Australian National University

Melded Bayesian Inference for Stochastic Theoretical Models with Applications in Agent Based Modelling

Mark Dawkins

November 2017

A thesis submitted in partial fulfilment of the requirements for the degree of Bachelor of Science with Honours in Statistics at the Australian National University

Declaration

This thesis contains no material which has been accepted for the award of any other degree or diploma in any University, and, to the best of my knowledge and belief, contains no material published or written by another person, except where due reference is made in the thesis.

Mark Dawkins

Acknowledgements

To my supervisors Grace and Anton. You have been generous with your time and energy and have made this a fantastic year for me. Thank you.

To my grandparents Edna, Wally, John and Judy. Thank you for raising families in which learning is encouraged.

Abstract

Bayesian melding is extended for applications to stochastic theoretical models. Agent Based models, a class of stochastic theoretical models, are investigated and it is found that the common challenge of parameter specification can be addressed with the extensions to Bayesian melding. Two versions of the extended framework are applied to the Agent Based model of bumblebee foraging behaviour published in Smolla, Alem, et al. 2016. The applications demonstrate both a comprehensive approach to parameter specification and an innovative approach to decomposing error. Posterior inference is implemented using a combination of Markov-Chain Monte Carlo and Sampling Importance Resampling algorithms.

Notation

List of theoretical modeling notation

θ = Inputs	ϕ = Outputs	β = Parameters
Θ = Input Space	Φ = Output Space	B = Parameter Space
x = Data related to inputs	y = Data related to outputs	

Notation system for expressing Bayesian melding

$q^1()$ = Contains information about inputs only	$q_{[\theta]}$ = Input marginal
$q^{1p}()$ = Contains information about input-parameters only	$q_{[\beta]}$ = Parameter marginal
$q^2()$ = Contains information about outputs only	$q_{[\theta,\beta]}$ = Input-parameter marginal
$\tilde{q}()$ = "Melded"	$q_{[\phi]}$ = Output marginal
$\hat{\tilde{q}}()$ = "Approximate melded"	$q_{[\theta,\beta,\phi]}$ = Joint

List of Bayesian melding notation

$q^{1p}_{[\theta,\beta]}(\theta,\beta)$ = "Pre-model" prior on input-parameters

$q^{2}_{[\phi]}(\phi)$ = "Pre-model" prior on outputs

$q^{1p}_{[\phi]}(\phi)$ = "Induced" prior on outputs

$\tilde{q}_{[\phi]}(\phi)$ = "Melded" prior on outputs

$\tilde{q}_{[\theta,\beta]}(\theta,\beta)$ = "Melded" prior on input-parameters

$\tilde{q}_{[\theta,\beta,\phi]}(\theta,\beta,\phi)$ = "Melded" joint prior

$\hat{\tilde{q}}_{[\theta,\beta]}(\theta,\beta)$ = "Approximate melded" prior on input-parameters

$\hat{\tilde{q}}_{[\phi]}(\phi)$ = "Approximate melded" prior on outputs

$\hat{\tilde{q}}_{[\theta,\beta,\phi]}(\theta,\beta,\phi)$ = "Approximate melded" joint prior

Contents

1. Introduction

Improvements in computer technology are empowering scientists to seek understanding in areas that were previously beyond our grasp. Specifically modern computers give scientist two "super powers": they have more processing power than ever before, which allows them to build complicated models; and they have more data than ever before, which means they have more information about the real world. In this thesis we study one technique for building complicated models called Agent Based modelling, and one technique for leveraging large data sets called Bayesian melding. The key result of this thesis will be that, with a small extension to Bayesian Melding, we can combine these two techniques resulting in a method that will allow us to comprehend the previously unknowable.

The body of this thesis is comprised of chapters two through four. In chapter two, we discuss Agent Based modelling concepts and how they relate to Bayesian melding. In chapter three, the technical aspects of Bayesian melding are introduced and extended to stochastic models. In chapter four, a specific Agent Based model relating to the foraging behaviour of bumblebees is used to demonstrate the ideas developed in earlier chapters.

Literature Review

Bayesian Melding

Bayesian melding is a novel technique, so the literature is small. Some Bayesian melding papers will be discussed in detail, with emphasis on how incorporating theory improves statistical inference and the idiosyncrasies of the particular application. Of particular interest is Ševčíková, Raftery, and Waddell 2007 which extended Bayesian melding to stochastic models in a reduced way. The Agent Based modelling literature is much larger; here the history and motivations of Agent Based modelling are discussed.

Bayesian melding incorporates theory into statistical inference by constructing Bayesian

priors that incorporate the relationships described in a theoretical model. The idea to construct priors that encode a theoretical model can be attributed to Raftery, Givens, and Zeh 1995 which proposed a technique called Bayesian synthesis. Bayesian melding was introduced in Poole and Raftery 2000 as a response to the realisation that Bayesian synthesis is subject to the Borel paradox (Wolpert 1995). The improved technique proposed in Poole and Raftery 2000 was applied to whale population dynamics. Since this first paper a literature on Bayesian melding has developed focusing mostly on subject matter applications.

Chiu and Gould 2010 demonstrated Bayesian melding in the context of ecological networks. The scientific goal of the paper was to perform inference on the average incoming, average outgoing and average diffusion of mass or energy across network nodes. Bayesian melding was chosen because subject matter theory suggests that these quantities satisfy a mass balance equation. Mass balance equations are a formulation of conservation of mass, the idea that (in a particular setting) mass can neither be created nor destroyed. Bayesian melding was used to constrain inference to the space where these average quantities obey the mass balance equation. Constraining inference to this space reduced the dimensionality of the problem and improved inference. Importantly if inference is not constrained, positive probabilities can be placed on combinations of average incoming, average outgoing and average diffusion that the theory would suggest are impossible. The approach taken in Chiu and Gould 2010 is notable in that the mass balance equation is applied to the average quantities and not to each specific node. This choice to use the theoretical model at an aggregate level allowed more data to be captured in the statistical component of the Bayesian melding procedure. This demonstrated the flexibility of Bayesian melding to capture elements of the system in either a theoretical model or a statistical model.

Alkema, Raftery, and Clark 2007 applied Bayesian melding in the context of generalised HIV/AIDs epidemics. The quantities under investigation were the underlying characteristics of an epidemic and the resulting infected rates through time. Epidemiology theory usually models epidemics with compartmental ordinary differential equation models, called susceptible-infected-recovered models. Using Bayesian melding, inference was gained jointly on all model quantities using records of the prevalence of HIV/AIDs among pregnant women at antenatal clinics. While the available data relates only to the infected rates over time Bayesian melding allowed for inference to be made on the under-

lying characteristics of the epidemic. Alkema, Raftery, and Clark 2007 demonstrate that Bayesian melding allows for inference on all model quantities, given data relating to only some.

In Ševčíková, Raftery, and Waddell 2007 an attempt is made to extend Bayesian melding to stochastic models. Bayesian melding is applied in the context of urban development. A stochastic simulation model similar to an Agent Based model describes the dynamics of urban development. Using a reduced form of Bayesian melding, inference on input quantities is gained given data on outputs. This thesis builds on the methods used in Ševčíková, Raftery, and Waddell 2007, and extends them so that all of the desirable properties of the original Bayesian melding in its full form apply to stochastic models.

Other applications of Bayesian melding include: Falk, Denham, and Mengersen 2010 where Bayesian melding is applied to the Revised Universal Soil Loss Equation (RUSLE) model of soil loss; Radtke, Burk, and Bolstad 2002 where Bayesian melding is applied in the context of forest ecosystem modelling; and Ševčıková, Raftery, and Waddell 2011 where Bayesian melding is applied to traffic modelling, allowing for the uncertainty around the long term effect of tearing down the Alaskan Way Viaduct in Seattle to be evaluated.

Agent Based modelling

The Agent Based modelling literature has a long and rich history, however it is yet to be transfered into mainstream use. Agent Based modelling is a type of computer simulation model that is premised on the idea that complex system-wide behaviour can be "grown" from the interactions of simply defined agents in a process referred to as "emergence". The literature on these ideas is often cited as deriving from John von Neumann's cellular automata. Von Neumann defined the concept of cellular automata in the 1940s, in the process of designing an abstract self replicating machine. The cellular automaton is a cell that interacts with other cells on a grid. This idea was taken and simplified in John Conway's Game of Life. In the 1970s computers became sufficiently powerful to implement the ideas of Conway and von Neumann. This allowed subject matter science to benefit from applications of Agent Based modelling. A particularly notable early work is Thomas Schelling's segregation model (Schelling 1969). His model of the racial composition of neighbourhoods showed the special power of Agent Based modelling in the social sciences. Another major work was Epstein and Axtell's Sugarscape (Epstein and Axtell 1996). It

represented an inflection point in Agent Based modelling as one of the first large and detailed simulation models. It attempted to recreate history, society and economy in a single model. For a summary of current work being done with Agent Based modelling across a range of scientific fields, see Heard et al. 2015.

Despite the promising work being done on Agent Based modelling in the 1990s and early 2000s, the field has stalled in recent years. This stall may be due to the difficulty in combining Agent Based models with data. Grazzini and M. Richiardi 2015 discuss this in the context of macroeconomics. Grazzini and Richiardi find that Agent Based models are at a disadvantage to Dynamic Stochastic General Equilibrium (DSGE) models (the most popular modelling technique in macroeconomics) because there are methods for rigorous parameter estimation in DSGE while in Agent Based modelling there are not. Paraphrasing Chen, Chang, and Du 2012, "The AB camp has to move from stage I (the capability to grow stylised facts in a qualitative sense) to stage II (the selection of the appropriate parameter values based on sound econometric techniques)."

Agent Based modelling is one approach to building computer simulation models, but there are many others and the distinctions between these approaches is not always clear. There are a number of other labels given to simulation models that share the important features of Agent Based models. Most of these techniques are essentially the same approach but are named differently for historical reasons. This category includes Agent Based Computational Economics models (ACE) and Individual Based Models (IBM). Some techniques have subtle differences but can be grouped under Agent Based modelling due to their large number of similarities. A prominent example of this is Micro-simulation modelling. Micro-simulations are like Agent Based models in that they derive complex behaviour through the interactions of agents, but are different in that the actions of agents do not necessarily reflect a behavioural strategy but instead a probability of transition between states. The commonalities and differences between Micro-simulation and Agent Based modelling are discussed in Bae et al. 2016.

2. Agent Based Modelling

Agent Based models are a sub-group of computer simulation models where complex system-wide behaviour can be "grown" from the interactions of simply defined agents. In this chapter we take a fresh look at what that may mean. We will explain why Agent Based models are used and consider their properties. We will then argue that combining Agent Based models with Bayesian melding results in a powerful new tool, which can perhaps reinvigorate the Agent Based literature.

2.1 Background on Modelling and Simulation

To understand the importance of Agent Based modelling it helps to draw a distinction between a system, a model, and a simulation:

A system is some discrete part of the real world that we wish to study. A system is a very difficult concept to define as whenever we think about it we are inevitably simplifying. Consequently it is best to think of a system as some part of the real world that exists outside of abstract thought. This may seem problematic but while a system cannot be totally conceptualised its features can be measured. Through measurement of a system we obtain information about it and this information forms the basis of understanding. In this thesis the system under study is a population of bees but one could study anything from a single molecule to a whole galaxy.

A model is the tool that we use to describe a system, for our purposes this will be a collection of mathematical expressions or computer code. As the often repeated George Box quotation goes, "All models are wrong but some are useful", implying no model will be a complete description of the real system, but a model may be able to capture some major effects that account for the properties we are interested in. So if a useful model captures these major effects then the properties of the system determine how easily one can construct a useful model. Some systems are "simple", dominated by a few major

effects and hence can be well understood with a simple description. Other systems are "complex". "Complex systems are highly composite ones, built up from very large numbers of mutually interacting subunits (that are often composites themselves) whose repeated interactions result in rich, collective behaviour that feeds back into the behaviour of the individual parts." (Rickles, Hawe, and Shiell 2007). In such systems it is insufficient to model a few effects, but the many subunits must all be modelled. In these cases, models must be very complicated in order to be adequate descriptions.

This leads to the concept of simulation. When models become very complicated they often cannot be understood analytically or become so time consuming to specify and solve that they are impractical. However while humans alone may not be able to employ these complicated models we can use modern computer technology to gain insight from them. Simulation refers to the range of techniques for using a computer to gain insight from a model. Simulation is not an alternative to modelling, but it is an alternative way to use a model (Dubois 2018). Some common simulation techniques include numerical approximation and Monte-Carlo simulation.

Given the above it can be assumed that we will end up simulating our models, when dealing with complex systems. So if simulation is inevitable why not design models specifically to be simulated? Agent Based models are such models. Instead of mathematically describing the system's global properties Agent Based models allow the modeller to describe some relatively simple sub-units, called agents, and "grow" the global behaviour by simulating their interactions. In this way complicated macro-level behaviour can "emerge" from simple micro-level behaviour. This is significant to modellers because it only requires an understanding of the micro-level behaviour; the simulation reveals the macro level behaviour. One can imagine a process by which we start very small (perhaps at the atomic level). Then if simulation reveals that the macro-level (in this case molecular) behaviour is somewhat regular we can build a micro-description of it. In turn we can use this to simulate the next level of complexity (in this case cellular) and if we are fortunate we could continue this leap-frog procedure to gain a deeper understanding of our world.

2.2 Properties of Agent Based Models

Given Agent Based models do not include a mathematical description of a system's global behaviour, how are they characterised? Unfortunately the only complete characterisation of an Agent Based model is the computer code used to implement it. This can cause problems as the stylistic choices of the author, as well as the coding language, and even the machine on which a piece of code is run can all influence the model's behaviour. As well as this, with complicated pieces of code it can be a challenge to verify that the code is behaving as the author intended. This particular problem is addressed in Heard 2014.

To overcome some of the difficulties in communicating a model that exist only in code, prominent Agent Based modellers will usually use some conceptual description of the model. One of the most practical conceptual descriptions was illustrated in Epstein and Axtell's Sugarscape, (Epstein and Axtell 1996). The concept of the Sugarscape is that "Sugarpeople" live in the "Sugarscape", they collect resources, form relationships, trade and fight. In their book, Epstein and Axtell decomposed the model into three parts: the Agents, the Environment, and the Rules. Here this description is introduced using the Sugarscape model as an example.

Agents

Agents often represent people or businesses, in the Sugarscape model they are the Sugarpeople. Computationally an agent is a list of attributes. These attributes can be adaptive, or they can be fixed. In the Sugarscape model the Sugarpeople have some adaptive attributes, x and y positions, sugar stores, and some fixed attributes, metabolism, vision, and gender.

Environment

The environment represents the space in which agents interact. Two common examples of environments are a grid on which agents are positioned and a network of direct relationships between agents. A grid is useful for representing a physical landscape where a network is often used to model non-physical relationships. In the Sugarscape model the environment is a grid. The grid cells in the Sugarscape environment have an attribute called sugar level, so that the environment acts also as an agent. Environments often

play a dual role as the medium through which agents interact and also as an agent with attributes and its own interactions.

Rules

Rules represent the decisions and actions of agents. It is common for each rule to be associated with its own function in the computer code. An example of a rule in the Sugarscape model is the "Agent Movement Rule":

- Look out as far as vision permits in the four principle lattice directions and identify the unoccupied site(s) having the most sugar;
- if the greatest sugar value appears at multiple sites then select the nearest one;
- move to this site;
- collect all the sugar at this new position.

One may notice that, on a grid with multiple agents each enacting the above rule, the order in which agents take their turn will affect the outcome. As well as defining rules it is important to define the notions of time and order. Usually the notion of a turn or a time step is defined by an iteration in which every agent enacts all of their relevant rules. Within the time step, updating is often referred to as synchronous or asynchronous. In synchronous updating all agents act simultaneously, then their respective actions are resolved. Asynchronous updating proceeds by randomly ordering the agents and applying their rules sequentially.

For this thesis the "Agent, Environment, Rules" description will be sufficient. There has been some work on a more general description framework called the "Overview, Design concepts, and Details protocol (ODD)", a discussion of which can be found in Grimm et al. 2006 and Polhill et al. 2008.

Thus, we see that Agent Based models are special models designed to be simulated, however given the above discussion a natural question to ask is: "do Agent Based models need to be simulated?". To answer this question, we will demonstrate one way an Agent Based model can be described in terms of a more traditional mathematical modelling technique, Markov chains. (For a more detailed discussion of Agent Based models as Markov chains see Banisch 2015.)

A Markov chain is a discrete time stochastic process in which a point in the model's state space is sampled with probabilities that depend on the previous state. To characterise an Agent Based model as a Markov chain we must first establish the correct state space. If each agent in our Agent Based model has a state defined by the list of its attributes, then the state of the whole system at a given time can be specified by the list of the states of all the agents:

$$x_{i,t} = \text{state of agent } i \text{ at time } t$$

$$X_t = x_{1,t}, \ldots, x_{N,t} = \text{system state at time } t$$

Consequently the state space of the Agent Based model is the space of all possible values of X_t, and we denote it by Σ. To complete the Markov chain characterisation we need only specify the probability of transition between each of the points in the state space. If the Agent Based model's rules are written such that updates are conditional on only the current state, then the Markov chain transitional probabilities will also be conditional only on the current state. We denote the Markov chain transitional properties as $\hat{P}$. Given that the state space and the transitional properties are well defined, a Markov chain characterisation is also well defined, and we denote it as $(\Sigma, \hat{P})$.

While the above is technically possible for any Agent Based model, whose rules depend only on the current period, we note that Σ is usually very large. In a very simple Agent Based model, with 20 agents each with only five possible states, the state space has $5^{20} = 9.54 * 10^{13}$ possible points, with $\left(5^{20}\right)^2 = 9.09 * 10^{27}$ associate transitional probabilities. Consequently it is practically impossible to actually find the Markov chain characterisation of most Agent Based models. There are other attempts at finding more traditional characterisations of Agent Based models (see Laubenbacher et al. 2009), but the difficulty in the Markov chain case illustrates why in general Agent Based models must be simulated.

2.2.1 Agent Based Modelling and Bayesian Melding

Earlier we noted that the adoption of Agent based modelling has been slow due to the lack of techniques for parameter selection and error modelling. This problem occurs because traditional approaches such as Maximum Likelihood Estimation and Generalised Method of Moments estimation break down in the absence of a global description of the model,

i.e. in simulated contexts. However there are in fact a handful of lesser known techniques for modelling error in simulated models. It was pointed out in Grazzini and M. Richiardi 2015 that Approximate Bayes Computation, Method of Simulated Moments, and Indirect Inference could all be used in simulated settings. These approaches are similar in that they add an error modelling component to the simulated theoretical model resulting in a sort of simulated likelihood function. In this thesis we take a different approach, that of Bayesian melding. Bayesian melding uses the simulated theoretical model to construct a joint prior. Error modelling is then done in the usual way but is constrained by the simulated prior.

With Bayesian melding we can place a joint prior on theoretical quantities and use data to gain inference about their values jointly. This is illustrated below in Figure 2.1. Note that, while each data point will be connected to a particular model quantity, inference is gained jointly. An implication of this is that we are gaining inference on quantities that have no associated data. This process allows us to properly estimate parameter values, the problem that both Grazzini and M. Richiardi 2015 and Chen, Chang, and Du 2012 cited as a major limitation on the use of Agent Based model.

However there is a practical problem: most Agent Based models are stochastic and this is not addressed by standard Bayesian melding. Above we argued that a good model is not a complete description of a system but one that captures the properties of interest. Stochasticity is an invaluable tool for aggregating out uninteresting effects, allowing us to focus on what is important. For a simple example, take the rolling of a 6 sided die. A full deterministic model would require accounting for all the forces and geometry at play that lead to one side landing face up. However a very useful model is the stochastic model that assigns each side a 1/6 chance of occurring. Given the prevalence of stochastic effects in Agent Based modelling any true treatment of the subject should be robust to stochasticity. In the following chapter we show how one can account for a stochastic theoretical model within Bayesian melding, which should result in a more generally applicable technique.

Before proceeding to the technical details in chapter 3 we wish to draw attention to the fact that as we are combining work from the fields of applied mathematics and statistics, there are some potential linguistic pitfalls. Because statisticians are always dealing with data in a hands-on fashion, statistics draws a clear linguistic distinction between data and parameters. Data being observations of the real world and parameters being unobserved

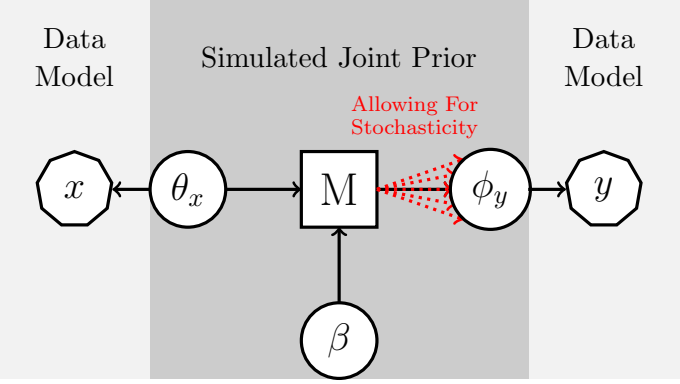

Figure 2.1: Bayesian Melding Schematic

quantities whose values we are trying to infer. In contrast, applied mathematics tends to deal with data in a hands-off way, using it for inspiration or calibration. As such a statistician would consider all the quantities in a mathematical model to be parameters as they are not data. However, especially in simulated models a distinction is made between the following: inputs - quantities that are fed into the mathematical model, outputs - quantities that result from the model, and parameters - quantities that are specified before the model is run. We keep with the applied mathematics tradition of distinguishing between inputs, outputs and parameters. However, the reader would do well to keep in mind that in a statistical sense, what we are referring to as "inputs", "outputs" and "parameters" are all parameters whose values we are attempting to infer.

3. Bayesian Melding and Stochastic Extension

In this chapter Bayesian melding will be introduced by outlining the technique for application to deterministic models, as proposed in Poole and Raftery 2000. The original Bayesian melding method will then be extended for application to stochastic models. To this end the process of Bayesian inference will be divided into three components: constructing priors, defining likelihoods and updating beliefs. These steps will be recapitulated and we discuss how they are adapted to allow a theoretical model to be incorporated into the inferential process.

Constructing Priors

The first step in Bayesian inference is to define prior beliefs about the unknown quantities. In the multivariate case this is a joint probability distribution which includes not only information about the values of each quantity but also the relationships between quantities.

Defining Likelihoods

Often data is not perfectly informative about the values of the quantities under inference. Likelihoods are probability distributions that model the randomness of the data generating process so that data can inform beliefs appropriately.

Updating Beliefs

Given likelihoods, prior beliefs can be updated to incorporate information from data according to Bayes' rule. This results in a joint posterior distribution on unknown quantities. Often the posterior distribution does not have a closed form solution, in which case a sampling algorithm can be developed so that properties of the posterior distribution can be approximated empirically.

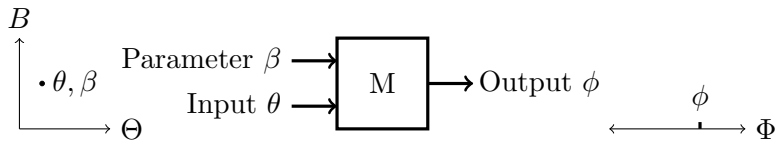

Figure 3.1: Generic Description of Deterministic Mathematical Models

3.1 Background - Bayesian Melding for Deterministic Models

With an understanding of Bayesian inference the reader needs only a minimal exposure to deterministic models to proceed. To understand deterministic models it is useful to have a generic way of describing mathematical models. Such a description is built upon three types of quantity: inputs θ, parameters β and outputs ϕ as described in chapter 2. We denote the mathematical relationship between these quantities by $M()$. Note that the distinction between inputs and parameters is that inputs are directly related to data, where parameters are not. Thus, in the deterministic case:

$$M : \Theta \times B \to \Phi, \quad \phi = M(\theta, \beta), \quad \theta \in \Theta, \quad \beta \in B, \quad \phi \in \Phi$$

Deterministic models have the property that each point in the input-parameter space is mapped to a single point in the output space. It may however be the case that multiple points in the input-parameter space map to the same point in the output space. In functional terms we say that deterministic models can be one-to-one or many-to-one but never one-to-many or many-to-many. This is illustrated in Figure(3.1).

3.1.1 Constructing Priors

In multivariate Bayesian inference the joint prior represents beliefs about not only the values of inputs and outputs but their relationships. The challenge addressed by Bayesian melding is to construct a prior that is constrained by a theoretical model. An initial attempt at constructing such a prior, called Bayesian synthesis, was made in Raftery, Givens, and Zeh 1995. In Bayesian synthesis a joint prior on θ, β and ϕ, that contained all non-model prior information, was defined. The value of the prior was then adjusted to zero outside of the subset where the model holds $\{\theta, \beta, \phi : M(\theta, \beta) = \phi\} \subset \Theta \times B \times \Phi$.

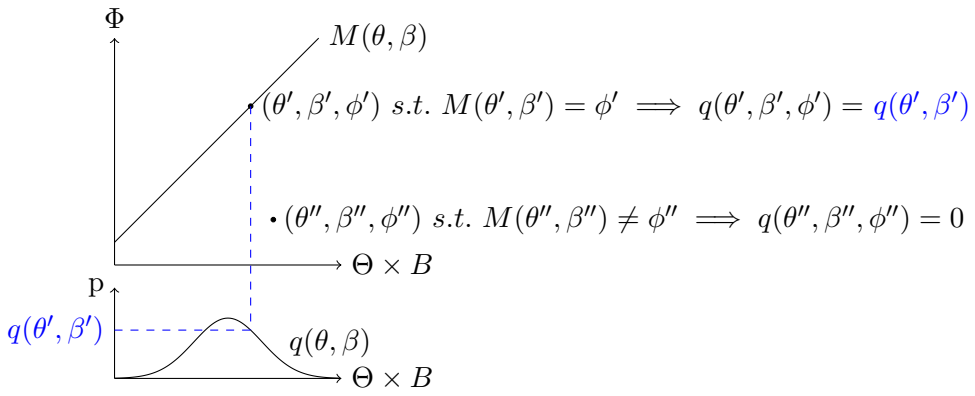

Figure 3.2: Joint Prior $q(\theta, \beta, \phi)$ implied by Input-Parameter Prior $q(\theta, \beta)$ and Model $M(\theta, \beta)$

Probabilities at points inside the subset were normalised to ensure the resulting probability density function was a proper probability distribution. This technique guaranteed zero probability at all points where the model does not hold, but as pointed out in Wolpert 1995, this alone does not lead to a well defined set of prior beliefs.

The failure of Bayesian synthesis leads to a key realisation, illustrated in Figure(3.2). If prior beliefs about input-parameters are denoted as $q(\theta, \beta)$ then the following joint prior is implied by the theoretical model:

$$\text{“Induced” joint prior} = q^*(\theta, \beta, \phi) = q(\theta, \beta)\mathbb{1}[M(\theta, \beta) = \phi]$$

Bayesian synthesis fails because the original joint prior is generally in disagreement with the joint prior implied by the model. A corollary of this is that any joint prior can be completely specified by a prior on input-parameters alone. With Bayesian melding, Poole and Raftery developed a technique that adequately incorporated this relationship.

In Bayesian melding one starts with separate priors on input-parameters and outputs, that capture all non-model information. These are called the “pre-model” priors, and are denoted by:

$$\text{“Pre-model” prior on input-parameters} = q^{1p}_{[\theta,\beta]}(\theta, \beta)$$

$$\text{“Pre-model” prior on outputs} = q^{2}_{[\phi]}(\phi)$$

Priors on input-parameters and outputs are defined separately, based on beliefs about their true values in the absence of the theoretical model. As such, it will almost always be the case that the joint prior, defined by these independent marginal priors, is inconsistent

with the theoretical model. More specifically, the pre-model prior on outputs and the "induced prior" on outputs will disagree:

$$\begin{array}{ccc} \text{"Induced" prior on outputs} & \neq & \text{"Pre-model" prior on outputs} \\ q^{1p}_{[\phi]}(\phi) = \int\limits_{\theta,\beta:M(\theta,\beta)=\phi} q^{1p}_{[\theta,\beta]}(\theta,\beta) & \neq & q^{2}_{[\phi]}(\phi) \end{array}$$

The disagreement between these two sets of priors needs to be resolved. Poole and Raftery proposed reconciling the disagreement by logarithmically "melding" the pre-model prior on outputs and the induced prior on outputs, resulting in a melded prior on outputs, denoted:

$$\text{"Melded" prior on outputs} = \tilde{q}_{[\phi]}(\phi) \propto \left[q^{1p}_{[\phi]}(\phi)\right]^{\alpha} \left[q^{2}_{[\phi]}(\phi)\right]^{1-\alpha}$$

The parameter α tunes the "melding" to consider only pre-model information about input-parameters ($\alpha = 1$), only pre-model information about outputs ($\alpha = 0$), or some combination ($0 < \alpha < 1$). Often α is assumed to be 0.5, indicating that pre-model information about input-parameters and pre-model information about outputs are considered to be equally reliable.

Together the melded prior on outputs and the model constitute the desired joint prior. However in many cases (including the Agent Based models that will be discussed in the next chapter) we will also need the associated marginal prior on input-parameters. We will call the problem of obtaining the melded prior on input-parameters from the model and the melded prior on outputs, the inversion problem, characterised by:

Inversion Problem - Deterministic Case

$$\text{Find} \quad \tilde{q}_{[\theta,\beta]}(\theta,\beta) \quad \text{s.t.} \int\limits_{\theta,\beta:M(\theta,\beta)=\phi} \tilde{q}_{[\theta,\beta]}(\theta,\beta) = \tilde{q}_{[\phi]}(\phi)$$

That deterministic models are either one-to-one or many-to-one mappings makes solving this system much easier. Poole and Raftery 2000 showed that for deterministic models that map one-to-one, the above system is solved by the input-parameter prior below:

Solution - One-to-one case

$$\text{"Melded" prior on input-parameters} = \tilde{q}_{[\theta,\beta]}(\theta,\beta) = \tilde{q}_{[\phi]}(M(\theta,\beta))$$

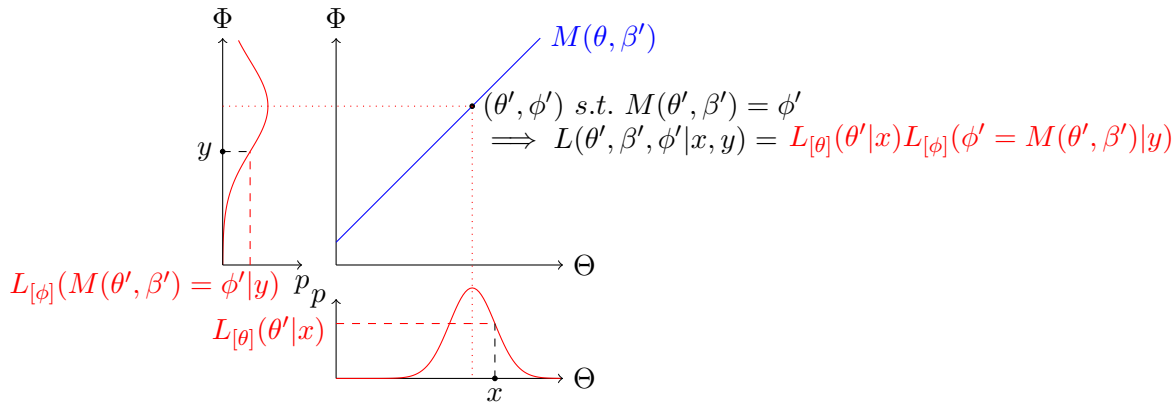

Figure 3.3: Multivariate Likelihood expressed in terms of Input-Parameters

The intuition here is that the probability associated with each point in the output space should be allocated to the point in the input-parameter space that maps to it. This intuition extends to the many-to-one case, but the probability is now allocated to many points. Poole and Raftery 2000 showed that any allocation of the output probability across the associated points in the input-parameter space will satisfy the above condition. They proposed that the allocation which results in the melded input-parameter prior that is most similar to the pre-model input-parameter prior, in terms of the Kullback-Leibler divergence, should be thus chosen:

Solution - Many-to-one case

$$\text{“Melded” prior on input-parameters} = \tilde{q}_{[\theta,\beta]}(\theta, \beta) = \tilde{q}_{[\phi]}(M(\theta, \beta)) \left[\frac{q_1(\theta, \beta)}{q_1^*(M(\theta, \beta))} \right]$$

On the existence of solutions

Deterministic theoretical models map each point in the input-parameter space to a single point in the output space. A corollary of this is that the input-parameter space can be partitioned into subsets based on which point in the output space the theoretical model assigns. The probability associated with a point in the output space can then be “inverse mapped” to the relevant subset in the input-parameter space. This “inverse mapping” guarantees that, given the melded-prior on outputs is a proper probability distribution, there exists a prior on the input-parameter space that solves the inversion problem.

3.1.2 Defining Likelihoods

As was the case for priors, likelihoods in the Bayesian melding setting can be expressed in terms of only input-parameters and the model. Furthermore if we assume independence between the randomness in the data generating process for inputs and outputs this likelihood can be decomposed into an input component and an output component:

$$L(\theta, \beta|x, y) = L(\theta, \beta|x)L(\theta, \beta|y)$$

The randomness in the data associated with inputs is assumed to be independent of parameter values, consequently the input component of the likelihood can be expressed as:

$$L(\theta, \beta|x) = L_{[\theta]}(\theta|x), \text{ where } x \text{ is data related to inputs}$$

The output component of the likelihood is slightly more involved as the data must be parsed through the theoretical model. This can be achieved by the following:

1. Define a likelihood on outputs in the usual way, $L(\phi|y)$
2. For a given point in input-parameter space evaluate the model to find the associated output $\phi' = M(\theta', \beta')$
3. The output component of likelihood at (θ', β') is $L(\phi'|y)$

or more concisely:

$$L(\theta, \beta|y) = L_{[\phi]}(M(\theta, \beta)|y), \text{ where } y \text{ is data related to outputs}$$

The resulting joint likelihood is expressed as:

$$L(\theta, \beta|x, y) = L_{[\theta]}(\theta)L_{[\phi]}(M(\theta, \beta))$$

This can be substituted into Bayes' rule, with the melded prior, giving a proportional formula for the posterior beliefs:

$$\text{Posterior on Input-Parameters} = \pi_{[\theta,\beta]}(\theta, \beta) \propto \tilde{q}_{[\theta,\beta]}(\theta, \beta)L_{[\theta]}(\theta)L_{[\phi]}(M(\theta, \beta))$$

3.1.3 Updating Beliefs

As is often the case with Bayesian techniques, the posterior distribution will not be a known parametric distribution. Consequently, to gain inference from the posterior distribution a

sampling algorithm is required. Poole and Raftery 2000 shows that the posterior can be expressed as the pre-model prior on input-parameters multiplied by a weight, which leads naturally to a sampling-importance-resampling (SIR) algorithm. The weight is found by expanding the melded prior on input-parameters into its component parts:

$$\pi_{[\theta,\beta]}(\theta,\beta) \propto \left[q^{1p}_{[\phi]}(M(\theta,\beta))\right]^{\alpha} \left[q^{2}_{[\phi]}(M(\theta,\beta))\right]^{1-\alpha} \left[\frac{q^{1p}_{[\theta,\beta]}(\theta,\beta)}{q^{1p}_{[\phi]}(M(\theta,\beta))}\right] L_{[\theta]}(\theta) L_{[\phi]}(M(\theta,\beta))$$

This can be rearranged and expressed as the pre-model prior on input-parameters multiplied by a weight:

$$\pi_{[\theta,\beta]}(\theta,\beta) \propto q^{1p}_{[\theta,\beta]}(\theta,\beta) \left[\frac{q^{2}_{[\phi]}(M(\theta,\beta))}{q^{1p}_{[\phi]}(M(\theta,\beta))}\right]^{1-\alpha} L_{[\theta]}(\theta) L_{[\phi]}(M(\theta,\beta))$$

This formula motivates the SIR algorithm put forward in Poole and Raftery 2000:

1. Draw a sample of size K from $q^{1p}_{[\theta,\beta]}(\theta,\beta)$

2. Apply the model to each of the K to obtain $(M(\theta_1,\beta_1)=\phi_1,...,M(\theta_K,\beta_K)=\phi_K)$

3. Apply non-parametric density estimation to $(\phi_1,...,\phi_K)$ to obtain an estimate of $q^{1p}_{[\phi]}(\phi)$

4. Using the density obtained in step 3, calculate the importance sampling weights

$$w_k = \left[\frac{q^{2}_{[\phi]}(M(\theta_k,\beta_k))}{q^{1p}_{[\phi]}(M(\theta_k,\beta_k))}\right]^{1-\alpha} L_{[\theta]}(\theta_k) L_{[\phi]}(M(\theta_k,\beta_k))$$

5. Draw L resamples from $((\theta_1,\beta_1),...,(\theta_K,\beta_K))$ with the probability of sampling (θ_k,β_k) proportional to w_k

3.2 Extension - Bayesian Melding for Stochastic Simulation Models

Stochastic models are like deterministic models in that inputs and parameters are fed into stochastic models, but where deterministic models map to a single point, stochastic models produce a probability distribution over the output space. Using a generic description,

stochastic models can be characterised by:

$$M(\theta, \beta) = P(\phi|\theta, \beta) \;\; \forall \phi \in \Phi, \quad \text{where: } \theta \in \Theta, \;\; \beta \in B, \phi \in \Phi$$

As illustrated in Figure(3.4) stochastic models map from a point in the input-parameter space, to a point in the space of probability densities on the output space. Note that the notions of one-to-one/many-to-one, that allowed for the straightforward handling of the inversion problem in the deterministic case, do not apply in the stochastic case.

3.2.1 Constructing Priors

In the stochastic case, we propose a procedure for the construction of model consistent priors by drawing on the deterministic case. As in the deterministic case, the process begins by defining independent priors on input-parameters and outputs:

$$\text{“Pre-model” prior on input-parameters} = q^{1p}_{[\theta,\beta]}(\theta, \beta)$$

$$\text{“Pre-model” prior on outputs} = q^{2}_{[\phi]}(\phi)$$

Again, the joint prior defined by these “pre-model” priors will likely be inconsistent with the theoretical model. The specific problem is the disagreement between the pre-model prior on outputs and the induced prior on outputs:

$$\begin{array}{rcl} & \text{“Induced” prior on outputs} & \neq \quad \text{“Pre-model” prior on outputs} \\ q^{1p}_{[\phi]}(\phi) = \int\limits_{\theta', \beta' \in \Theta \times B} q^{1p}_{[\theta,\beta]}(\theta', \beta') P(\phi|\theta', \beta') & \neq & q^{2}_{[\phi]}(\phi) \end{array}$$

The disagreement between the induced and pre-model priors on outputs is reconciled with logarithmic melding as in the deterministic case:

$$\text{“Melded” prior on outputs} = \tilde{q}_{[\phi]}(\phi) \propto \left[q^{1p}_{[\phi]}(\phi)\right]^{\alpha} \left[q^{2}_{[\phi]}(\phi)\right]^{1-\alpha}$$

Again this melded prior is a compromise between our beliefs about input-parameters and outputs, given the constraint of the theoretical model. However in most cases to proceed we will need an evaluation of the associated input prior, leading to the stochastic case of the inversion problem:

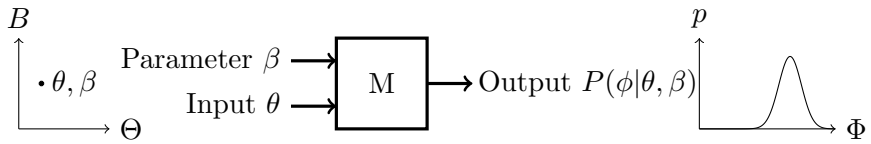

Figure 3.4: Generic Description of Stochastic Mathematical Models

Inversion Problem - Stochastic Case

$$\text{Find} \quad \tilde{q}_{[\theta,\beta]}(\theta,\beta) \quad \text{s.t.} \quad \tilde{q}_{[\phi]}(\phi) = \int\limits_{\theta',\beta' \in \Theta \times B} \tilde{q}_{[\theta,\beta]}(\theta',\beta') P(\phi|\theta',\beta')$$

On the existence of solutions

Because stochastic models are not many-to-one/one-to-one mappings from the input-parameter space to the output space, there is no "inverse mapping" analogous to what was used to solve the system in the deterministic case. As such it is much harder to make statements about the existence of solutions. Whether or not a solution exists will of course depend on the properties of the melded output prior and the specific stochastic model.

Agent Based models, the models that have been chosen to demonstrate these ideas, are computer simulations with no closed functional forms. When extending Bayesian melding to Agent Based models, the form of $P(\phi|\theta, \beta)$ will not be known. As a result the above system cannot be solved whether a solution exists or not. Later we will propose the "emulator approximation" method to address these problems, but first we will discuss how defining likelihoods and updating beliefs would proceed if a suitable joint prior could be found. For this discussion we denote this hypothetical prior on the input-parameter space by $\tilde{q}_{[\theta,\beta]}(\theta, \beta)$.

3.2.2 Defining Likelihoods

Defining likelihoods in the stochastic case is similar to the deterministic case, but the method for linking data related to outputs with input-parameters must be revised. Again it appears reasonable to assume that the random elements of gathering data on inputs and outputs are independent. Independence allows the joint likelihood to be broken into two separate terms:

$L(\theta, \beta|x, y) = L(\theta, \beta|x)L(\theta, \beta|y)$ where x is data related to inputs, y is data related to outputs

The likelihood for data related to inputs is unchanged:

$$L(\theta, \beta|x) = L_{[\theta]}(\theta|x)$$

As in the deterministic case the output data is related to input-parameters indirectly through the theoretical model. Again a likelihood function is defined to capture how likely it was to observe the data, given a value for output. However in the stochastic case, a weighted sum of the value of the likelihood function is taken. The weight captures how likely each output is to occur given an input-parameter value:

$$L(\theta, \beta|y) = \int_{\phi \in \Phi} L_{[\phi]}(\phi|y) P(\phi|\theta, \beta) d\phi, \text{ where } y \text{ is data related to outputs}$$

The resulting posterior is now:

$$\text{Posterior on input-parameters} = \pi_{[\theta,\beta]}(\theta, \beta) \propto \tilde{q}_{[\theta,\beta]}(\theta, \beta) L_{[\theta]}(\theta) \int_{\phi \in \Phi} L_{[\phi]}(\phi|y) P(\phi|\theta, \beta) d\phi$$

3.2.3 Updating Beliefs

As the posterior will generally not be a known parametric distribution, a sampling algorithm is devised to generate an approximate empirical distribution. The SIR sampling algorithm developed in Poole and Raftery 2000 for the deterministic case is used as a guide. As in the deterministic case, samples are taken from the pre-model prior on input-parameters and re-sampled. In the stochastic case the weights are given by:

$$\textit{Stochastic SIR weights}: \qquad \frac{\tilde{q}_{[\theta,\beta]}(\theta, \beta)}{q^{1p}_{[\theta,\beta]}(\theta, \beta)} L_{[\theta]}(\theta) \int_{\phi \in \Phi} L_{[\phi]}(\phi|y) P(\phi|\theta, \beta) d\phi$$

Upon inspection of the sampling weight we note it can be divided into two parts. The first term $\frac{\tilde{q}_{[\theta,\beta]}(\theta,\beta)}{q^{1p}_{[\theta,\beta]}(\theta,\beta)}$ re-weights samples from the pre-model prior to be samples from the melded prior on input-parameters. The second term $L_{[\theta]}(\theta) \int_{\phi \in \Phi} L_{[\phi]}(\phi|y) P(\phi|\theta, \beta) d\phi$ updates these prior samples to give samples from the posterior. Dividing the sampling into these two sections provides empirical approximations of both the melded prior and the posterior. This is useful as the empirical approximation of the melded prior can be used for checking our solution to the "inversion problem".

As this technique relies on numeric evaluation, the integral in the likelihood is a

concern. Ševčíková, Raftery, and Waddell 2007 address this problem by evaluating the theoretical model multiple times for each sample of input-parameters and averaging the likelihood at these points. Given a particular value of input-parameters a single model evaluation can be thought of as a draw from $P(\phi|\theta, \beta)$. In this way the procedure proposed by Ševčíková, Raftery, and Waddell 2007 can be thought of as taking a Monte Carlo approximation of the integral. We replicate the Ševčíková, Raftery, and Waddell 2007 procedure in this thesis, but as future work the adequacy of this approximation should be investigated.

Proposed Sampling Algorithm for Stochastic Bayesian Melding

1. Sample $((\theta_1, \beta_1), ..., (\theta_K, \beta_K))$ from $q^{1p}_{\theta,\beta}(\theta, \beta)$

2. Apply the theoretical model B times to each sample to obtain $(\phi_{11}, ..., \phi_{1B}, \cdots, \phi_{K1}, ..., \phi_{KB})$

3. Calculate a re-sample weight for each sample (θ_k, β_k)

$$w_k = \frac{\tilde{q}_{[\theta,\beta]}(\theta_k, \beta_k)}{q^{1p}_{[\theta,\beta]}(\theta_k, \beta_k)} L_{[\theta]}(\theta_k) \frac{1}{B} \sum_{b=1}^{B} L_{[\phi]}(\phi_{kb}|y)$$

4. Draw L resamples from $((\theta_1, \beta_1), ..., (\theta_K, \beta_K))$ with the probability of sampling (θ_k, β_k) proportional to w_k

The above will sample from the marginal posterior on input-parameters $\tilde{\pi}_{[\theta,\beta]}(\theta, \beta)$. To obtain a sample from the joint posterior $\tilde{\pi}_{[\theta,\beta,\phi]}(\theta, \beta, \phi)$, input-parameter samples can be fed back into the theoretical model. If the model has a long run time, the computational burden can be reduced by keeping track of the multiple model runs already obtained, in step 2, for each sample. That is store $(\phi_{i1}, ..., \phi_{iK})$ along with (θ_i, β_i). This way, instead of rerunning the model, a random number k is selected from $\{1, ..., K\}$ and the relevant ϕ_k is drawn.

3.3 Approximate Melded Prior and Emulator Methods

Earlier it was explained that the original Bayesian melding protocol for constructing a joint prior is problematic in the stochastic case. While a suitable melded prior on outputs can be found, we were unable to solve the inversion problem. A simple method for

avoiding this problem is to only place a prior on the input-parameter space, that is to set α to 1. Ševčíková, Raftery, and Waddell 2007 implements this method and reduces the problem further by only incorporating data on output quantities. Making these adjustments is convenient and may be the correct choice in certain situations, but it is not fully capitalising on the Bayesian melding framework. We contend that a full extension of Bayesian melding should allow for α to take any value in the interval $[0, 1]$ (as in the deterministic case) and not be arbitrarily set to 1. In what follows we present a method for finding a prior on the input-parameter space that approximately solves the inversion problem, without constraining α. This procedure is rather involved, and we acknowledge that in practice modellers may still choose to set α to 1 for convenience. However, we believe that making the full framework available to modellers allows them to make more informed decisions.

3.3.1 Proposed Method: Emulator Approximation

The existence of solutions to the inversion problem depends on the form of both the melded prior on outputs and the theoretical model. In the case of Agent Based modelling the form of the theoretical model is unspecified. Instead, information is drawn from model evaluations. To find an approximate solution to the inverse problem we propose a two step approach: First the functional form of the theoretical model must be approximated using model evaluations. This process is referred to as *emulation*. Hooten et al. 2011 and Heard 2014 demonstrate the use of emulators in the context of Agent Based modelling. Second, depending on the type of emulator, the melded prior on outputs is approximated such that the resulting system (composed of the emulated theoretical model and the approximate melded output) is both analogous to the original system and has sufficient structure for solvability. In this thesis theoretical models are emulated by a matrix and the melded prior on outputs is approximated by a vector, resulting in a linear system of equations. In the future we seek to investigate different types of emulators, but here a linear system of equations was found to produce satisfactory results. Because of the use of emulators this method will be referred to as the "Emulator Approximation" method. Why is the resulting system analogous to the original system? The reason is that the linear system can be regarded as a discrete approximation of the true system. Input-parameter and output spaces are bounded and divided into discrete grids.

Emulator Approximation Method (Linear System of Equations)

Approximate

$$\tilde{q}_{[\phi]}(\phi) = \int_{\theta,\beta \in \Theta \times B} \tilde{q}_{[\theta,\beta]}(\theta,\beta) P(\phi|\theta,\beta)$$

with

$$\begin{bmatrix} \tilde{q}_{[\phi]}(\phi_1) \\ \vdots \\ \tilde{q}_{[\phi]}(\phi_J) \end{bmatrix} = \begin{bmatrix} P(\phi_1|(\theta,\beta)_1) & \cdots & P(\phi_1|(\theta,\beta)_I) \\ \vdots & \ddots & \vdots \\ P(\phi_J|(\theta,\beta)_1) & \cdots & P(\phi_J|(\theta,\beta)_I) \end{bmatrix} \begin{bmatrix} \tilde{q}_{[\theta,\beta]}((\theta,\beta)_1) \\ \vdots \\ \tilde{q}_{[\theta,\beta]}((\theta,\beta)_I) \end{bmatrix}$$

where

$$\tilde{q}_{[\phi]}(\phi_j) = \int_{\text{grid cell } j} \tilde{q}_{[\phi]}(\phi) d\phi$$

$$P(\phi_j|(\theta,\beta)_i) = \frac{\text{no. samples in input-parameter grid cell } i \text{ mapped to output grid cell } j}{\text{total no. samples input-parameter grid cell } i}$$

Solving the linear system results in a discrete probability distribution over the input-parameter space. To restore continuity of the probability density function, the probability associated with each grid cell is assigned to its centre, and a continuous surface is found by interpolation. The accuracy of the above method is subject to both the number of model runs and the grid resolution. In practice the number of model runs is limited so the size of the grid cells is chosen to ensure a sufficient ratio of model runs to grid cells. This method was tested for two example models, below, to demonstrate the quality of the approximation. The first is a simple stochastic model, and the second is a deterministic example from Poole and Raftery 2000. There is no data attached to these examples. Thus there is no distinction between an input and a parameter.

3.3.2 Example: Stochastic Model

The emulator approximation method is demonstrated on the following simple stochastic model:

$$\phi = s\ \theta, \quad s \sim u(0,1)$$

With priors:

$$\theta \sim \text{Beta}(2,2), \quad q^1_{[\theta]}(\theta) = \frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{\frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha\beta)}} \quad \alpha = 2,\ \beta = 2, \quad \text{for} \quad 0 < \theta < 1$$

$$\phi \sim \text{Beta}(2,5), \quad q^2_{[\phi]}(\phi) = \frac{\phi^{\alpha-1}(1-\phi)^{\beta-1}}{\frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha\beta)}} \quad \alpha = 2,\ \beta = 5, \quad \text{for} \quad 0 < \phi < 1$$

The melded prior on outputs is found in the usual way. The input and output spaces are discretized into uniformly separated grids, each with 10 cells. The melded prior on outputs is discretized as shown above. Integrals across the grid cells are approximated by summation. The subsequent discrete probability density function is normalised to sum to one.

The theoretical model is emulated, reusing the model runs that were used to approximate the induced prior on outputs. The resulting matrix equation is then solved from which we obtain a discrete approximate melded prior on inputs. Cubic smoothing splines, as implemented in the R function smooth.spline, are applied as a form of interpolation and the result is then subjected to normalisation.

To evaluate the quality of the approximate solution multiple samples are obtained using the SIR algorithm discussed in section (3.2.3). The theoretical model is then applied to induce output samples. In Figure(3.5) the red line shows the desired melded prior on outputs. Comparing the sample from the approximate melded prior against the desired melded prior, there seems to be reasonable agreement between the two. One concerning feature is that there is no dip in probability at values close to zero in the approximate prior. It is likely that a finer grid may improve the result, but the process is limited by the number of samples needed per grid square. Also note that there may be concerns scaling the model for higher dimension applications.

3.3.3 Example: Deterministic Model

The second example used to demonstrate the emulator approximation is a deterministic model taken from Poole and Raftery 2000. As deterministic models are a special case of stochastic models it is important that the emulator approximation approach performs as intended. This model has two input quantities, making it higher dimensional than the

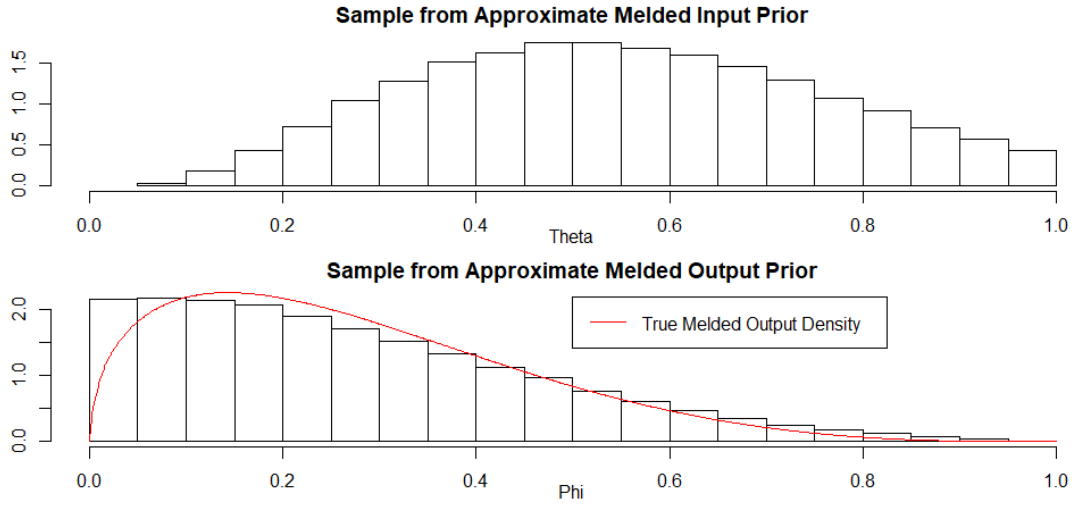

Figure 3.5: Stochastic Example - Emulator Approximation Method

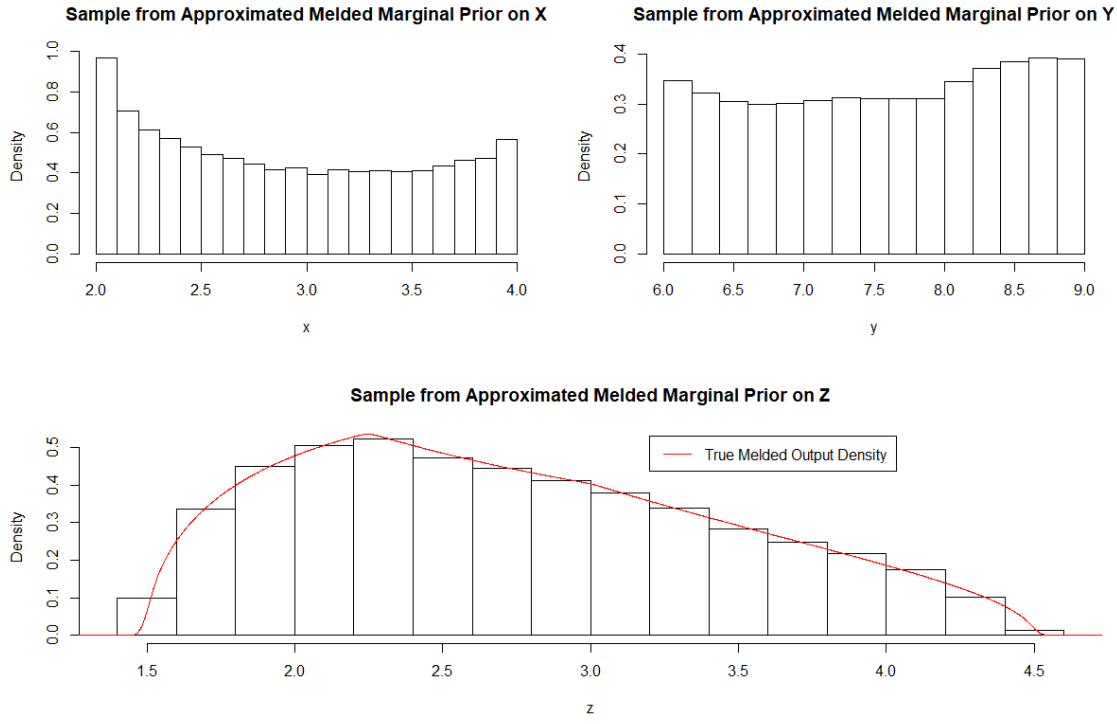

Figure 3.6: Deterministic Example - Emulator Approximation Method

previous stochastic model:

$$Z = \frac{Y}{X}$$

$$q_1(x, y) = 1/6, \quad \text{for} \quad 2 < x < 4, \quad 6 < y < 9$$

$$q_2(z) = 1/5, \quad \text{for} \quad 0 < z < 5$$

The melded output prior was found in the usual way. The input space and output spaces were each discretized into grids of 100 evenly sized cells. In this case the matrix used to approximate the theoretical model was found to be singular so a Moore-Penrose generalised inverse was used to solve the linear system. The smoothing/interpolation technique used was polynomial linear regression as implemented in the R function lm. Again samples from the approximate melded priors on inputs and outputs are obtained. The marginal distributions of X and Y are shown in Figure(3.6), with the resulting marginal prior on output. Again the red line represents the desired melded prior on outputs. Here the approximate distribution is similar to the true distribution. This further demonstrates that approximate solutions to the inverse problem can capture the desired prior information.

3.4 Bayesian Melding Procedure

Here steps are listed for the extended Bayesian melding technique. This reference material will be employed in chapter four:

1. Define a theoretical model that describes the system of interest
2. Decide which quantities are in inputs, parameters and outputs
3. Collect data related to the inputs and outputs of the system
4. Define likelihoods $L_{[\theta]}(\theta|x),\ L_{[\phi]}(\phi|y)$ that link data with inputs and outputs
5. Define marginal prior beliefs $q^{1p}_{[\theta,\beta]}$, $q^{2}_{[\phi]}$ about the values of inputs, parameters and outputs
6. Construct a joint prior across inputs, parameters and outputs that is consistent with the theoretical model and the marginal prior beliefs:
   (a) Draw a sample $((\theta_1, \beta_1)..., (\theta_K, \beta_K))$ from $q^{1p}_{[\theta,\beta]}$

(b) Apply the model B times to each (θ_k, β_k) to obtain $(\phi_{11}, ..., \phi_{1B}, \cdots, \phi_{K1}, ..., \phi_{KB})$

(c) Randomly select one of the B model runs associated with each k to obtain $(\phi_{1'}, ..., \phi_{K'})$, then apply non-parametric density estimation to obtain $q^{1p}_{[\phi]}(\phi)$

(d) Logarithmically meld $q^{1p}_{[\phi]}(\phi)$ and $q^{2}_{[\phi]}(\phi)$ to obtain the desired melded prior on outputs $\tilde{q}_{[\phi]}(\phi)$

(e) Use the emulator approximation method to obtain $\hat{\tilde{q}}_{[\theta,\beta]}(\theta, \beta)$:

i. Decide on an appropriate grid to discretize the input-parameter and output spaces; the number of cells will depend on the available number of model runs

ii. Obtain a discrete approximation of the desired melded prior on outputs

iii. Use the samples from step 6(b)

$$(\theta_1, \beta_1, \phi_{11}, ..., \theta_1, \beta_1, \phi_{1B}, \cdots, \theta_K, \beta_K, \phi_{K1}, ..., \theta_K, \beta_K, \phi_{KB})$$

to approximate the theoretical model as a matrix

iv. Solve the resulting linear system for a discrete approximation of the desired melded prior on input-parameters

v. Use smoothing or interpolation to obtain a continuous approximation of the desired melded prior on input-parameters

vi. Return a smoothed and normalised function for $\hat{\tilde{q}}_{[\theta,\beta]}(\theta, \beta)$

7. Use the SIR algorithm to obtain samples from and perform diagnostic checks on the approximate melded prior:

(a) For each sample from step6(a) calculate the following resample weight

$$w_k = \frac{\hat{\tilde{q}}_{[\theta,\beta]}(\theta_k, \beta_k)}{q^{1p}_{[\theta,\beta]}(\theta_k, \beta_k)}$$

(b) Draw L samples from $((\theta_1, \beta_1)..., (\theta_K, \beta_K))$ with probability proportional to $(w_1, ..., w_K)$. Accompany the draw with the B associated model runs $(\phi_{k1}, ..., \phi_{kB})$

(c) For each of the L samples randomly select one of the B model runs. The resulting L values of ϕ are a sample from the approximate melded output prior

(d) If desired, repeat steps 6(e) to 7(c) using different choices of grid, interpolation method, etc. until the approximate melded output prior is sufficiently similar to the desired melded output prior

8. Use the SIR algorithm to sample from the posterior beliefs:

(a) For the input-parameters $((\theta_1, \beta_1), \cdots, (\theta_L, \beta_L))$ from step 7 calculate the following re-sample weights

$$w_l = L_{[\theta]}(\theta_l) \frac{1}{B} \sum_{b=1}^{B} L_{[\phi]}(\phi_{lb}|y)$$

(b) Draw a value from $((\theta_1, \beta_1)..., (\theta_L, \beta_L))$ with probabilities proportional to $(w_1, ..., w_L)$

(c) Randomly select an associated model run from $(\phi_{l1}, ..., \phi_{lK})$ to accompany the above draw

(d) Repeat 8(b-c) until S samples have been taken from the joint posterior

4. Application - Social Learning in Bumblebees

In this chapter, our extended Bayesian melding approach is applied to an Agent Based model. The specific model was originally proposed in Smolla, Gilman, et al. 2015 and it describes the foraging behaviour of bumblebees. Each step of the procedure outlined in chapter two will be discussed with respect to the bumblebee model. The application is then extended by applying Bayesian melding with new likelihoods that further decompose error, based on colony effects.

4.1 Application of Bayesian Melding (Parameter Specification)

4.1.1 Step 1 - Defining the System of Interest and the Agent Based Model

As we spoke about in chapter 2, modelling involves identifying some process in the real world which we call our "system" and developing a mathematical description of it, which we call the "model". In this thesis we will be using an external model and data set, so these decisions have largely been made for us. Here, in step 1, we will describe the model and later, in step three, we will outline the real world system from which our experimental data were obtained. The reader may note that many of the details of the model and are not present in the system, in fact the connection occurs at a fairly abstract level. This is somewhat problematic as will become apparent in step four when we attempt to develop likelihoods that link the experimental data to the model. While these problems may reduce the scientific validity of the results as they pertain to biology, addressing this formally is beyond the scope of this thesis which is focused on methodology.

Environment

The environment consists of N flower patches that bees can visit. Each has some level of resources that can be collected. Bees do not take a position in the environment, instead on a round by round basis they may select a patch to visit that is unrelated to their previous location. That is to say the environment has no geometric structure or spatial relationships.

Agents

There are M agents in this model each representing a Bumblebee. Each agent has $N + 3$ internal attributes. Learning type, a fixed binary attribute, is either social or individualistic and determines the strategy employed by the bee. Resources collected is a variable attribute which tracks all the resources a bee has collected over its life. Age is a variable attribute which indicates the number of time steps for which a bee has been alive. Belief, a vector valued attribute of length N, stores what the bee believes the payoff to be at each of the N flower patches.

Rules

The model is run with 6 functions...

Initialise Patches

N values are sampled from a gamma distribution with some shape and rate parameters, S and R. These are the resources available at each of the N patches in the model.

Initialise Bees

A data frame with M rows is generated each row representing a bee. Half the bees are assigned learning type 0 corresponding to individual learning, and half are assigned learning type 1 corresponding to social learning. All bees are given an age of 0, have collected 0 and have no knowledge about the resources at each flower patch.

Update Patches

For each patch a draw is made from a Bernoulli distribution with probability of success τ.

A failure results in keeping the same resources as last round, a success results in resources being re-sampled from the gamma distribution with shape S and rate R.

Update Bees

For each bee a draw is made from a Bernoulli distribution with probability of success β. Bees with value 1 are allocated to "exploit" this round while bees with value 0 are allocated to "explore". Additionally, in our implementation, bees with no knowledge of any patch will always explore.

Exploiting bees are allocated to the flower believed to have the greatest resources. Exploiting bees add, to their resources collected, the true amount of resources available at a patch divided by the number of bees exploiting that patch this turn. After collecting the resources the exploiting bees update their beliefs about the exploited flower patch to reflect the resources they received this turn.

Exploring bees decide on a patch to explore by employing their assigned learning strategy. "Social bees" randomly select an exploiting bee and explore the patch they are exploiting. "Individual bees" randomly select a patch and explore it. If the patch is already being exploited then the exploring bee's beliefs are updated to be the payoff received by exploiters. If the patch is not being exploited the beliefs are updated with the total resources available.

After exploiting and exploring is complete the ages of all bees are increased by one. Bees that turn one-hundred die. For each bee under the age of 100 a draw is made from a Bernoulli distribution with probability of success d. Bees assigned a value of 1 also die. Dead bees are replaced by newly initialized bees. The learning type of the new bee is assigned in an evolutionary manner; a bee from the remaining population is randomly selected with probabilities proportional to resources collected, the new bee is assigned the same learning type as the randomly selected living bee.

Record Average Collected By Type and Proportion of Learning Types

After each round is complete the average resources collected by social learners and by individual learners are recorded. Additionally the proportion of social learners is recorded.

Time steps and Runs

The model is run for 10000 time steps. This is a sufficiently long time such that the system almost always ends in an equilibrium where all bees are of one learning type. Equilibrium is due to new bees being assigned learning types from the current distribution of learning types. In Smolla, Alem, et al. 2016 the model is then run 100 times under different seeds and the average of the proportion of social learners in the last 2500 time steps is taken as the output.

4.1.2 Step 2 - Choice of Inputs, Parameters and Outputs

The Bumblebee model requires the following quantities to be fixed or fed in as inputs or parameters:

- The number of flowers, N
- The probability of re-sampling from the distribution of resources each time step, τ
- The number of bees, M
- The initial proportion of social learners, p
- The probability of early death each time step, d
- The probability that in a given time step a bee exploits rather than explores, β
- The shape parameter of the gamma distribution describing resources, S
- The rate parameter of the gamma distribution describing resources, R

As the scientific question of interest in the original paper was to investigate the effect of the distribution of resources on the foraging behaviour of bees, it is natural to consider the shape and rate parameters as inputs. We use the fixed values provided by Smolla, Alem, et al. 2016 for the following quantities $N = 100$, $M = 33$, $\tau = 10^{-4}$, $p = 0.5$, $d = 0.02$. Finally to demonstrate that Bayesian melding is robust to unknown parameters, the probability of exploit will be treated as a parameter. It is believed that β has the greatest impact on the relative effectiveness of the social and individualistic learning strategies, and is therefore the most important variable to tune accurately.

To address the high dimensional nature of the output produced by an Agent Based model, we take summary statistics that align with observable real world quantities. In

Smolla, Alem, et al. 2016 the output of this model was summarised by averaging the proportion of social learners in the last 2500 periods of 100 runs of the model with different set seeds. We will follow suit and use the same summary statistic which we call "Sociabillity" denoted by P.

4.1.3 Step 3 - Experimental data

An experiment is conducted in Smolla, Alem, et al. 2016 that investigates the relationship between environmental uncertainty and the sociability of bumblebee foraging. The experiment was run by taking individual bees and placing them in an artificial environment with fake flowers and small objects to simulate other bees. Specifically the "flight arena" was set up with 12 flowers in a regular 3×4 array. Bees completed two training phases in which they were exposed to an environment for five bouts of five minute and then Bees completed a single testing phase.

Training Cuing - Phase One

Bees were trained in an environment where the flowers were transparent and four out of the twelve flowers had cues attached. The flowers without additional cues were filled with 30 ml of water. Of the flowers with cues two were empty and two were filled with 30 ml of 30% sucrose solution each. This was interpreted as training bees to use cues as part of a social learning strategy.

Training Uncertainty Level - Phase Two

Bees were trained in one of two environments where the flowers were yellow and there were no cues. In one of the environments 100ml of 30% sucrose solution was equally divided among all 12 flowers (no-variance) in the other the sucrose was divided between only two, the rest being filled with water (high-variance). This was interpreted as training one group of bees to be in a certain environment and the other an uncertain environment.

Test Phase

In the test phase, bees were placed in an environment with yellow flowers all water-filled, of which four had cues that were consistent with phase one. Bees were allowed only one foraging bout. This was interpreted as placing a bee who believes the environment to be

either certain or uncertain into an environment with social cues and seeing whether they follow a social strategy.

Resulting Data

The experiment provided a data set with the following variables:

- *Id* identifies the specific bee.
- *Colony* identifies the the colony it comes from.
- *CueType* refers to the style of cue used to simulate real bees. For our purposes only clay cues designed to look like bees are considered as other cue types appeared to not be significantly different to no cues.
- *Group* refers to whether the data is drawn from a training round or a test round, and if drawn from a test round, what uncertainty training did the bee undergo.
- *Landed* is an ordered list of the flowers that the bee landed on.
- *Cue* is a list of the flowers which had a clue attached.
- *Reward* lists which flowers had resources present in the training rounds for uncertainty.
- *FirstLanded* records the time between the start of the experiment and the first time a bee lands on a flower.
- *LastLanded* records the time between the start of the experiment and the last time a bee lands on a flower.

4.1.4 Step 4 - Connecting Bee Model with Bee Data

As was noted above the model and system are similar in a fairly abstract way. In step 2 we identified the shape and rate parameters as input quantities and the Sociability summary statistic as an output quantity. Here we construct the likelihoods that will allow us use the experimental observations of our system to inform our beliefs in the appropriate manner.

Connecting Sociability to data

For each bee in the reduced data set we consider, if the first flower that they landed on in

the test round, had a social cue attached. Bees whose first landing was on a flower with a social cue are classified as a social learner. If the flower did not have a social cue, the bee is classified as an individualistic learner.

Using this new variable, the model output is connected to the data. The model output is considered as the proportion of social learning in the super population of bees, who live in environments with this level of uncertainty. Each experimental data point is then a sample from this population. The likelihood for the model output "Sociability" is as follows:

$$y = (s_1, ..., s_n), \quad \text{where} \quad s_i = \begin{cases} 1 \text{ if bee } i \text{ is a social learner} \\ 0 \text{ if bee } i \text{ is an individualistic learner} \end{cases}$$

$$n = 16 = \text{The number of bees in data set}, \qquad k = 14 = \sum_{i=1}^{n} y_i$$

$$L(Sociability|y) = \binom{n}{k} Sociability^k (1 - Sociability)^{n-k}$$

Connecting Uncertainty to data

The experimental environment encountered by each bee is recorded in the variable Reward. For each bee trained in the uncertain environment, two of the twelve flowers had 50ml of sucrose and the others had none. In the theoretical model resources are sampled from a gamma distribution.

This is an example of how the abstract connection between model and system can be problematic. In a gamma distribution the probability of sampling a value of zero is zero, hence it is not obvious how to connect the system quantity (level of sucrose) to the model quantities (Shape and Rate). To overcome this we consider that the system values have been filtered or rounded. We define a new variable that takes a value of 1 if the was a reward and 0 if not. When we do this the data can be seen as independent draws from a binomial distribution. The binomial distribution can then linked to the gamma distribution in the model, by making the probability of success equal to the probability of drawing a sample of greater than, say, 0.1 from the model gamma distribution.

$$x = (u_1, ..., u_n), \quad \text{where} \quad u_i = \begin{cases} 1 \text{ if a reward was present at flower } i \\ 0 \text{ if a reward was not present at flower } i \end{cases}$$

$n = 12 =$ The total number of flowers, $\quad k = 2 =$ The number of flowers with rewards

$$L(shape, rate|x) = \binom{n}{k} P^k (1-P)^{n-k} \quad \text{where } P = \int_{0.1}^{\infty} \frac{rate^{shape}}{\Gamma(shape)} x^{shape-1} \exp^{-(rate)x} dx$$

4.1.5 Step 5 - Defining Marginal Priors

Given the choices made above there are four quantities of interest: The probability that on a given turn each bee exploits rather than explores (β), the shape and rate parameters of the gamma distribution describing resources (S) and (R), and the average proportion of social learning (P).

The quantities have the following supports: $0 < S$, $0 < R$, $0 \leq \beta \leq 1$, $0 \leq P \leq 1$. The original paper prescribes the following values: $S = 0.183$,$R = 0.022$ and $\beta = 0.8$. Given this we take as priors the most diffuse uniform distributions that are both within the supports and are roughly centred at the prescribed values. The prior on P was arbitrarily chosen to be uniform on $(0.75, 1)$.

$$S \sim u(0, 0.366), \quad R \sim u(0, 0.044), \quad \beta \sim u(0.5, 1), \quad P \sim u(0.75, 1)$$

$$q^{1p}_{[S,R,\beta]} = \begin{cases} 124.1927, & S \in (0, 0.366), R \in (0, 0.044), \beta \in (0.5, 1) \\ 0, & \text{elsewhere} \end{cases}$$

$$q^{2}_{[P]} = \begin{cases} 4, & P \in (0.75, 1) \\ 0, & \text{elsewhere} \end{cases}$$

4.1.6 Step 6 - Constructing the Joint Prior

The emulator approximation method is used to construct a joint prior that is both consistent with the Bumblebee model and approximately in agreement with the desired marginal priors. As was the case in the deterministic example from chapter two, the matrix used to emulate the Bumblebee model is singular, so a Moore-Penrose generalised inverse is used. For interpolation/smoothing, multivariate adaptive regression splines are used as implemented by the R package mars. The smoothed function is normalised into a proper probability distribution.

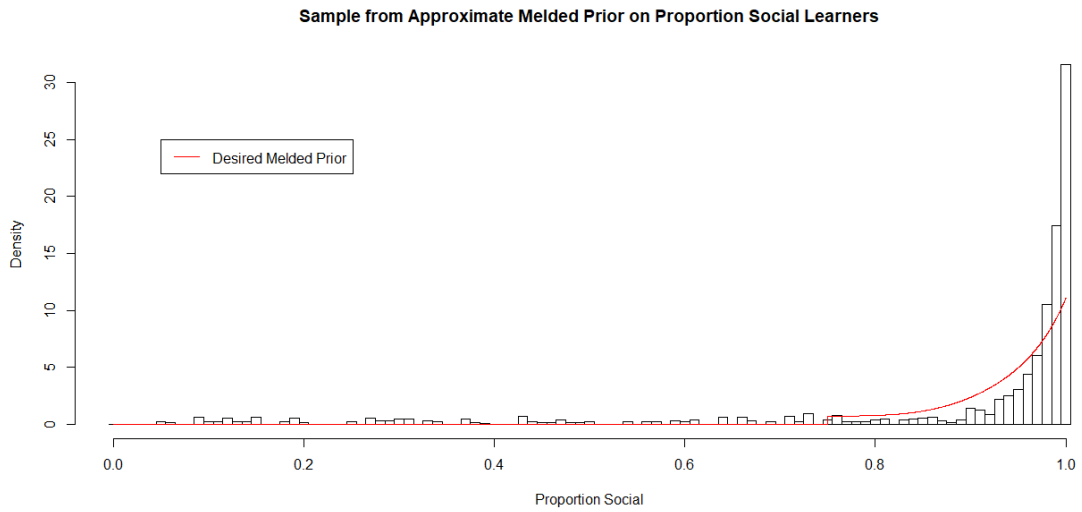

Figure 4.1: Result Bumblebee Model - Emulator Approximation of Prior

4.1.7 Step 7 - Checking the Joint Prior

A sample from the resulting marginal prior on outputs is shown in Figure(4.1). The red line is the desired melded prior. There are two major points of interest.

Firstly, the desired melded prior places zero probability on values of P below 0.75 yet the approximate prior places non-zero probability in this interval. We believe this has occurred because, for all values of input-parameters the theoretical model places at least some small probability on all values of P. That is to say, there is no combination of inputs and parameters that places zero probability on $p < 0.75$. Given this, choosing the pre-model prior on outputs to have that feature may have been unreasonable. In this sense the emulator approximation method has allowed us to identify an unrealistic assumption.

Secondly, the curve of approximate prior at high values of P is noticeably steeper than in the desired prior. It is unclear whether this discrepancy is due to the non-existence of a solution with the desired property or to an inexact approximation. Notwithstanding the approximate prior sufficiently captures the key features of the desired prior and we proceed with posterior inference.

4.1.8 Step 8 - Sampling from the Posterior

Using the prior constructed in step 6 and the sampling method outlined in chapter two, samples are obtained from the joint posterior. The empirical marginal distributions are shown in Figure(4.2).

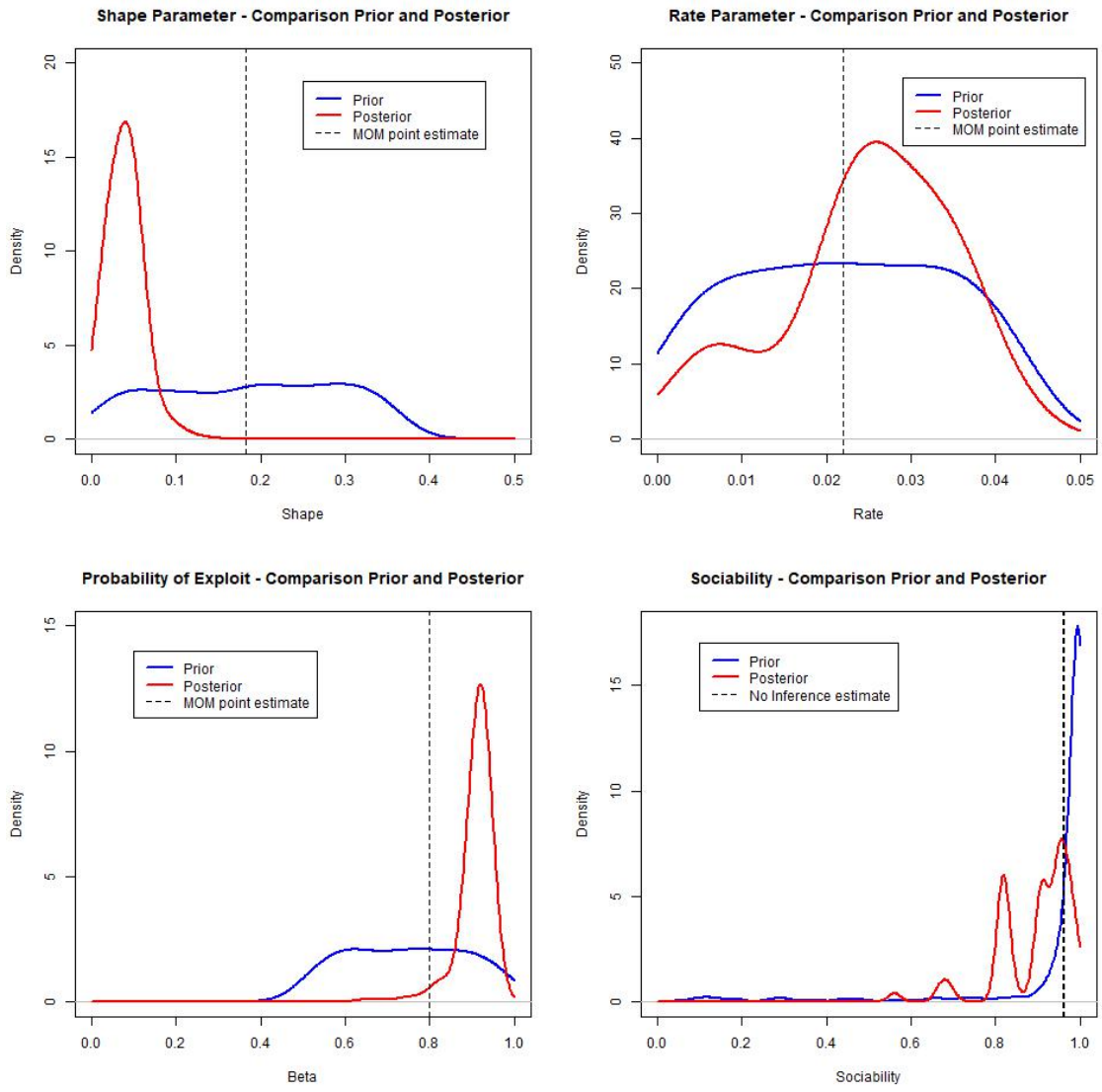

Figure 4.2: Result Bumblebee Model - Comparison of Priors and Posteriors

4.1.9 Discussion

The first thing to note about the results is that, by using Bayesian melding to incorporate all data into a joint inference, our posterior beliefs about the shape and rate quantities, and the proportion of social learning are different to what we would have concluded if the inference had been made separately. This illustrates how our theoretical understanding of the system is able to inform our conclusions in a rigorous and unified manner.

Furthermore we note the substantial narrowing of our beliefs about the probability of exploiting, β. This illustrates how, by using Bayesian melding to perform joint inference, we were able to learn about a parameter value, associated with $M()$, that is otherwise unobservable. This learning is exactly the missing link referred to by Grazzini and M. Richiardi 2015 and Chen, Chang, and Du 2012, and it demonstrates how Bayesian melding can be used for rigorous "parameter selection".

4.2 Application of Bayesian Melding (Flexible Likelihoods)

Here we present an extension to the above model, that illustrates the flexibility of likelihoods in Bayesian melding. On the input side we demonstrate a likelihood that accounts for missing/censored data. On the output side we demonstrate a likelihood that decomposes error into a colony effect and a sampling effect. First the statistical models are discussed and presented. An algorithm is then designed to sample from the posterior. Finally we present the results.

4.2.1 Missing/Censored Input Data

It is noted that there is a discrepancy between the observed data on the distribution of resources and how they are described in the theoretical model. Specifically, the data was that ten of the twelve flowers contained no resources. The theoretical model describes resources as being random samples from a gamma distribution, but there is zero probability associated with sampling zero from a gamma distribution. To proceed, we formulate a meaningful interpretation of the data, in the context of the theoretical model. Hence the data is regarded as censored if a value is in the interval $(0, 0.1]$. The associated statistical

model is presented below.

$$x_3, \cdots, x_{12} \overset{iid}{\sim} \text{Gamma}\,(S, R)\,\mathbb{1}(0, 0.1]$$

4.2.2 Colony Effects

The full list of variables in the experimental data are explained above. The theoretical model does not account for the colony the bee comes from. Here we investigate the colony effect by modelling it within the likelihood. Thus the output data on sociability is grouped by colony and modelled with random effects.

$$\begin{gathered} y_{ij} \overset{iid}{\sim} \text{Bern}(\pi_i) \\ \log\left(\frac{\pi_i}{1-\pi_i}\right) = \log\left(\frac{P}{1-P}\right) + \tau_i \\ \tau_i \overset{iid}{\sim} N(0, \sigma^2) \end{gathered}$$

In the random effects model we consider a grand mean equal to the model quantity P and a colony effect which we denote τ. The full statistical model is written out below. Note: the sociability data is now separated by colony. There are three colonies, a, b and c with 2, 11 and 3 data points respectively. The priors on S, R and β are the same as in the previous application. A gamma prior is placed on σ:

$$\sigma \sim \text{Gamma}\,(1, 0.05)$$

4.2.3 Updating Beliefs

Here the more detailed posterior of the resulting model is presented and a sampling algorithm is developed to examine the posterior. A Markov-chain Monte Carlo (MCMC) sampling algorithm is used. Notably an SIR step within the MCMC allows for reuse of draws from the approximate melded prior in all MCMC iterations. Thus reevaluating the computationally expensive Agent Based model is avoided.

Posterior

$\pi(S, R, \beta, \sigma, \tau_a, \tau_b, \tau_c, x_3, \cdots, x_{12}|X, Y) \propto$

$$q(S,R,\beta,\sigma)L(S,R,x_3,\cdots,x_{12}|X)\int\limits_{P\in[0,1]} L(P,\sigma,\tau_a,\tau_b,\tau_c|Y)p(P|S,R,\beta)dP$$
$$Y=y_a,y_b,y_c=y_{a1},y_{a2},y_{b1},\cdots,y_{b11},y_{c1},\cdots,y_{c3},\quad X=x_1,x_2$$

Input Likelihood: $L(S,R,x_3,\cdots,x_{12}|X)$

$$L(S,R,\beta,x_3,\cdots,x_{12}|x_1,x_2)=p(x_1,x_2|S,R)p(x_3,\cdots,x_{12}|S,R)$$

Output Likelihood: $L(P,\sigma,\tau_a,\tau_b,\tau_c|Y)$

$$L(P,\sigma,\tau_a,\tau_b,\tau_c|y_{a1},y_{a2},y_{b1},\cdots,y_{b11},y_{c1},\cdots,y_{c3})=$$
$$p(y_{a1},y_{a2}|\tau_a,P)p(y_{b1},\cdots,y_{b11}|\tau_b,P)p(y_{c1},\cdots,y_{c3}|\tau_c,P)p(\tau_a|\sigma)p(\tau_b|\sigma)p(\tau_c|\sigma)$$

Priors: $q(S,R,\beta,\sigma)$

$$\tilde{q}_{[S,R,\beta]}(S,R,\beta)q(\sigma)$$

Sampling Algorithm

1. Obtain samples from the melded input-parameter prior, using the emulator approximation method

2. Select initial values for: $S,R,\beta,\sigma,\tau_a,\tau_b,\tau_c$

3. Conduct an MCMC iteration:

   (a) Gibbs step: Sample $x_3,...,x_{12}$ from a truncated gamma

   $$x_3,...,x_{12}\overset{iid}{\sim}\text{Gamma}\,(S,R)\,\mathbb{1}(0,0.1]$$

   (b) Metropolis-Hastings step: Propose, accept/reject a new value for σ:

      i. Propose new value

      $$\sigma^*=\mid\sigma^{[s]}+j\mid,\quad j\sim N(0,10)$$

      ii. Calculate Metropolis-Hastings ratio

      $$r=\frac{q_\sigma(\sigma^*)p(\tau_a|\sigma^*)p(\tau_b|\sigma^*)p(\tau_c|\sigma^*)}{q_\sigma(\sigma^{[s]})p(\tau_a|\sigma^{[s]})p(\tau_b|\sigma^{[s]})p(\tau_c|\sigma^{[s]})}$$

iii. Accept/Reject

$$\begin{cases} \sigma^{[s+1]} = \sigma^{[s]}, & r < u \\ \sigma^{[s+1]} = \sigma^{*}, & r \geq u \end{cases}, \qquad u \sim u(0,1)$$

(c) Metropolis-Hastings step: Propose, accept/reject new values for τ_a

i. Propose new value

$$\tau_a^{*} = \tau_a^{[s]} + j, \quad j \sim N(0, 30)$$

ii. Calculate Metropolis-Hastings ratio

$$r = \frac{p(\tau_a^{*}|\sigma)\frac{1}{K}\sum_{k=1}^{K} p(y_a|\pi_{a,k}^{*})p(y_b|\pi_{b,k})p(y_c|\pi_{c,k})}{p(\tau_a^{[s]}|\sigma)\frac{1}{K}\sum_{k=1}^{K} p(y_a|\pi_{a,k}^{[s]})p(y_b|\pi_{b,k})p(y_c|\pi_{c,k})}, \quad \pi_{i,k} = \frac{\exp(\log(\frac{P_k}{1-P_k}) + \tau_i)}{1 + \exp(\log(\frac{P_k}{1-P_k}) + \tau_i)})$$

Where $P_1, \cdots, P_K$ are the model evaluations associated with current values of S,R, and β. The sum is a Monte-Carlo approximation as discussed chapter two

iii. Accept/Reject

$$\begin{cases} \tau_a^{[s+1]} = \tau_a^{[s]}, & r < u \\ \tau_a^{[s+1]} = \tau_a^{*}, & r \geq u \end{cases}, \qquad u \sim u(0,1)$$

(d) Metropolis-Hastings step: Propose, accept/reject new values for τ_b

i. Propose new value

$$\tau_b^{*} = \tau_b^{[s]} + j, \quad j \sim N(0, 30)$$

ii. Calculate Metropolis Hastings ratio

$$r = \frac{p(\tau_b^{*}|\sigma)\frac{1}{K}\sum_{k=1}^{K} p(y_a|\pi_{a,k})p(y_b|\pi_{b,k}^{*})p(y_c|\pi_{c,k})}{p(\tau_b^{[s]}|\sigma)\frac{1}{K}\sum_{k=1}^{K} p(y_a|\pi_{a,k})p(y_b|\pi_{b,k}^{[s]})p(y_c|\pi_{c,k})}, \quad \pi_{i,k} = \frac{\exp(\log(\frac{P_k}{1-P_k}) + \tau_i)}{1 + \exp(\log(\frac{P_k}{1-P_k}) + \tau_i)})$$

Where $P_1, \cdots, P_K$ are the model evaluations associated with current values of S,R, and β. The sum is a Monte-Carlo approximation as discussed chapter two

iii. Accept/Reject

$$\begin{cases} \tau_b^{[s+1]} = \tau_b^{[s]}, & r < u \\ \tau_b^{[s+1]} = \tau_b^{*}, & r \geq u \end{cases}, \qquad u \sim u(0,1)$$

(e) Metropolis-Hastings step: Propose, accept/reject new values for τ_c

i. Propose new value

$$\tau_c^{*} = \tau_c^{[s]} + j, \quad j \sim N(0, 30)$$

ii. Calculate Metropolis-Hastings ratio

$$r = \frac{p(\tau_c^{*}|\sigma)\frac{1}{K}\sum_{k=1}^{K} p(a_i|\pi_{a,k})p(y_b|\pi_{b,k})p(y_c|\pi_{c,k}^{*})}{p(\tau_c^{[s]}|\sigma)\frac{1}{K}\sum_{k=1}^{K} p(y_a|\pi_{a,k})p(y_b|\pi_{b,k})p(y_c|\pi_{c,k}^{[s]})}, \quad \pi_{i,k} = \frac{\exp(\log(\frac{P_k}{1-P_k}) + \tau_i)}{1 + \exp(\log(\frac{P_k}{1-P_k}) + \tau_i}))$$

Where $P_1, \cdots, P_K$ are the model evaluations associated with current values of S,R, and β. The sum is a Monte-Carlo approximation as discussed chapter two.

iii. Accept/Reject

$$\begin{cases} \tau_i^{[s+1]} = \tau_i^{[s]}, & r < u \\ \tau_i^{[s+1]} = \tau_i^{*}, & r \geq u \end{cases}, \qquad u \sim u(0,1)$$

(f) SIR step: sample of S, R, β

i. Calculate the SIR weights for each of the L draws from the melded prior

$$w_l = p(x_1, x_2, x_3, \cdots, x_{12}|S, R)\frac{1}{K}\sum_{k=1}^{K} p(y_{a1}, y_{a2}, y_{b1}, \cdots, y_{b11}, y_{c1}, \cdots, y_{c3}|P_k, \tau_a, \tau_b, \tau_c)$$

Where $P_1, \cdots, P_K$ are the model evaluations associated with current values of S,R, and β. The sum is a Monte-Carlo approximation as discussed chapter two

ii. Draw S, R, β and the associated $P_1, ..., P_K$

4. Repeat step 3

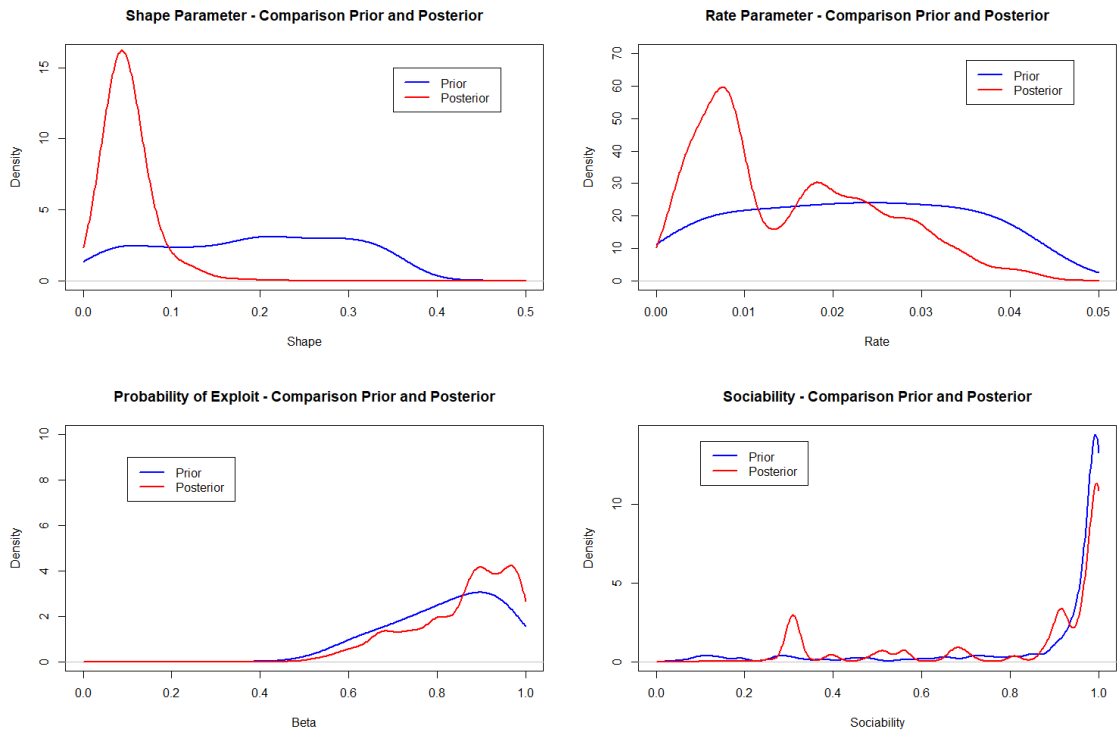

Figure 4.3: Result Extended Bumblebee Model - Comparison of Priors and Posteriors

4.2.4 Results

Here we present the results from the extended bumblebee application. 100,000 scans were run and the first 999 were removed for burn-in. In Figure (4.3) the marginal priors and posteriors for model quantities are shown. The inference is similar to the model without the hive effect but the posteriors are more similar to the priors here. We note that because inference is being conducted on more variables with the same data, the marginal inference is weakened.

In Figure (4.4) we show the posteriors for each colony effect, τ_i. There were relatively few data points per colony, and we note that the posteriors exhibit much overlap. This suggests that the colony effect may not be significant. To test this idea more formally, Bayes factors are calculated with and without the colony effect. The Bayes factor for the full model was 1519.061 and for the reduced model 1312.406. As suggested in Kass and Raftery 1995 $2*\log(\text{Colony Model/No Colony Model})$ is taken and found to be -0.292461. Such a low score indicates that the colony model is likely not necessary.

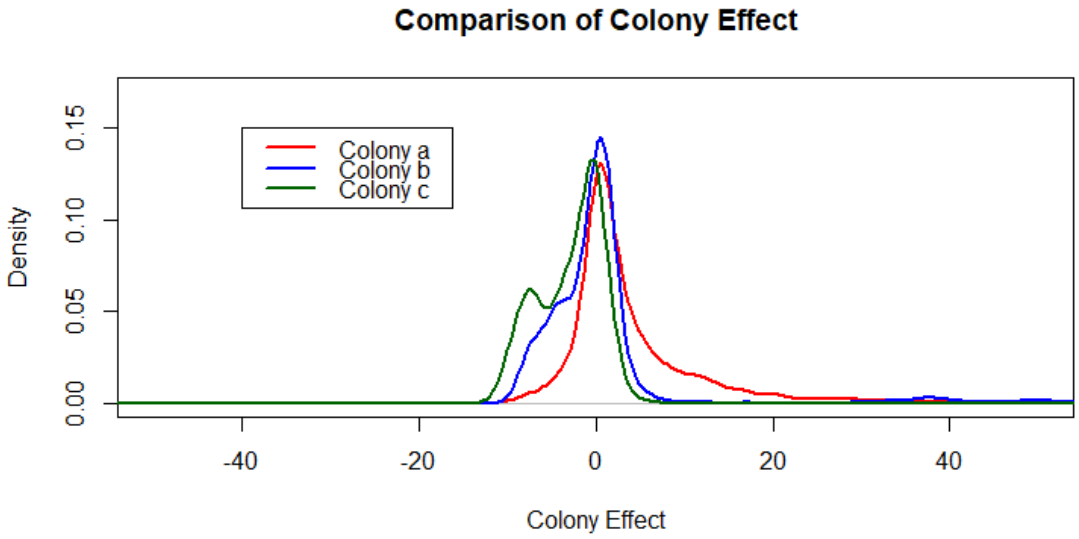

Figure 4.4: Result Extended Bumblebee Model - Comparison of Colony Effect Posteriors

5. 5. Discussion

5.1 Future Research

Prediction and Model Validation

In this thesis we have demonstrated the use of Bayesian melding for parameter specification in the context of Agent Based modelling. Parameter specification is just one step in the process of scientific modelling. In future work we intend to investigate prediction and model validation techniques that are natural extensions to the work here.

High Dimensional Outputs

The issue of high dimensional outputs has repeatedly appeared in this thesis. We have observed that the dimension of outputs is often reduced using some summary notion. Here we offer a categorization of the levels at which outputs are summarized:

1. *Types of information*: Instead of reporting each agent's full list of attributes, some are omitted or summary statistics are used.
2. *Space*: Spatial relationships are no longer reported.
3. *Agents*: Agents are no longer identified.
4. *Time*: Time can either be summarized by taking an aggregate measure over some time period or by waiting for the system to reach an equilibrium and reporting the equilibrium result.
5. *Stochasticity*: Stochasticity can be summarized by aggregating over multiple runs of the model.

In future work we intend to investigate the nature of these reductions. Of specific interest is the nature of reductions in time. Grazzini, M. G. Richiardi, and Tsionas 2017 offers

some discussion, and touches on ideas of equilibrium from economics. We hope to use the model in Lum et al. 2014 to investigate ways Bayesian melding can be used to align the notion of model time with real time.

Alternative Emulators

In this thesis we presented the emulator approximation method. The emulator approximation method extended the Bayesian melding procedure for constructing priors, to stochastic theoretical models. In the application presented here, linear systems of equations were used to approximate the true relationship between the melded prior on outputs the theoretical model and the melded prior on inputs. This choice resulted in emulating the theoretical model with a matrix of transitional probabilities from discrete grid cells. A matrix is a non-parametric emulator and requires relatively many model evaluations. In future work we will investigate the use of parametric emulators such as Copulas. Specifically we are interested in whether the associated system is easily solved and if efficiency can be improved allowing us to extend the emulator approximation method to higher dimension problems.

5.2 Contributions

The primary contributions of this thesis to the literature are summarized below:

- Extending Bayesian melding to stochastic theoretical models.
- The use of Bayesian melding to address parameter specification in Agent Based models.
- Integration of Sampling Importance Resampling with Markov Chain Monte Carlo for efficient inference on quantities related to the model as well as the extended likelihoods associated with complicated data structures.

Secondary contributions include:

- The implementation of the methods of this thesis in R.
  - Bayesian Melding
  - Emulator Approximation method

  - Agent Based modelling

- Discussion of the philosophy and practicality of Bayesian melding.
- Discussion of the philosophy and practicality of Agent Based modelling.

Bibliography

Alkema, Leontine, Adrian E Raftery, and Samuel J Clark (2007). "Probabilistic projections of HIV prevalence using Bayesian melding". In: *The Annals of Applied Statistics*, pp. 229–248.

Bae, Jang Won et al. (2016). "Combining microsimulation and agent-based model for micro-level population dynamics". In: *Procedia Computer Science* 80, pp. 507–517.

Banisch, Sven (2015). *Markov Chain Aggregation for Agent-Based Models*. Springer.

Chen, Shu-Heng, Chia-Ling Chang, and Ye-Rong Du (2012). "Agent-based economic models and econometrics". In: *The Knowledge Engineering Review* 27.2, pp. 187–219. DOI: 10.1017/S0269888912000136.

Chiu, GS and JM Gould (2010). "Statistical inference for food webs with emphasis on ecological networks via Bayesian melding". In: *Environmetrics* 21.7-8, pp. 728–740.

Dubois, Guillaume (2018). *Modeling and Simulation: Challenges and Best Practices for Industry*. CRC Press.

Epstein, Joshua M and Robert Axtell (1996). *Growing artificial societies: social science from the bottom up*. Brookings Institution Press.

Falk, MG, RJ Denham, and KL Mengersen (2010). "Estimating uncertainty in the revised universal soil loss equation via Bayesian melding". In: *Journal of agricultural, biological, and environmental statistics* 15.1, pp. 20–37.

Grazzini, Jakob and Matteo Richiardi (2015). "Estimation of ergodic agent-based models by simulated minimum distance". In: *Journal of Economic Dynamics and Control* 51, pp. 148–165.

Grazzini, Jakob, Matteo G Richiardi, and Mike Tsionas (2017). "Bayesian estimation of agent-based models". In: *Journal of Economic Dynamics and Control* 77, pp. 26–47.

Grimm, Volker et al. (2006). "A standard protocol for describing individual-based and agent-based models". In: *Ecological Modelling* 198.1, pp. 115–126. ISSN: 0304-3800.

DOI: https://doi.org/10.1016/j.ecolmodel.2006.04.023. URL: http://www.sciencedirect.com/science/article/pii/S0304380006002043.

Heard, Daniel (2014). "Statistical Inference Utilizing Agent Based Models". In:

Heard, Daniel et al. (2015). "Agent-Based models and microsimulation". In: *Annual Review of Statistics and Its Application* 2, pp. 259–272.

Hooten, Mevin B et al. (2011). "Assessing first-order emulator inference for physical parameters in nonlinear mechanistic models". In: *Journal of Agricultural, Biological, and Environmental Statistics* 16.4, pp. 475–494.

Kass, Robert E and Adrian E Raftery (1995). "Bayes factors". In: *Journal of the american statistical association* 90.430, pp. 773–795.

Laubenbacher, Reinhard et al. (2009). "Agent Based Modeling, Mathematical Formalism for". In: *Encyclopedia of Complexity and Systems Science*. Ed. by Robert A Meyers. New York, NY: Springer New York, pp. 160–176. ISBN: 978-0-387-30440-3. DOI: 10.1007/978-0-387-30440-3_10. URL: https://doi.org/10.1007/978-0-387-30440-3%7B%5C_%7D10.

Lum, Kristian et al. (2014). "The contagious nature of imprisonment: an agent-based model to explain racial disparities in incarceration rates". In: *Journal of The Royal Society Interface* 11.98, p. 20140409.

Polhill, J Gary et al. (2008). "Using the ODD protocol for describing three agent-based social simulation models of land-use change". In: *Journal of Artificial Societies and Social Simulation* 11.2, p. 3.

Poole, David and Adrian E Raftery (2000). "Inference for deterministic simulation models: the Bayesian melding approach". In: *Journal of the American Statistical Association* 95.452, pp. 1244–1255.

Radtke, Philip J, Thomas E Burk, and Paul V Bolstad (2002). "Bayesian melding of a forest ecosystem model with correlated inputs". In: *Forest Science* 48.4, pp. 701–711.

Raftery, Adrian E, Geof H Givens, and Judith E Zeh (1995). "Inference from a deterministic population dynamics model for bowhead whales". In: *Journal of the American Statistical Association* 90.430, pp. 402–416.

Rickles, Dean, Penelope Hawe, and Alan Shiell (2007). "A simple guide to chaos and complexity". In: *Journal of Epidemiology & Community Health* 61.11, pp. 933–937.

Schelling, Thomas C (1969). "Models of segregation". In: *The American Economic Review* 59.2, pp. 488–493.

Ševčíková, Hana, Adrian E Raftery, and Paul A Waddell (2007). "Assessing uncertainty in urban simulations using Bayesian melding". In: *Transportation Research Part B: Methodological* 41.6, pp. 652–669.

Ševčıková, Hana, Adrian E Raftery, and Paul A Waddell (2011). "Uncertain benefits: Application of bayesian melding to the alaskan way viaduct in seattle". In: *Transportation Research Part A: Policy and Practice* 45.6, pp. 540–553.

Smolla, Marco, Sylvain Alem, et al. (2016). "Copy-when-uncertain: bumblebees rely on social information when rewards are highly variable". In: *Biology letters* 12.6, p. 20160188.

Smolla, Marco, R Tucker Gilman, et al. (2015). "Competition for resources can explain patterns of social and individual learning in nature". In: *Proc. R. Soc. B.* Vol. 282. 1815. The Royal Society, p. 20151405.

Wolpert, Robert L (1995). "Comment". In: *Journal of the American Statistical Association* 90.430, pp. 426–427.