore: Sexual Subjectivity, 'Empowerment' and COmmunity-Based HIV Prevention Research," *Medical Anthropology Quarterly*, 23, 142–160.

Margvelashvili, N. (2011), "Sequential data assimilation in fine-resolution models using error-subspace emulators: Theory and preliminary evaluation," *Journal of Marine Systems*, 90, 13–22.

Marjoram, P., Molitor, J., Plagnol, V., and Tavar, S. (2003), "Markov chain Monte carlo without likelihoods," *Proceedings of the National Academy of Sciences*, 100, 15324–15328.

Marks, R. E. (2012), "Analysis and Synthesis: Multi-Agent Systems in the Social Sciences," *Knowledge Engineering Review*, 27.

Marks, R. E. (2013), "Validation and Model Selection: Three Similarity Measures Compared," *Complexity Economics*, 22, 41–61.

Martens, H. and Naes, T. (1989), *Multivariate calibration*, John Wiley & Sons, New York.

Massy, W. F. (1965), "Principal Components Regression in Exploratory Statistical Research," *Journal of the American Statistical Association*, 60, 234–256.

Mevik, B. H. and Wehrens, R. (2007), "The pls package: principal component and partial least squares regression in R," *Journal of Statistical Software*, 18, 1–24.

Michigan Headache & Neurological Institute (2012), "Program Evaluation Data," http://www.mhni.com/why-mhni/program-evaluation-data, Accessed: 2013-10-28.

Niazi, M., Hussain, A., and Kolberg, M. (2009), "Verification and Validation of Agent-Based Simulations using the VOMAS approach," *Proceedings of the Third Workshop on Multi-Agent Systems and Simulation*.

North, M. J., Collier, N. T., Ozik, J., Tatara, E., Altaweel, M., Macal, C. M., Bragen, M., and Sydelko, P. (2013), "Complex Adaptive Systems Modeling with Repasy Simphony," *Complex Adaptive Systems Modeling, Springer, Heidelberg, FRG.*

O'Hagan, A. (1978), "Curve fitting and optimal design for prediction," *Journal of the Royal Statistical Society*, 40, 1–42.

Pappas, P. G., Kontoyiannis, D. P., Perfect, J. R., and Chiller, T. M. (2013), "Real-time treatment guidelines: considerations during the Exserohilum rostratum outbreak in the United States," *Antimicrobial agents and chemotherapy*, 57, 1573–1576.

Pitts, A., Pitts, M., Smith, G., Grierson, J., Smith, A., McNally, S., and Couch, M. (2011), "Versatility and HIV vulnerability: investigating the proportion of Australian gay men having both insertive and receptive anal intercourse," *Journal fo Sexual Medicine*, 8, 2164–2171.

Polhill, J. G., Parker, D., Brown, D., and Grimm, V. (2008), "Using the ODD Protocol for Describing Three Agent-Based Social Simulation Models of Land-Use Change," *Journal of Artificial Societies and Social Simulation*, 11.

Prishlyak, A. O. (2002), "Topological equivalence of smooth functions with isolated critical points on a closed surface," *Topology and its Applications*, 119, 257–267.

Pritchard, J., Perez-Lezaun, M., and Feldman, M. (1999), "Population growth of human Y chromosomes: a study of Y chromosome microsatellites," *Molecular Biology and Evolution*, 16, 1791.

Railsback, S. F., Lytinen, S. L., and Jackson, S. K. (2006), "Agent-based Simulation Platforms: Review and Development Recommendations," *Simulation*, 82, 609–623.

Rasouli, S. and Timmermans, H. (2013), "Using Emulators to Approximate Predicted Performance Indicators in Complex Micro-simulation and Multi-Agent Models of Travel Demand," *Transportation letters: International Journal of Transportation Research*, 5, 609–623.

Robinson, A. P., Duursma, R. A., and Marshall, J. D. (2005), "A regression-based equivalence test for model validation: shifting the burden of proof," *Tree Physiology*, 25, 903–913.

Rodgers, J. L. and Nicewander, W. A. (1988), "Thirteen ways to look at the correlation coefficient," *The American Statistician*, 42, 59–66.

Sacks, J., Welch, W., MItchell, T., and Winn, H. (1989), "Design and Analysis of Computer Experiments," *Statistical Science*, 4, 409–423.

Salzmann, H. (1969), "Geometries on surfaces," *Pacific Journal of Mathematics*, 29, 397–402.

Sanchez, S. M. (2001), "Abc's of output analysis," *Proceedings of the 2001 Winter Simulation Conference*.

Santner, T. J., Williams, B. J., and Notz, W. I. (2003), *Design and analysis of computer experiments*, Springer, New York.

Schneider, J., Kapur, A., Schumm, P., Laumann, E., Rani, S., and Oruganti, G. (2011a), "A novel men who have sex with men (MSM) digital communication network analytic approach: cell phone assisted network detection and identification (CANDID)," Presented at 6th IAS Conference on HIV Pathogenesis and Treatment, Abstract no. MOPE3280.

Schneider, J. A., Kapur, A., Oruganti, G., Schumm, P., and Laumann, E. (2011b), "A Novel Hybrid Egocentric-Archival Network Characterization Approach Using Cell Phones to Identify Bridging Actors in a High Risk HIV/STI Network in India: The Secunderabadi Mens Study (SMS)." Presented at Sunbelt XXXI: International Network for Social Network Analysis.

Schnelling, J. (1971), "Dynamic models of segregation," *Journal of Mathematical Sociology*, 1, 143–186.

Shang, F., Uber, J. G., and Rossman, L. (2008), "EPANET Multi-Species Extension Software," *US Environmental Protection Agency EPA/600/C-10*, 2.

Sharko, V. V. (2003), "Smooth and topological equivalence of functions on surfaces," *Ukrainian Mathematical Journal*, 55, 832–846.

Sisson, S. A., Fan, Y., and Tanaka, M. M. (2007), "Sequential Monte Carlo without likelihoods," *Proceedings of the National Academy of Sciences of the United States of America*, 104, 1760–1765.

Spiegelhalter, D. J., Carlin, N. G. B. B. P., and Linde, A. V. D. (2002), "Bayesian measures of model complexity and fit," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64, 583–639.

Talbott, J. R. (2007), "Size matters: the number of prostitutes and the global HIV/AIDS pandemic," *PloS one*, 2, e543.

Tanaka, M. M., Francis, A. R., Luciani, F., and Sisson, S. A. (2006), "Using approximate Bayesian computation to estimate tuberculosis transmission parameters from genotype data," *Genetics*, 173, 1511–1520.

Tang, B. (1993), "Orthogonal array-based Latin hypercubes," *Journal of the American Statistical Association*, 88, 1392–1397.

Tavare, S., Balding, D. J., and Griffiths, R. C. (1997), "Inferring Coalescence Times from DNA Sequence Data," *Genetics*, 145, 505–518.

Teh, Y. W., Newman, D., and Welling, M. (2008), "A collapsed variational bayesian inference algorithm for latent dirichlet allocation," *Advances in Neural Information Processing Systems*, 19, 1353.

Thiele, J. C., Kurth, W., and Grimm, V. (2012), "RNetLogo: an R package for running and exploring individual-based models implemented in NetLogo," *Methods in Ecology and Evolution*, 3, 480–483.

Thomas, B., Mimiaga, M., Kumar, S., Swaminathan, S., Safren, S. A., and Mayer, K. H. (2011), "HIS in Indian MSM: Reasons for a concentrated epidemic & strategies for prevention," *Indian Journal of Medical Research*, 134, 920–929.

Thomas, M. T. C. A. J. (1991), *Elements of information theory*, John Wiley & Sons, Inc., New York.

Toni, T., Welch, D., Strelkowa, N., Ipsen, A., and Stumpf, M. P. H. (2009), "Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems," *Journal of the Royal Society Interface*, 6, 187–202.

US Census Bureau (2013), "Michigan QuickFacts from the US Census Bureau," http://quickfacts.census.gov/qfd/states/26000.html, Accessed: 2013-11-4.

US Food and Drug Administration (2012), "List 2: New England Compounding Center (NECC) Customer List SInce 5/21/2012, Sorted by Customer - With Product Information," http://www.fda.gov/downloads/Drugs/DrugSafety/FungalMeningitis/UCM325466.pdf, Accessed: 2013-10-29.

Vanpaemel, W. (2009), "Measuring model complexity with the prior predictive," *Advances in neural information processing systems*, pp. 1919–1927.

Wasserman, S. and Faust, K. (1994), *Social Network Analysis: Methods and Applications*, Cambridge University Press, Cambridge, UK.

Welleck, S. (2010), *Testing statistical hypotheses of equivalence and noninferiority*, CRC Press, Boca Raton, FL.

Welsh, A. H. (1996), *Aspects of statistical inference*, John Wiley & Sons, New York.

Westveld, A. and Hoff, P. (2012), "A mixed effects model for longitudinal relational and network data, with applications to international trade and conflict," *The Annals of Applied Statistics*, 55, 843–872.

WHO Global Health Observatory Data (2011), "Number of deaths due to HIV/AIDS," http://www.who.int/gho/hiv/epidemic$_{s}tatus/deaths_{t}ext/en/, Accessed : 2014 - 1 - 12$.

Wilensky, U. (1999), "Netlogo," http://ccl.northwestern.edu/netlogo/. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

Windrum, P., Fagiolo, G., and Moneta, A. (2007), "Empirical Validation of Agent-Based Models: Alternatives and Prospects," *Journal of Artificial Societies and Social Simulation*, 10.

Xiang, X., Kennedy, R., Madey, G., and Cabaniss, S. (2005), "Verification and validation of agent-based scientific simulation models," *Procedings of the 2005 Agent-Directed Simulation Symposium*, 37, 47–55.

XJ Technologies (2013), "AnyLogic," www.anylogic.com, Accessed: 2013-11-29.

Yang, Y., Monseraud, R. A., and Huang, S. (2004), "An evaluation of diagnostic tests and their roles in validating forest biometric models," *Canadian Journal of Forest Research*, 34, 619–629.

Ye, K. Q., Li, W., and Sudjianto, A. (2000), "Algorithmic construction of optimal symmetric Latin hypercube designs," *Journal of Statistical Planning and Inference*, 90, 145–159.

Zechman, E. M. (2011), "Agent-Based Modeling to Simulate Contamination Events and Evaluate Threat Management Strategies in Water Distribution Systems," *Risk Analysis*, 31, 758–772.

Ziv, J. and Lempel, A. (1977), "A Universal Algorithm for Sequential Data Compression," *IEEE Transactions on Information Theory*, 23, 337–342.

Ziv, J. and Lempel, A. (1978), "Compression of Individual Sequences Via Variable-Rate Coding," *IEEE Transactions on Information Theory*, 24, 530–536.

Zou, Y., Fonoberov, V. A., Fonoberova, M., Mezic, I., and Kevrekidis, I. G. (2012), "Model reduction for agent-based social simulation: coarse-graining a civil violence model," *Physical Review E*, 85.

Biography

Daniel Heard was born on August 1, 1986 in St. Louis, MO. He received his BS in Mathematics from Arizona State University in 2008 and an MA in Mathematics from St. Louis University in 2010. Subsequently, Daniel was accepted into the Ph.D. program in Statistical Science at Duke University and awarded the Duke University Dean's Graduate Fellowship. In 2013, he earned an MS degree *en route* to his Ph.D. He graduated with a Ph.D. under the supervision of Professor David Banks in May 2014.