Input Uncertainty and Data Collection Problems in Stochastic Simulation

Drupad Parmar, BSc (Hons), MSc, MRes

Submitted for the degree of Doctor of Philosophy at Lancaster University.

September 2023

STOR-i
excellence with impact

Abstract

Stochastic simulation is an important tool within the field of operational research. It allows for the behaviour of random real-world systems to be analysed, evaluated, and optimised. It is critical to understand the uncertainty and error in outcomes from simulation experiments, to ensure that decisions are made with appropriate levels of confidence.

Frequently, input models that actuate stochastic simulations are estimated using samples of real-world data. This introduces a source of uncertainty into the simulation model which propagates through to output measures, causing an error known as input uncertainty. Input uncertainty depends on the samples of data that are collected and used to estimate the input models for the simulation. In this thesis, we consider problems relating to input uncertainty and data collection in the field of stochastic simulation.

Firstly, we propose an algorithm that guides the data collection procedure for a simulation experiment in a manner that minimises input uncertainty. Reducing the uncertainty around the simulation response allows for improved insights to be inferred from simulation results. Secondly, we outline an approach for comparing data collection strategies in terms of the input uncertainty passed to outputs in simulations of viral loads. This represents a different type of data collection problem to the ones usually studied in simulation experiments. Thirdly, we adapt two techniques for quantifying input uncertainty to consider a quantile of the simulation outputs, rather than the

mean. Quantiles are regularly used to provide alternative information regarding the simulation outputs relative to the mean, therefore it is equally important to understand the uncertainty of such measures. Finally, we begin to investigate how input uncertainty impacts predictive models fit to simulation data. This relates to the field of simulation analytics, a novel and emergent area of research where the problem of input uncertainty has not previously been examined.

Acknowledgements

I would like to start by thanking my supervisors, Lucy, Andrew, Susan, and Richard. It has been a real pleasure to work with you all. You have each made unique and valuable contributions to this thesis, all of which are very much appreciated. Thank you for the enduring advice, guidance, patience, and wisdom over an unusually turbulent few years, you have taught me a great deal.

I would also like to thank all the staff and students at STOR-i, both past and present, for creating such a friendly and stimulating environment in which to conduct research, I have thoroughly enjoyed my time here. I am grateful for the financial support, as well as the numerous opportunities that STOR-i students are privileged to receive. It would be negligent of me not to mention my cohort, who I must thank for being constant sources of both help and hilarity, it has been an unforgettable journey.

A major part of my PhD experience, though not part of this thesis, was the internship I completed at Sportlight Technology. I would like to thank the whole Sportlight team for being so welcoming and making my time there remarkably insightful and enjoyable. Particular thanks go to Ian, who afforded me the opportunity.

Penultimately I would like to thank my friends from Durham and LSE, who are kind, generous, and funny, and have always taken a keen interest in the culmination of my PhD (asking when I will be finished). Last, but by no means least, I must thank my family, who have been unwavering with their incredible encouragement and support, for that I am extremely grateful.

Declaration

I declare that the work in this thesis has been done by myself and has not been submitted elsewhere for the award of any other degree.

A version of Chapter 3 has been published as Parmar, D., Morgan, L.E., Titman, A.C., Williams, R.A., and Sanchez, S.M. (2021). A two stage algorithm for guiding data collection towards minimising input uncertainty. In *Proceedings of the Operational Research Society Simulation Workshop 2021*, pages 127-136. Operational Research Society.

Chapter 4 has been published as Parmar, D., Morgan, L.E., Titman, A.C., Regnier, E.D., and Sanchez, S.M. (2021). Comparing data collection strategies via input uncertainty when simulating testing policies using viral load profiles. In *Proceedings of the 2021 Winter Simulation Conference*, pages 1-12. IEEE.

Chapter 5 has been published as Parmar, D., Morgan, L.E., Titman, A.C., Williams, R.A., and Sanchez, S.M. (2022). Input uncertainty quantification for quantiles. In *Proceedings of the 2022 Winter Simulation Conference*, pages 97-108. IEEE.

This thesis is approximately 35700 words.

Drupad Parmar

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Stochastic Simulation

Stochastic simulation serves as a tool for decision-making, allowing users to experiment with and optimise a model of some inherently random real-world system. Simulation models usually take the form of computer code, and can be as simple or as complicated as needed, depending on the objectives of the simulation study. Examples of real-world systems frequently modelled using simulation include airports, hospitals, and manufacturing lines. When studying these complex systems and assessing their behaviour, certain characteristics or performance measures are often of interest, such as resource usage, queue lengths, and waiting times. Stochastic simulation enables the estimation of such performance measures by capturing the randomness of the system via input distributions or processes, and tracing random variables as they change stochastically over time.

Simulation models are widely applicable and allow practitioners to run experiments with a proxy of a real-world system, or even a system that does not exist. Stochastic simulation is typically utilised when analytical models cannot be applied, and where performance measures are mathematically intractable or cannot be approximated with

a bound on the error (Nelson, 2013). In the case of mathematically tractable models there is an argument that stochastic simulation should be preferred, since in such models assumptions are frequently made purely for tractability reasons. These unrealistic simplifying assumptions affect the validity of the model and the impact of this cannot be quantified, potentially leading to false insights and poor decisions (Lucas et al., 2015). Stochastic simulation allows complex systems to be analysed and estimates of output metrics to be obtained using less restrictive modelling assumptions, which results in a more valid and realistic (albeit an often more complicated) model. This in turn leads to a more accurate understanding of the behaviour of the system and importantly should culminate in better decision-making.

Simulation has far and wide-ranging capabilities, from finding optimal revised aviation schedules due to airline disruption (Rhodes-Leader et al., 2018), to modelling the impact of impaired communication abilities on the capacity of the U.S. army to carry out an effective attacking operation (Cioppa et al., 2004). Even when considering a specific application, simulation can be used to tackle many aspects of a multi-faceted problem. For example, Currie et al. (2020) provide a discussion of the myriad ways in which simulation modelling can be used to mitigate the effects of the COVID-19 pandemic. They discuss the applicability of various simulation techniques to different decision-making scenarios, from quarantine strategies and social distancing measures at the population level, to staffing levels and capacity planning in hospital operations. Although they specifically consider the coronavirus disease, the matching of simulation methods appropriate for supporting these kinds of healthcare decisions is no doubt relevant for other epidemics, whilst simulation can also be used to model the propagation of infectious diseases themselves (Sanchez and Sanchez, 2015).

Stochastic simulation is utilised across an extensive range of industries. Günal and Pidd (2010) provide a literature review of discrete event simulation models applied in the health care industry. Even with a narrow focus on patient flow in secondary care

provided by hospitals, there exists an extremely diverse set of studies with a variety of different objectives, from increasing throughput in an emergency department, to designing an appointment system for a dermatology clinic. The literature includes numerous simulation projects conducted in conjunction with local branches of the UK National Health Service (NHS), which is reported to be one of the largest non-military public organisations in the world. Another area where simulation is regularly applied is the manufacturing industry. Jahangirian et al. (2010) conduct a broad review of publications in this field, and in doing so find an increasing number of simulation studies applied to real-world problems with real-world data. Specifically, simulation has been used within this area to solve problems relating to scheduling, resource allocation, workforce planning, and transportation management, to name a few. The literature review includes papers written in collaboration with Deutsche Bahn AG, the national railway company of Germany, and Hyundai Heavy Industries, the largest shipbuilding company in the world.

1.2 Input Uncertainty

The abstract framework of a simulation model consists of two key factors, inputs, and logic. Inputs refer to the stochastic or random elements within the system, and logic refers to the rules or behaviours of the system as a function of the inputs (Nelson, 2013). Results from stochastic simulation experiments are conditional on the input models and logic of the simulation model accurately representing those of the real-world system, or at least being similar enough to yield useful outcomes. Three types of error arise in stochastic simulation. These may cause the simulation model to exhibit far different behaviour from the real-world system, or result in unreliable and uncertain outputs, both of which may induce poor decision-making. In the context of healthcare or military operations, for example, such errors could have life-changing implications.

Firstly, there exists modelling error, which arises when the logic of the simulated system differs from the logic of the real-world system. In practice, this will almost always be the case, since some rules or behaviours of the real-world system may be inexplicable or unquantifiable. What is important, however, is that the logic of the simulation model is accurate enough to render any results useful, and this can be checked through model verification and validation; see Sargent (2010) for more details.

Secondly, due to the inherent randomness in any system, performance measures will differ from simulation to simulation and this describes a type of error known as stochastic uncertainty, or stochastic estimation error. Typically, the effects of this error are easily quantifiable, and can be made negligible through a large enough simulation effort, however there can sometimes be complexities due to bias and correlated data (Nelson, 2013).

Finally, if the simulation input probability models used to approximate the input models of the real-world system are estimated from data, then these will contain some error due to the finite nature of the samples. This constitutes the third source of error, known as input uncertainty, which is the focus of this thesis.

The input models for the simulation, represented by probability distributions or processes, are often estimated by applying statistical methods, such as maximum likelihood estimation, to observations collected from the real-world system. Since the sample of data collected is finite, uncertainty arises in the estimated input models. As the number of observations increases, the error in the input models decreases, but they never perfectly represent the input models of the real system. The unknown difference between the input models used in the simulation and those that best characterise the real-world system is a source of error generally referred to as input uncertainty. As the input models drive the simulation, this uncertainty propagates through to the performance measure outputs. Rarely, however, is the propagation of input model error considered in simulation output analysis (Barton, 2012), and traditional methods for simulation

output analysis do not account for such uncertainty.

Common practice is to report a simulation-based confidence interval for the performance measure outputs, however these typically ignore the effects of input uncertainty, and only include stochastic uncertainty. Input uncertainty, or input uncertainty variance, refers to an additional source of variance around the performance measure due to having estimated the input models. Input uncertainty can overwhelm stochastic uncertainty, particularly when only small samples of real-world data are available (Corlu et al., 2020). Unlike stochastic uncertainty, the effects of input uncertainty cannot be reduced by simply increasing the simulation effort. Without considering the propagation of input model error in simulation-based confidence intervals, crucial and expensive decisions are at risk of being made with misleading levels of confidence, which could have a serious impact on the success of projects, corporate profitability and even human lives (Barton et al., 2014). Interest therefore lies in quantifying the uncertainty that arises in the simulation output as a result of the uncertainty from estimating the input models. This variability can then be taken into account when making decisions based on the simulation outputs, or be used to justify the collection of additional real-world data.

Though input uncertainty is used to broadly describe the impact of not knowing the true input models, quantification of input uncertainty typically focuses on the variance in the simulation output due to a lack of knowledge about the true input models. However, the impact of the uncertain input models is actually twofold; in addition to the input uncertainty variance, input model bias is introduced. Input model bias refers to the bias in the expected value of the simulation performance measure due to having estimated the input models. Until recently, input model bias has largely been ignored, since it is known to decrease by an order of magnitude quicker than input uncertainty when the number of real-world observations increases. If, however, the number of observations is small, then the bias could be consequential, and so should

not be ignored (Morgan et al., 2019).

As simulation models become increasingly more complex, there has been a proliferation of specialised techniques for working with such models. This includes utilising efficient, high-dimensional designed experiments to explore the impact of input factors on simulation responses (Sanchez et al., 2020), and the development of data farming, where simulation analysts aim to grow their simulated data set meaningfully, analogous to the way a farmer cultivates his land to maximise yield (Sanchez, 2020). Such techniques are required since models potentially have thousands of input factors interacting in convoluted and nonlinear ways. In these instances, uncertainty quantification becomes even more important, since larger models are likely to contain more uncertain input models, as well as display more complex behaviour that could significantly propagate such uncertainty.

1.3 Research Problems and Motivations

In this thesis, we investigate various problems relating to data collection and input uncertainty in the field of stochastic simulation. We now outline our research problems and motivations for the contributions that follow in each chapter of this thesis.

The first research problem we are interested in is whether input uncertainty can be taken into consideration prior to any input data collection. The input uncertainty contribution to the variance of the simulation output depends on the collection of input model data, and in particular, the number of observations used to estimate each of the input model distributions. If input uncertainty is of a considerable magnitude, it becomes difficult to draw meaningful inferences from the simulation results, and decision-making is impeded. Unlike stochastic uncertainty, input uncertainty cannot be reduced by exerting additional simulation effort. Instead, to reduce input uncertainty additional input data can be collected to enable estimation of higher fidelity input models, which

can then be used to rerun the simulation experiment. In some instances, collecting input data may be challenging, costly, or impractical, and hence knowing which input model data will reduce input uncertainty efficiently is invaluable information. Usually it is assumed that some input data has been collected, and existing methodologies can then be used to inform a data collection scheme to reduce input uncertainty effectively. However, in some instances, it may be that no input data has been collected at all. In these cases, guiding the initial data collection process to account for input uncertainty, as opposed to collecting data instinctively, would ensure that more insightful results and comparisons can be generated from the simulation experiment.

The second research problem we are interested in is developing an approach for comparing data collection strategies in viral load simulations. Modelling and simulation of viral diseases has become extremely pertinent in the past few years, with the potential to produce hugely impactful results. For such models, real-world data collection can be extremely difficult and expensive, especially in the case of a novel virus. Consequently, many of the input models to the simulation can carry a large amount of uncertainty, which will propagate through to the simulation outputs. In the case of modelling viral loads, input data collection may present a different challenge to the typical scenario. There may be decisions on how many individual subjects to take data from, how much data to collect from each subject, and when to collect data from each subject. Understanding how different approaches for collecting input data impact the uncertainty of results from the simulation model, prior to any actual data collection, can inform the data collection process resulting in higher fidelity simulation results.

The third research problem we are interested in is developing frequentist methods for quantifying input uncertainty for a quantile of the simulation response. Input uncertainty quantification usually assumes that the goal of the simulation experiment is to estimate the expected value of the simulation response, thus many approaches quantify the input uncertainty variance for the sample mean of the simulation outputs. Mean

performance measures mask the variability of the simulation response. Distributional properties of the response could provide both crucial and relevant information, and may help to differentiate between systems with similar mean performances. Quantiles are helpful for assessing risk, and can sometimes offer a more useful statistical description than the mean. Input uncertainty quantification for quantiles of the simulation response exist for Bayesian input modelling, but not for frequentist, therefore developing methodology from a frequentist perspective would aid more decision makers who are interested in the distributional behaviour of the simulation response. In addition to this, it would allow them to understand the uncertainty in a wider range of simulation response performance measures, facilitating more comprehensive comparisons across different systems.

The final research problem we are interested in is developing an understanding of the impact of input uncertainty within simulation analytics. Approaches within the growing area of simulation analytics frequently fit predictive machine learning models to simulation-generated data. Specifically, they utilise the sample path data generated within simulations that is typically not captured, or ignored in standard simulation output analysis. The predictive models can be used to uncover information about the simulated system, whilst also enabling real-time predictions to aid system control. If these approaches use input models estimated from real-world data to drive the simulation, and do not acknowledge or seek to understand the impact of uncertain input models, then overly assertive inferences may be drawn from the predictive models. Therefore, studying how the performance of a predictive model may change with respect to the uncertain input models provides useful information to assist with decision-making.

1.4 Thesis Outline

The remainder of this thesis is structured as follows. In Chapter 2, we describe the input uncertainty problem and introduce notation to define input uncertainty. We provide a thorough literature review on techniques and approaches for input uncertainty quantification. This is followed by two shorter reviews, one on methodologies for reducing input uncertainty and one on the emerging topic of simulation analytics.

Chapter 3 considers the problem of guiding an initial collection of input data in order to minimise input uncertainty. We present a new breakdown of the Taylor series approximation to input uncertainty (Cheng and Holland, 1997), which utilises the proportions in which input data is allocated to each input distribution. Using this breakdown, we solve two optimisation problems to find the optimal proportions of input data such that input uncertainty is minimised. The solutions to these two problems form part of a two stage algorithm, which aims to hone in towards an optimal collection of input data. This is the first study to consider input uncertainty at the initial data collection stage.

In Chapter 4, we develop a simulation model of viral load profiles, which can be used to estimate the performance of different testing policies. We compare strategies for input data collection in terms of the input uncertainty passed to different simulation outputs, and we run an experiment motivated by the COVID-19 pandemic, comparing three data collection strategies across two different sets of testing policies.

Chapter 5 considers the problem of quantifying input uncertainty for a quantile of the simulation response, in contrast to the usual input uncertainty problem, which considers the mean. We discuss how the problems differ, before describing how two existing input uncertainty quantification techniques, the bootstrap resampling approach described in Nelson (2013) (Section 7.2) and the Taylor series approximation of Cheng and Holland (1997), can be adapted for the case of quantiles.

In Chapter 6 we consider how a predictive model fit to simulation data generated by

estimated input parameters may not perform as anticipated on the real-world system. We use hypothesis testing and a designed experiment to look for changes in the performance of a predictive model, when the model is applied to simulation data generated by different input parameter values. We also consider the sensitivity of the predictive model performance with respect to the input parameters.

Finally, in Chapter 7 we conclude with a summary of the contributions made in this thesis, a discussion of the generalisations, limitations, and implications for practice, and identify a couple of areas for further work.

Chapter 2

Background and Literature Review

In this chapter, we introduce notation to describe the input uncertainty problem and provide a formal definition of input uncertainty. We briefly discuss some illustrative examples of input uncertainty, and reference literature where in-depth reviews can be found. We conduct a literature review of techniques for quantifying input uncertainty, and also review some methods that consider slight variations of the initially defined input uncertainty problem. We conduct a literature review on methodologies for reducing input uncertainty, before finally discussing the emerging integration of simulation models with machine learning methods, and reviewing the relevant literature here which falls under the recently coined term of simulation analytics. We end with a brief conclusion.

2.1 Input Uncertainty

In stochastic simulation, input models are specified to characterise the behaviour of the random procedures in the system. Examples of such random procedures include service times of customers in a restaurant, or the time to failure of machines in a manufacturing plant. Often input models are represented by some probability distributions (e.g. exponential, log-normal), and the practice of selecting and fitting these is known as input modelling; see Biller and Gunes (2010) for an introduction. When real-world

data is available, the probability distributions can be fit to the samples of data, or the samples of data themselves may be used as empirical distributions. Since the samples of data are necessarily finite, there is no guarantee that the input models fitted to the data are completely representative of the random procedures of the real-world system. This unknown difference introduces a source of error into the simulation model that propagates through to the outputs which is broadly referred to as *input uncertainty* or *input model uncertainty*.

Consider a simulation model that has a single input model G. Let the correct, but unknown, input model be denoted by G^0. In a parametric framework, G may represent the parameter(s) of a specific distribution, whilst in a nonparametric framework, G may denote a whole probability distribution. Suppose that data $Z_1, Z_2, \ldots, Z_m$, is collected from G^0, and used to estimate the fitted input model $\hat{G}$, which drives the simulation model. We can write the output of replication j from the simulation as

$$Y_j(\hat{G}) = \eta(\hat{G}) + \epsilon_j(\hat{G}),$$

where $\eta(\hat{G}) = \mathrm{E}[Y_j(\hat{G})]$ is the expected simulation response depending on the fitted input model, and $\epsilon_j(\hat{G})$ is a random variable with mean 0 representing stochastic noise.

Frequently, the goal of the simulation experiment is to estimate $\eta(G^0)$, the expected simulation response given the true input model. Since G^0 is unknown, this can be done by running a nominal experiment consisting of n replications driven by $\hat{G}$, and taking the sample mean of the simulation outputs

$$\begin{aligned} \bar{Y}(\hat{G}) &= \frac{1}{n}\sum_{j=1}^{n} Y_j(\hat{G}), \\ &= \frac{1}{n}\sum_{j=1}^{n} \left(\eta(\hat{G}) + \epsilon_j(\hat{G})\right), \\ &= \eta(\hat{G}) + \frac{1}{n}\sum_{j=1}^{n} \epsilon_j(\hat{G}). \end{aligned} \tag{2.1.1}$$

As the number of replications n increases, the second term in (2.1.1), which represents the error due to stochastic noise, tends towards 0. Applying the law of total variance to the estimator, we get

$$\text{Var}[\bar{Y}(\hat{G})] = \text{E}[\text{Var}(\bar{Y}(\hat{G})|\hat{G})] + \text{Var}[\text{E}(\bar{Y}(\hat{G})|\hat{G})]. \tag{2.1.2}$$

The first term in (2.1.2) measures the stochastic uncertainty associated with the point estimate. This can be easily estimated by dividing the sample variance of the simulation outputs by the number of replications. The second term in (2.1.2) is defined to be input uncertainty σ_I^2, and this simplifies to

$$\sigma_I^2 = \text{Var}[\eta(\hat{G})]. \tag{2.1.3}$$

Input uncertainty, as defined in (2.1.3), measures the uncertainty in the system mean due to having estimated the input model. This is not straightforward to estimate for two reasons. Firstly, the response surface $\eta(\cdot)$ is usually unknown. Secondly, the uncertainty around the estimated input model may not always be easily calculated. Input uncertainty depends on some complex interaction between the structure of the simulation model, and the sample of data used to estimate the input model. Unlike stochastic uncertainty, input uncertainty cannot be reduced by running extra simulation replications, but it can be reduced through the collection of additional input data or via alternative input modelling. Typically, simulation experiments beyond the nominal experiment, known as diagnostic experiments, are required to estimate input uncertainty. To motivate an interest in studying input uncertainty, we consider some examples where the problem has been illustrated in the literature.

A motivating example of input uncertainty can be found in Barton (2012). A simple $M/M/1/k$ queueing model is considered, where the arrival and service rates are estimated using 500 data points each. The queueing model is simulated to ensure that

stochastic uncertainty is close to 0, so that the variability in the simulation output predominantly due to the estimated input models can be studied. When input uncertainty is ignored, there is a greater than 25% chance that the simulation estimator of the mean number of customers in the system has an error greater than 25%.

Further to this, Corlu et al. (2020) present some examples using an $M/M/1$ queueing model. They show that the expected time a customer spends in the system can exhibit significant variability, even when stochastic uncertainty is negligible. The subsequent ramifications of this on a simulation optimisation problem illustrate that ignoring input uncertainty can lead to a so-called optimal solution that has poor performance. They also show that increasing the number of replications, whilst ignoring input uncertainty, can lead to confidence intervals which exclude the true performance measure.

An analytical example of input uncertainty can be found in Nelson (2013) (Section 7.2.1). The variance of the simulation estimator is mathematically derived when estimating the mean number of customers in an $M/M/\infty$ queue via stochastic simulation. The analytical variance consists of two terms, with one term depending on the number of replications, and the other depending on the number of real-world observations used to estimate the input models. This derivation highlights the distinguishing features of the two terms, stochastic uncertainty and input uncertainty, that comprise the total variance of the simulation estimator. The author notes that this kind of analytical derivation is not possible for realistic simulation problems.

An introduction to input uncertainty in stochastic simulation can be found in Nelson (2013) (Section 7.2), along with the analytical example discussed above. For further descriptions and illustrations of input uncertainty, see Henderson (2003), Barton (2012), Song et al. (2014), and Lam (2016). These also provide overviews of techniques for quantifying input uncertainty, which we will cover in more detail in Section 2.2, whilst the latter also includes some discussion of simulation optimisation techniques that account for input model uncertainty. A more recent review conducted by Corlu

et al. (2020) provides a classification of major research streams on input uncertainty in stochastic simulation, with a particular focus on newer developments. This also includes a literature review on applications of input uncertainty quantification, and discusses some areas for future research. Another summary of newer developments for simulation output analysis and optimisation under input uncertainty can be found in Barton et al. (2022).

2.2 Quantifying Input Uncertainty

Broadly speaking, input uncertainty quantification techniques focus on either estimating the input uncertainty term (2.1.3) extrapolated for the case of multiple input models, or constructing a confidence/credible interval for the mean simulation response that accounts for this term. The specific representation of input uncertainty and the confidence/credible interval will differ depending on whether input modelling is done in a frequentist or Bayesian way. Approaches for quantifying input uncertainty can be distinguished by whether they assume Bayesian or frequentist input modelling, whether they utilise direct simulation or a metamodel to propagate uncertainty, and what characteristics of input models they can be applied to. Here, we conduct a literature review on methodologies for quantifying input uncertainty.

Barton and Schruben (1993) present two approaches for studying the impact of input uncertainty when using an empirical distribution to drive the simulation model. The first, bootstrap resampling, utilises the bootstrap technique of Efron and Tibshirani (1986), to mimic the effect of having multiple samples of data. This involves sampling the input data, with replacement, in order to compute a new empirical distribution with which to run each simulation replication. The second, uniform resampling, uses samples from a standard uniform distribution to induce random changes to the increments in the empirical distribution which are used to run each simulation replication.

Further information on these approaches can be found in Barton and Schruben (2001), where direct resampling is also discussed. The premise of direct resampling is to divide the input data into subsets, and use the empirical distribution from each subset to run replications. The outputs from each of these three approaches can be used to construct percentile confidence intervals that account for input uncertainty. The bootstrap resampling method of Barton and Schruben (1993) is extended for the case of parametric input models in Cheng (1995).

Nelson (2013) (Section 7.2) describes how simulating multiple replications with each bootstrap sample can induce a random-effects model. Using a simplifying approximation that stochastic uncertainty is independent of the fitted input models allows for the random-effects model to provide an equation for approximating input uncertainty. Ankenman and Nelson (2012) use a similar approach, however they focus on generating a point estimate and confidence interval for the ratio of stochastic uncertainty to input uncertainty. They also develop a follow-up analysis, which can be used to identify the contributions made to input uncertainty by the input models. This is useful information, since input models that contribute more than their fair share can potentially be targeted for additional data collection. We will provide more detail on this approach, and other methodologies related to this topic, in Section 2.3.

The two-layer sampling required to estimate input uncertainty via bootstrapping can result in computationally expensive simulation experiments. Lam and Qian (2018b) propose a subsampling method to reduce the computational requirements, within a nonparametric framework. The empirical distributions used to drive the simulation are re-estimated by drawing subsamples from the input data. Input uncertainty can then be estimated using less simulation effort, by scaling the results with a multiplicative factor that depends on the size of the subsample. Further work is presented in Lam and Qian (2022), which provides theoretical results on the optimal choice of subsample size and allocation of bootstraps and replications.

An alternative approach for reducing the computational load of bootstrapping is the cheap bootstrap approach presented in Lam (2022). Unlike the aforementioned subsampling approach, this method avoids the requirement for a tuning parameter. This method reduces the need for a large amount of bootstrap resamples, by utilising the independence between the original simulation outputs and the simulation outputs generated by the bootstrap resampled input models. A confidence interval is constructed using an adjusted version of the sample variance of the bootstrapped estimates and a critical value calibrated via the t-distribution.

Barton et al. (2018) point out that in the nonparametric framework, the direct bootstrap resampling confidence intervals from Barton and Schruben (2001) can result in overcoverage. This seems to be particularly problematic when stochastic uncertainty is relatively large. They propose two new approaches for constructing confidence intervals that account for both stochastic and input uncertainty in terms of the correct coverage. One approach utilises a shrinkage operation to reduce the variance of the bootstrap samples, whilst the other makes use of a hierarchical bootstrap to remove the bias in the quantile estimates used to construct the intervals.

Another method for constructing confidence intervals that account for input uncertainty in the nonparametric framework can be found in Lam and Qian (2016). The confidence interval bounds are found by solving a pair of distributionally robust optimisation problems, with asymptotic guarantees verified via the empirical likelihood method. This optimisation-based approach is shown to generate slightly narrower and more stable confidence intervals than bootstrap resampling, whilst maintaining similar coverage probabilities.

Glynn and Lam (2018) also consider constructing confidence intervals that account for both stochastic and input uncertainty. They apply a technique known as sectioning to the input data. This involves dividing the input data up into smaller sections of equal size, and using these to drive sets of simulation replications. Selecting the number and

size of sections presents a trade-off between statistical validity and half-width properties of the confidence intervals. This approach requires few structural assumptions and is applicable within both the parametric and nonparametric framework.

Feng and Song (2019) develop a method to quantify input uncertainty using green simulation, so-called because it involves reusing simulation outputs from previous experiments to assist with future ones. In particular, they utilise the likelihood ratio method on sample path data to estimate performance measures for each set of bootstrapped input models, using only a single set of simulation replications. This can significantly reduce the computational requirements to compute input uncertainty, compared to the direct bootstrap resampling approach, however there are limitations. In some instances, the likelihood ratio calculations themselves can present a hefty computational burden.

Barton et al. (2010) introduce a framework for quantifying input uncertainty known as metamodel-assisted bootstrapping. Here, the premise is to create a metamodel of the expected value of the simulation response as a function of the input models, typically via some experimental design. Bootstrap samples are generated and fed into the metamodel to provide estimators to the simulation response, and these estimators are used to construct a percentile interval that accounts for both stochastic and input uncertainty. One advantage of using a metamodel, is that unlike direct simulation, the output can be a smooth function of the input data, which is a requirement for bootstrap convergence. They provide examples using parametric input models and a stochastic kriging metamodel. Extensions to this approach are presented in Barton et al. (2014), where they investigate which experimental designs are effective for fitting the stochastic kriging metamodel, when using general input distributions with multiple parameters.

Song and Nelson (2013) point out that the follow-up analysis presented in Ankenman and Nelson (2012) is in no sense quick, since it requires multiple additional simulation experiments beyond the one used to quantify input uncertainty. To combat this, they present an approach that provides an estimate of input uncertainty, as well as the

contributions made to input uncertainty by each input model, in one single experiment. Furthermore, their approach also provides a measure of sensitivity, which approximates the reduction in input uncertainty due to additional input data. They use a metamodel-assisted bootstrapping approach to quantify input uncertainty, where the metamodel is a linear combination of the means and variances of each input model. Bootstrapping is used to estimate the metamodel parameters, rather than a designed experiment. Although the bootstrapping approach utilises empirical cumulative distributions, the nominal experiment may be conducted with any kind of fitted distributions. A detailed version of this approach along with further experimental results can be found in Song and Nelson (2015).

Within the parametric framework, Cheng (1994) uses a first-order Taylor series expansion to provide an approximation to input uncertainty. This approximation, which expands the expectation of the simulation output at the estimated parameters as a Taylor series about the true parameters, is described in more depth in Cheng and Holland (1997). The estimator combines the parameter covariance matrix and the gradient of the expected simulation output with respect to the input parameters. The components of the latter are often referred to as the sensitivity coefficients. If maximum likelihood estimation is used to estimate the parameters, then the parameter covariance matrix can be approximated by the inverse observed Fisher information. Cheng and Holland (1997) describe an approach for estimating the gradient known as the delta method, so-called as it makes simulation runs using parameter values perturbed by a small displacement value delta. However, this method grows linearly with the number of input parameters, and can thus become computationally expensive very quickly.

The delta method is modified in Cheng and Holland (1998), so that the majority of computational effort is applied to just two parameter settings, following an initial application of the delta method. This is described as the delta/two-point method, and makes gradient estimation much more efficient and largely independent of the number

of input parameters. They also present a method known as the simplified two-point method, which removes the delta method step and makes all simulation runs at just two parameter settings. This provides an estimate for the upper bound of input uncertainty.

Cheng and Holland (2004) describe how estimates of input uncertainty and stochastic uncertainty from the delta method and the delta/two-point method can be used to construct confidence intervals for the expected simulation response using asymptotic normality theory. They also present an adaptation of the simplified two-point method known as the direct two-point method, which again uses just two parameter settings. However, this method can directly be used to generate a conservative confidence interval estimate, with the advantage of using limited computational resources relative to the other methods.

The Taylor series approximation requires an estimate of the gradient, however both the delta and delta/two-point method require additional simulation replications beyond the nominal experiment. Furthermore, they also require specification of the delta parameter, which presents a bias-variance trade-off. Additionally, the so-called quick methods developed by Ankenman and Nelson (2012) and Song and Nelson (2013) still require diagnostic experiments. To combat this, Lin et al. (2015) develop a single-experiment method for estimating the gradient that requires no further simulation replications beyond those from the nominal experiment.

They do so by exploiting the gradient estimation method of Wieland and Schmeiser (2006). Applying least-squares regression to the internal parameter estimates (computed using the random variates generated within replications) and the simulation outputs, returns a model whose coefficients provide an estimator to the gradient without the need for any diagnostic experiments. Moreover, Lin et al. (2015) point out that the Taylor series approximation also generates the approximate contribution to input uncertainty made by each input model. When input uncertainty is considered to be problematic and additional real-world data can be collected, these measures are

particularly invaluable.

Lam and Qian (2019) apply the delta method approximation in the nonparametric framework. The gradient is estimated via random perturbation, bearing some similarity to the two-point method of Cheng and Holland (1998) and the regression approach of Wieland and Schmeiser (2006). Since in this case the input models consist of entire distribution functions, the gradient is measured using the influence function and the problem grows proportionally to the size of the input data. We now turn our attention to some Bayesian methods.

The Bayesian model averaging (BMA) approach, developed by Chick (2001), accounts for not only parameter uncertainty, but also structural uncertainty about the correct choice of input distributions. For each replication, the input distribution and parameters are randomly sampled from a Bayesian model average, consisting of all the candidate distributions weighted according to their posterior probabilities based on historical data collection. The distribution and parameter uncertainty is then averaged out across the simulation results. This method is in contrast to frequentist approaches, which typically utilise a single choice of input models that offer the best fit.

Although the approach described considers real-world data, it is also valid when no such data is available, resulting in a sensitivity analysis for the output. A potential concern when taking a Bayesian approach is the specification of prior distributions. Chick (2001) provides several methods for assisting with this, including a moment matching approach, which ties in with subjective assessments of input distributions that are usually required for simulation experiments anyway. One particular challenge of the BMA approach is the computational effort required to estimate the posterior distributions, which often requires the use of Markov chain Monte Carlo.

Zouaoui and Wilson (2004) describe an alternative BMA approach, building upon their previous work on parameter uncertainty (Zouaoui and Wilson, 2001b, 2003) and input model and parameter uncertainty (Zouaoui and Wilson, 2001a). For each candi-

date model, they run replications with parameters generated from the posterior input parameter distributions. They allow for flexibility in the number of parameters generated for each candidate model, since the impact of the parameters on the variability of the simulation response is likely to vary across the models. The final posterior response is given by an average of the mean responses for each candidate model, weighted according to their posterior probabilities.

The approach of Zouaoui and Wilson (2004) is designed to offer three improvements over the BMA method of Chick (2001). Firstly, it provides a decomposition of the variance into the three sources of uncertainty: stochastic uncertainty, model uncertainty, and parameter uncertainty. Secondly, it guarantees to sample from each of the candidate models, which may not occur with the BMA method of Chick (2001) if a model has a small posterior probability and the number of simulation replications is limited. Finally, should any additional candidate models need to be considered, then the calculations can easily be updated by running additional replications. Using the approach of Chick (2001), all existing simulation replications would become redundant, and thus the whole analysis would need to be repeated using updated posterior distributions.

Both Chick (2001) and Zouaoui and Wilson (2003) consider an interval estimate for the expected simulation response, averaged over the uncertain input parameters. As an alternative, Xie et al. (2014a) provide a credible interval for the mean response at the true parameters, which they believe is more desirable to simulation practitioners. To do this, they develop a fully Bayesian framework that utilises a Gaussian process metamodel for the simulation response, as opposed to the direct simulation approaches of Chick (2001) and Zouaoui and Wilson (2003).

Biller and Corlu (2011) note that although the BMA method developed by Chick (2001) can account for dependence amongst the input models, to implement this requires a Bayesian model for simulations with multiple correlated inputs. They use a flexible normal-to-anything (NORTA) distribution to characterise the behaviour of the

correlated inputs, and develop a Bayesian model to sample NORTA parameters from their posterior density functions. This model is then incorporated into the simulation replication algorithm from Chick (2001).

Xie et al. (2014b) also consider the input uncertainty problem with dependent input models. Motivated by situations which require a complex simulation model and a restricted computational budget for replications, they utilise a stochastic kriging metamodel to propagate uncertainty, in contrast to the direct simulation approach of Biller and Corlu (2011). They focus on input models characterised by marginal distributions, and where dependence is measured by Spearman rank correlations. To account for the dependent input models, they also use NORTA distributions, applying bootstrapping to quantify the estimation error of the distributions and dependency measures. An extended version of this method can be found in Xie et al. (2016), where dependence measured by product-moment correlations is also studied.

The methods described above all assume the input distributions are stationary, however in reality, non-stationarity is a reasonable assumption. Morgan et al. (2016) consider input uncertainty quantification for non-stationary input distributions. In particular, they consider simulation models driven by piecewise-constant non-stationary Poisson arrival processes, since these provide flexibility, can be easily fit to count data, and are often included in simulation software. They extend both the Taylor series approximation of Cheng and Holland (1997), and the mean-variance effects metamodel approach of Song and Nelson (2015), to account for these specific arrival processes.

Yi and Xie (2017) study the input uncertainty quantification problem from a budget allocation perspective. Specifically, they focus on efficiently constructing a percentile confidence interval within a nonparametric framework for cases when simulation replications are computationally expensive. They use bootstrapping to approximate the distributional uncertainty and direct simulation to propagate this uncertainty to the response. Given a simulation budget, they develop a sequential approach to allocate

more replications to the important bootstrap samples that contribute the most to the percentile estimation, as opposed to existing methods which allocate simulation replications equally across bootstrap samples.

Zhou and Liu (2018) propose a method for input uncertainty quantification designed to work efficiently for online data, that is, data which arrives sequentially in time. All existing methods typically consider input data in a batch, that is, data which is collected and available all at once. Though these methods could be repeated as new data arrives, this would clearly be inefficient, particularly when simulation replications are computationally expensive. Zhou and Liu (2018) assume parametric input models, and take a Bayesian approach to estimating the input parameters and quantifying input uncertainty. Importance sampling is used to transform existing simulation outputs so that they can be reused under the new posterior distribution each time a data point arrives. This is somewhat related to the green simulation idea utilised in Feng and Song (2019).

Zhu et al. (2020) extend the standard input uncertainty quantification problem to provide rigorous risk quantification of the mean simulation response under input uncertainty. They propose a nested Monte Carlo approach to estimate common risk measures such as value at risk (VaR) and conditional value at risk (CVaR), frequently used in financial portfolio management. These allow for assessment of extreme mean responses under the uncertain input models, which may be especially useful when the decisions made based on the simulation analysis are irreversible.

Xie et al. (2018) consider quantifying the uncertainty of percentiles from the simulation response, as opposed to traditional approaches which focus on the mean simulation response. Building upon the Bayesian framework from Xie et al. (2014a), they construct a rigorous distributional metamodel, motivated by quantile kriging, that models the response of a sequence of percentiles. The metamodel is used to propagate the parameter uncertainty, quantified via the posterior distributions of input parameters, to the per-

centile outputs. This approach produces credible intervals that quantify the overall uncertainty of percentiles of the simulation response, and a variance decomposition allows for the input uncertainty and stochastic uncertainty to be separated.

Chen et al. (2022) also consider input uncertainty from a distributional perspective. They propose a method to construct simultaneous confidence bands for the entire distribution of the simulation response, rather than just the mean. A Brownian bridge is used to represent stochastic uncertainty, whilst a mean-zero Gaussian process is used to represent input uncertainty. The covariance of the Gaussian process is controlled by an influence function, which can be thought of as the gradient of the distribution of the simulation response, with respect to the input parameters, and a subsampling approach is used to estimate the covariance function.

Comprehensive tables classifying the literature on input uncertainty quantification can be found in Corlu et al. (2020). The literature is split into frequentist and Bayesian approaches, and sorted based on the characteristics of the input models (the dependence structure and distributional form), and the approach used to propagate the input model error (direct or metamodel).

2.3 Reducing Input Uncertainty

As we have seen, some methods for quantifying input uncertainty (Song and Nelson, 2015; Cheng and Holland, 1997) naturally provide an approximation of the contributions made to input uncertainty by each input model. Alternatively, other methods (Ankenman and Nelson, 2012) prescribe additional diagnostic experiments, beyond the one used to quantify input uncertainty, to reveal similar contribution information. Generally speaking, the motivation for obtaining any contribution style measure, is so that the input models with the largest contributions can be targeted for additional data collection in an attempt to effectively reduce input uncertainty. Here, we conduct a

literature review on methodologies for reducing input uncertainty.

More formally, Ng and Chick (2001) specifically consider the problem of allocating resources for extra data collection to effectively reduce input uncertainty. Parameter uncertainty is modelled via a Bayesian approach, and the posterior distribution of each parameter is approximated using asymptotic normality properties. These are updated via the expected information of additional observations to make inference about the value of additional real-world data. To optimise the allocation of the real-world data budget, they require an approximation to the gradient of the response function. They utilise a Bayesian model average over a set of linear metamodels to approximate the input-output relationship and thus the gradient, however any technique for gradient estimation can be applied here.

This work is continued in Ng and Chick (2006), where results are extended to a broader class of input distributions. They also consider the additional scenario where the response surface is unknown, and the asymptotic approximation may not be sufficient. Here, their sampling allocation shows whether it is more meaningful to run additional simulation replications to improve the gradient estimate, or whether to collect additional data to reduce parameter uncertainty.

Freimer and Schruben (2002) describe two approaches for collecting data to reduce the impact of input uncertainty for simulations driven by parametric input distributions. Their approaches utilise analysis of variance (ANOVA) based on a fixed effects and a random effects model. The fixed effects approach uses replications driven by the confidence intervals bounds of each parameter and tests whether there is a significant difference in the expected simulation response. If there is a significant difference, then more input data must be collected to reduce the width of the confidence interval, and thus reduce the effect of the parameter. The random effects approach does not require confidence intervals. Instead, bootstrap samples are generated and the random effects model is used to test whether the expected simulation response is constant over

the different bootstrap samples. If this is not the case, then additional observations should be collected. The random effects approach has the advantage of not requiring calculation of joint confidence intervals, however it is more computationally intensive than the fixed effects approach.

Ankenman and Nelson (2012) define a measure to describe the fair share of input uncertainty contributed by each input model, with the aim of identifying input models that contribute more than their fair share. In an approach motivated by sequential bifurcation, a technique used to screen factors, they test the contributions made to input uncertainty by groups of input distributions. The input distributions not in the group are treated as fixed, whilst those in the group are bootstrapped, to capture the input uncertainty solely due to the group. The resulting confidence interval can be used to gauge whether any distributions in the group are contributing more than their fair share, and if so, the group can be split and the subgroups can undergo the same procedure.

Song and Nelson (2015) model the expected simulation response as a function of the mean and variance of each input model. This approach provides the contributions made to input uncertainty by each input model and sample size sensitivity measures. These approximate the reduction in input uncertainty should an additional real-world observation be collected to estimate each input model. To effectively reduce input uncertainty, the input models with the most negative sensitivity measures should be targeted for additional data collection, subject to feasibility and costs. Note that the contribution and sensitivity measures do not necessarily coincide. That is, input models with large contributions to input uncertainty may not be the most sensitive, particularly if they are estimated from a relatively large sample of observations.

Xie et al. (2019) consider guiding input data collection for their previously developed Bayesian framework (Xie et al., 2014a). Here, parameter uncertainty is quantified by the posterior distribution of input parameters, and a Gaussian process metamodel is

used to propagate the parameter uncertainty to the expected simulation response. They adapt the functional ANOVA approach of Oakley and O'Hagan (2004), to provide a global metamodel-assisted sensitivity analysis for stochastic simulation models. Using this, they are able to estimate the contribution to input uncertainty from each input model, and also predict the value in collecting additional data for each input model. They avoid the assumption that the response surface can be modelled in a linear form, unlike the methods of both Ng and Chick (2006) and Song and Nelson (2015), that utilise potentially restrictive linear metamodels.

Instead of collecting additional input data, simulation users can also consider reducing input uncertainty via better input modelling. Nelson et al. (2021) use frequentist model averaging as a method to propagate less uncertainty to the simulation output, weighting a set of candidate parametric distributions using cross-validation. This approach is particularly applicable in situations where no parametric distribution seems appropriate for the input data. Morgan et al. (2023) develop a spline-based method for modelling nonhomogenous Poisson processes. The spline-based method used to model the rate function is flexible, robust to non-Poisson data, and is shown to outperform other recent methods in terms of recovering the true rate function of such processes. Consequently, their approach generally passes less input uncertainty variance to simulation outputs compared to the alternative methods.

Recently, there has been a focus on solving traditional simulation optimisation problems under input uncertainty, see Lam (2016) and Corlu et al. (2020) for discussion and reviews. Within the literature on simulation optimisation under input uncertainty, a few authors have considered the trade-off between running simulation replications to reduce stochastic uncertainty, and collecting additional input data to reduce input uncertainty. We briefly review some of these works.

Xu et al. (2020) propose a general framework to study the joint resource allocation problem for simulation replications and input data collection. Their approach permits

closed-form solutions for resource allocation that maximise the asymptotic probability of correctly selecting the system with the best performance. In their approach, they exploit common random numbers within the simulations and correlations between input data to reduce costs.

Ungredda et al. (2022) propose a sequential algorithm, that in each iteration decides whether to collect additional input data or run simulation replications. They use a Gaussian process metamodel for the expected simulation response and take a Bayesian approach to modelling the input parameters. The impact of additional input or simulation data is quantified using a value of information criteria. This measures the future expectation of the existing optimal solution, and actions are chosen to maximise this value.

Kim and Song (2022) consider a Bayesian ranking and selection problem under input uncertainty, and investigate optimal sampling ratios from the input data sources. Specifically, they focus on the optimal estimator being the most probable best, defined as the solution with the largest posterior probability of being optimal. They then devise a sequential approach for acquiring input data that aims to optimise the convergence of the posterior preference of the most probable best, based on the sampling ratios amongst the input data sources.

2.4 Simulation Analytics

The recent field of simulation analytics, proposed by Nelson (2016), aims to harness the dynamic data generated within stochastic simulations to develop a more sophisticated understanding of the system. Typically, the vast quantities of dynamic data generated within simulation replications are not retained, and additionally, high-level, static summaries are often produced, which hinders the ability to evaluate the risk of, or predict, the system behaviour. Nowadays, technology allows for the large quantities of dynamic

data to be stored, and these have the ability to form a synthetic transactional data set on which modern data analytic techniques can be applied. A conceptual framework and examples of the increasingly common integration of simulation models and machine learning methods can be found in von Rueden et al. (2020).

Some proposed objectives for simulation analytics include producing dynamic conditional statements about the relationship of system state to outputs, generating dynamic distributional statements of the output behaviour at specific points in time, and developing an understanding of why certain system designs behave differently. Simulation analytics could be used to develop a model that can provide real-time predictions, and knowing the key drivers behind this model may help with managing and controlling the system. Nelson (2016) notes the need for methods to consider, and possibly exploit, the differences between simulation-generated and real-world data, as well address the differences in the problem context that mean a simulation model is required over a traditional field analysis. Here, we review some applications and methodologies within the field of simulation analytics.

Ouyang and Nelson (2017) present proof-of-concept results for predictive models in the context of network queueing systems. Motivated by system control (e.g. managing congestion), they use the dynamic samples paths generated within simulation replications to develop a regression model that can predict some aspect of the future behaviour of the queue, based on the given state and time information. They propose a two-step method, to solve the problem of handling both the state and time of the simulation as predictor variables. The method is able to predict the probability of the system state belonging to a certain subset at a future time, given the currently observed state information. These subsets might represent events such as the queue being blocked or excessive congestion in the system. Within the model these might be based on, for example, whether the total number of customers at some collection of nodes exceeds some threshold, or whether all the servers are busy at another particular set of nodes.

Jiang et al. (2020) utilise logistic regression to enable online risk monitoring and classification of a financial portfolio. The existing literature generally considers estimation of the portfolio risk measures once, i.e. at a single point in time. More pertinently, these approaches usually estimate an unconditional risk measure, only taking into account the current values of the underlying risk factors, and using static risk limits. Instead, Jiang et al. (2020) take the simulation analytics approach. By using the sample path simulation data, they are able to estimate the same exceedance probability, but conditional on the underlying risk factors at any time. This means the portfolio risk can be classified at any given time, rather than just from the initial time point. They take advantage of their knowledge of the simulation model, to perturb the simulation sample paths and improve the quality of the risk estimators and classifiers, especially for the earlier time periods. They also consider how to incorporate data from additional simulation experiments into the analysis to improve the risk estimators and classifiers.

Baldwa et al. (2020) combine simulation and machine learning to make real-time delay predictions for a neurosurgery ward. Such predictions are typically made using analytical delay predictors or by using a data-driven approach whereby machine learning methods are applied to the data logs from the queueing system. However, in this instance, the system is not mathematically tractable, hence analytical predictors cannot be derived, and the admissions data does not capture sufficient system state information to make accurate predictions. Consequently, they develop a discrete-event simulation model of the admission and patient stay procedures of the ward. Using the steady-state simulation data, they train and validate machine learning models to predict whether newly arriving patients will be admitted to the ward, and their waiting time before admission. In practice, these models can then be used to make real-time predictions as patients arrive to the ward, provided the hospital administration tracks the system state variables required as input to make the predictions.

Laidler et al. (2020) apply a k-nearest neighbours classification model to data gen-

erated from simulating a wafer fabrication facility. The simulation model, which represents the manufacturing process of semiconductor wafers, is a network of queues complicated by different product types, repeat visits, and batch processing. The classification model is trained to identify whether a wafer will be completed early or late with respect to its due date, based on the system state when it arrives. The motivation for using the k-nearest neighbours approach is the flexibility, since the model is nonparametric, and the dynamic sample path data is likely to exhibit non-stationarity that would be difficult to capture with a parametric model. Although the main focus is on using metric learning to measure the similarity between state observations, the application of the method is motivated by system control and the ability to make real-time predictions. One particular benefit of the approach is the interpretability, since the matrix produced by metric learning informs them which state variables are most significant in determining the binary indicator of system performance.

Dantas et al. (2022) develop constructive simulations of beyond visual range air combat, to estimate the most effective moment for launching missiles. The input variables that feed into the simulations, such as altitudes, speeds, and distances, are sampled via Latin hypercube sampling from some prespecified intervals. The motivation for the Latin hypercube sampling approach is to fill the sample space in a better manner than a purely random process, such as Monte Carlo sampling. The simulations terminate after all aircraft have been destroyed or until a simulation time of 30 minutes has been reached. The results from the simulation go through a variable selection process, before being used to train various machine learning models. Since the data set is imbalanced, they apply resampling techniques to balance the class distributions, and retrain the models on the newly balanced datasets. They find that the resampling techniques improve certain model metrics, whilst downgrading others. The air combat data from the simulations, which are tricky to obtain from the real-world, are able to help develop a decision support tool which can assist pilots in real-world air combat situations.

2.5 Conclusion

We end this chapter with a brief conclusion to highlight some research gaps in the literature and how they align with our research problems. Within the literature on reducing input uncertainty, all existing approaches suppose an initial collection of input data. Our first research problem seeks to address a gap in the literature by considering how to account for input uncertainty prior to any initial data collection. A method for this is developed in Chapter 3. Considering input uncertainty quantification for a quantile is a recently researched problem, as such, there exists no approaches within the literature for quantifying the input uncertainty of a quantile under frequentist input modelling. Combating this is the goal of our third research problem, and is addressed in Chapter 5. Within the literature on both input uncertainty and simulation analytics, there exists no area of crossover. That is, there is no investigation into the impact of input uncertainty when applying methods from simulation analytics. This forms the basis of our fourth research problem which is exploratory, and covered in Chapter 6.

Chapter 3

Guiding Data Collection to Minimise Input Uncertainty

In stochastic simulation, the input models used to drive the simulation are often estimated by collecting data from the real-world system. This can be an expensive and time-consuming process, so it would therefore be useful to have some guidance on how much data to collect for each input model. In this chapter, we propose a two stage algorithm that guides the initial data collection procedure for a simulation experiment that has a fixed data collection budget, with the objective of minimising input uncertainty in the simulation response.

The majority of content in this chapter has been published as Parmar et al. (2021b). The main point of difference is that the future research section of the paper is converted to an additional experiments section here. This still describes the same problems that were outlined in the future research section of the initially published paper, however we include some experiments to further investigate and better illustrate these, as well as giving consideration to one extra problem that motivates an area for future work.

Note that we also update the experiment section to include a table of the parameter values and optimal proportions for the $M/M/1$ experiment, and a table of the parameter

values for the network queueing model experiment, to aid with the interpretation of both these experiments. Furthermore, in the published version, the derivation of the optimal proportions for the two optimisation problems were not included. Here we provide them in Appendix A.1 and Appendix A.2. We also include further detail on the $M/M/1$ queueing model experiment in Appendix A.3, where we show the derivation of the optimal proportions using the analytical gradient measures. Notational changes are made to maintain consistency with other chapters.

3.1 Introduction

The randomness in stochastic simulation is caused by input models, which are often represented by probability distributions or processes. When the real-world processes can be observed, samples of data can be collected and used to estimate the input models. The samples of data from which to estimate the input models are finite, and thus the input models will never be truly representative of reality. The uncertainty in the estimated input models is propagated through the simulation model, resulting in an error in the simulation response known as input uncertainty. Input uncertainty must be quantified, along with stochastic estimation error, to measure the variability around the simulation response and ensure that decisions are made with an appropriate level of confidence; Barton (2012) illustrates the significant risk of ignoring input uncertainty. A reduction in input uncertainty can be achieved by collecting additional observations from the real-world processes. One way this is done is by studying the contribution made to input uncertainty by each of the input models, and specifying how best to allocate a budget for additional data collection amongst the input models.

Here, instead of looking at additional data collection to reduce input uncertainty, we introduce the idea of guiding the initial data collection process in a manner that minimises input uncertainty. We consider the case of parametric input models and by

assuming some knowledge of what values the parameters may take, we develop a two stage algorithm that allocates observations amongst the input models with the objective of minimising input uncertainty. Collecting data in this way is likely to reduce input uncertainty, and hence the level of variability, in the simulation response compared to alternative approaches, thus increasing the level of insight that can be derived from experimental results. This will lessen the need for additional data collection in order to reduce input uncertainty and may also reduce unnecessary data collection more generally, both of which are particularly beneficial when data collection is expensive and time-consuming.

We discuss background literature in Section 3.2 and detail the existing methodology we build upon in Section 3.3. We present a new breakdown of the existing methodology in Section 3.4, which allows us to form and solve optimisation problems which minimise input uncertainty. We describe the two stage algorithm for data collection in Section 3.5 and illustrate the algorithm with some experiments in Section 3.6. We run some additional experiments in Section 3.7 and then conclude in Section 3.8.

3.2 Background

Various methods have been proposed to quantify input uncertainty, for an overview of existing techniques see Barton (2012) or Song et al. (2014). We focus on the methodology developed by Cheng and Holland (1997) for the case of parametric input distributions. Here, input model uncertainty reduces to parameter uncertainty and is modelled using a first order Taylor series expansion around the true input parameters. A recent development to this approach was made by Lin et al. (2015), who exploit the gradient estimation method of Wieland and Schmeiser (2006), to estimate input uncertainty in a single experiment. Although initially restricted to the case of stationary input distributions, further work by Morgan et al. (2016) has since extended this input un-

certainty quantification method for simulation models which utilise piecewise-constant non-stationary Poisson arrival processes.

The problem of allocating resources for extra data collection was considered by Ng and Chick (2001), who use asymptotic normality properties to approximate the posterior distribution of each parameter. By considering the expected information of additional observations and propagating uncertainty through the simulation using a linear metamodel, they provide sampling plans for further data allocation which aim to reduce input uncertainty effectively. Alternatively, Freimer and Schruben (2002) use an ANOVA test to detect whether a parameter has a significant effect on the expected simulation response as the parameter varies over its confidence interval. If the effect is significant, then more data should be collected to narrow the confidence interval until the effect is no longer significant. Finally, Song and Nelson (2015) model the expected simulation response as a function of the mean and variance of each input model, and use the sample size sensitivity of each distribution to recommend how to collect further data. These methods aim to guide data collection based on input models that have been estimated using real-world observations, however our method aims to guide data collection before any real-world observations have been collected. We now describe an existing input uncertainty quantification technique that we will utilise within our approach for guiding data collection.

3.3 Taylor Series Approximation

Consider a simulation driven by L random processes which follow known independent parametric distributions with unknown parameters. Let the unknown but true parameters be denoted by $\boldsymbol{\theta}^0 = (\theta_1^0, \ldots, \theta_q^0)$, where $q \geq L$, as some distributions may require more than one parameter. Suppose that real-world data can be collected from each input distribution and that parameters are estimated via their maximum likelihood es-

timators (MLEs). Let $\hat{\boldsymbol{\theta}} = (\hat{\theta}_1, \ldots, \hat{\theta}_q)$ denote the MLEs of the input parameters given the observed data. In this parametric setup, the simulation response can be thought of as a function of the input parameters. The output of replication j of the simulation can be denoted by

$$Y_j(\hat{\boldsymbol{\theta}}) = \eta(\hat{\boldsymbol{\theta}}) + \epsilon_j(\hat{\boldsymbol{\theta}}),$$

where $\eta(\hat{\boldsymbol{\theta}})$ is the expected value of the simulation output given the estimated parameters and ϵ is a random variable with mean 0 representing stochastic noise.

The goal of the simulation experiment is to estimate $\eta(\boldsymbol{\theta}^0)$, the expected value of the simulation output given the true input parameters. This can be estimated via the sample mean of the simulation output over n replications

$$\begin{aligned}\bar{Y}(\hat{\boldsymbol{\theta}}) &= \frac{1}{n}\sum_{j=1}^{n} Y_j(\hat{\boldsymbol{\theta}}), \\ &= \eta(\hat{\boldsymbol{\theta}}) + \frac{1}{n}\sum_{j=1}^{n} \epsilon_j(\hat{\boldsymbol{\theta}}).\end{aligned}$$

As n increases the error caused by stochastic noise tends towards 0, however the impact of $\hat{\boldsymbol{\theta}}$ on the expected simulation response is not affected by the choice of n. The variance of this estimator breaks down into two distinct terms, stochastic estimation error and input uncertainty. The former arises from the random variates generated in each replication and can be easily estimated via the sample variance. The latter measures the variability in the expected output due to having estimated the input parameters, that is

$$\sigma_I^2 = \mathrm{Var}[\eta(\hat{\boldsymbol{\theta}})].$$

Using a first-order Taylor series approximation around the true input parameters $\boldsymbol{\theta}^0$, Cheng and Holland (1997) provide the following estimate of input uncertainty

$$\sigma_I^2 \approx \nabla\eta(\boldsymbol{\theta}^0)\mathrm{Var}(\hat{\boldsymbol{\theta}})\nabla\eta(\boldsymbol{\theta}^0)^\top,$$

where $\nabla\eta(\boldsymbol{\theta}^0)$ is the gradient of the expected value of the simulation output with respect to the input parameters $\boldsymbol{\theta}$, evaluated at $\boldsymbol{\theta}^0$. This estimate of input uncertainty depends on how sensitive the simulation output is to the input parameters and how well the input parameters have been estimated. Neither of these terms are known, and so both have to be estimated. Note that this method for quantifying input uncertainty has been extended for the case of non-stationary input models by Morgan et al. (2016).

3.3.1 Variance Estimation

As the parameters are estimated via maximum likelihood estimation, the variance matrix can be approximated by

$$\widehat{\mathrm{Var}}(\hat{\boldsymbol{\theta}}) = \mathbf{I}(\hat{\boldsymbol{\theta}})^{-1},$$

the inverse Fisher information matrix evaluated at the MLEs $\hat{\boldsymbol{\theta}}$. This follows since the asymptotic distribution of the MLEs is multivariate normal with covariance matrix $\mathbf{I}(\boldsymbol{\theta}^0)^{-1}$, and $\mathbf{I}(\boldsymbol{\theta}^0)^{-1}$ can be consistently estimated by $\mathbf{I}(\hat{\boldsymbol{\theta}})^{-1}$.

3.3.2 Gradient Estimation

As the true parameters $\boldsymbol{\theta}^0$ are unknown, Cheng and Holland (1997) approximate $\nabla\eta(\hat{\boldsymbol{\theta}})$ instead of $\nabla\eta(\boldsymbol{\theta}^0)$, however the simulation effort for the method they propose increases linearly with the number of input parameters. Lin et al. (2015) improve upon this by providing a method for estimating $\nabla\eta(\hat{\boldsymbol{\theta}})$ which is independent of the number of input parameters. This method, which extends the work of Wieland and Schmeiser (2006), requires running simulation replications using the fitted input parameters and recording the simulation output along with internal parameter estimates for each replication. Internal parameter estimates are obtained using the realisations generated from the input distributions during a simulation replication, for example inter-arrival times observed within a replication can provide an internal estimate of the arrival rate. Fitting

a least-squares regression model, with the simulation output as the response variable and the internal parameter estimates as the explanatory variables, gives a regression model whose coefficients $\hat{\boldsymbol{\delta}}(\hat{\boldsymbol{\theta}})$ provide an estimator to $\nabla\eta(\hat{\boldsymbol{\theta}})$.

3.3.3 Contributions to Input Uncertainty

Input uncertainty can then be approximated by combining the estimates for the variance matrix and the gradient vector as follows

$$\sigma_I^2 \approx \hat{\boldsymbol{\delta}}(\hat{\boldsymbol{\theta}})\, \mathbf{I}(\hat{\boldsymbol{\theta}})^{-1}\, \hat{\boldsymbol{\delta}}(\hat{\boldsymbol{\theta}})^{\top}.$$

This approximation also provides us with an estimate of the contribution made to input uncertainty by each input distribution. Let $\boldsymbol{\theta}_l$ denote the parameter vector for input distribution l, note that this could be a scalar or a vector depending on the distribution. Since the input distributions are independent, the variance matrix has a block diagonal form, with elements consisting of individual variance matrices $\text{Var}[\hat{\boldsymbol{\theta}}_l]$ for each input distribution. Let $\hat{\boldsymbol{\delta}}(\boldsymbol{\theta}_l)$ denote the gradient vector for the parameters belonging to input distribution l, then

$$\sigma_I^2 \approx \sum_{l=1}^{L} \hat{\boldsymbol{\delta}}(\hat{\boldsymbol{\theta}}_l)\mathbf{I}(\hat{\boldsymbol{\theta}}_l)^{-1}\hat{\boldsymbol{\delta}}(\hat{\boldsymbol{\theta}}_l)^{\top},$$

where the l^{th} term in the sum represents the contribution made to input uncertainty by input distribution l. This breakdown can be used to show which input distributions should be targeted for further data collection in order to reduce input uncertainty, for example, see Lin et al. (2015).

3.4 Data Collection for Minimising Input Uncertainty

We now present a new breakdown of the Taylor series approximation to input uncertainty, which we propose as a tool for guiding data collection. The Fisher information for an i.i.d. sample of size m is simply m times the Fisher information for a single observation. Let m_l denote the number of observations used to estimate the parameters for input distribution l. The Fisher information matrix of $\hat{\boldsymbol{\theta}}_l$ is then given by

$$\mathbf{I}(\hat{\boldsymbol{\theta}}_l) = m_l \mathbf{I}_0(\hat{\boldsymbol{\theta}}_l),$$

where $\mathbf{I}_0$ represents the Fisher information of a single observation. Let $m = \sum_{l=1}^{L} m_l$ denote the total number of observations used to estimate all input distribution parameters. For each input distribution l, we can write $m_l = r_l m$, where $r_l \in (0, 1)$ represents the proportion of all observations which are from input distribution l, and $\sum_{l=1}^{L} r_l = 1$. Input uncertainty can then be written as

$$\sigma_I^2 \approx \frac{1}{m} \sum_{l=1}^{L} \hat{\boldsymbol{\delta}}(\hat{\boldsymbol{\theta}}_l)[r_l \mathbf{I}_0(\hat{\boldsymbol{\theta}}_l)]^{-1} \hat{\boldsymbol{\delta}}(\hat{\boldsymbol{\theta}}_l)^\top. \tag{3.4.1}$$

Initially, let us consider a set of specific parameter values $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_q)$. For these parameters, the Fisher information matrix can be calculated and the gradient vector can be estimated using the method outlined above. Plugging these into (3.4.1) would give an approximation of input uncertainty at $\boldsymbol{\theta}$ in terms of the total number of observations m, and the proportions r_l in which they are allocated to each input distribution. If $\boldsymbol{\theta}$ were the set of true input parameters, we could use this to guide data collection by finding the proportions in which to collect data such that input uncertainty is minimised. Note that the proportions which will minimise input uncertainty are invariant to the total number of observations.

We are therefore interested in solving the following optimisation problem

$$\{\min \sum_{l=1}^{L} \frac{a_l}{r_l} \quad \text{s.t.} \quad \sum_{l=1}^{L} r_l = 1 \quad \text{and} \quad r_l > 0 \text{ for } l = 1, \ldots, L\}, \tag{3.4.2}$$

where $a_l = \hat{\boldsymbol{\delta}}(\boldsymbol{\theta}_l)\mathbf{I}_0(\boldsymbol{\theta}_l)^{-1}\hat{\boldsymbol{\delta}}(\boldsymbol{\theta}_l)^\top$, and r_l are the proportions to be optimised. This problem can be converted to an inequality-constrained nonlinear programme and solved to optimality by studying the first-order KKT conditions proved by Karush (1939) and Kuhn and Tucker (1951). In Appendix A.1, we show the optimal proportions for (3.4.2) are given by

$$r_l = \sqrt{\frac{a_l}{\left(\sum_{l=1}^{L} \sqrt{a_l}\right)^2}}. \tag{3.4.3}$$

Alternatively suppose that input parameters $\hat{\boldsymbol{\theta}}$ have been fitted via a collection of real-world data and that input uncertainty has been quantified via the Taylor series approximation. If input uncertainty is a cause for concern, then it may be of interest to collect more real-world data in a manner which effectively reduces input uncertainty. This is often done by considering a sampling budget D for extra data collection. If we wish to collect data in a manner such that the overall proportions minimise input uncertainty, then there is an extra consideration to be made. Given the budget for collecting extra data and the existing data collected, there is a restriction on how small each proportion can be, that is, each proportion will have a lower bound.

We are now interested in solving the following optimisation problem

$$\{\min \sum_{l=1}^{L} \frac{a_l}{r_l} \quad \text{s.t.} \quad \sum_{l=1}^{L} r_l = 1 \quad \text{and} \quad r_l \geq b_l \text{ for } l = 1, \ldots, L\}, \tag{3.4.4}$$

where $a_l = \hat{\boldsymbol{\delta}}(\hat{\boldsymbol{\theta}}_l)\mathbf{I}_0(\hat{\boldsymbol{\theta}}_l)^{-1}\hat{\boldsymbol{\delta}}(\hat{\boldsymbol{\theta}}_l)^\top$, $b_l = m_l/(m + D)$, and r_l are the proportions to be optimised. Again, this problem can be converted to an inequality-constrained nonlinear programme and solved to optimality using first-order KKT conditions. In Appendix

A.2, we show the optimal proportions for (3.4.4) are given by finding a partition (I, J) of $\{1, \ldots, L\}$ such that

$$\text{for } l \in I: \; r_l = \sqrt{\frac{a_l}{\mu}} \geq b_l, \qquad \text{for } l \in J: \; r_l = \sqrt{\frac{a_l}{\mu - \lambda_l}} \geq b_l, \quad \lambda_l = \mu - \frac{a_l}{b_l^2} \geq 0,$$

where

$$\mu = \left(\frac{\sum_{l \in I} \sqrt{a_l}}{1 - \sum_{l \in J} b_l} \right)^2.$$

We use the solutions to these two optimisation problems to develop an algorithm that aims to guide the initial data collection process in a manner that minimises input uncertainty. Note that additional constraints could be added to either formulation to incorporate features of the data collection procedure.

3.5 Two Stage Algorithm for Data Collection

When modelling some systems, for example medical practices or manufacturing processes, collecting data to estimate the input models may be an expensive and time-consuming task. In these scenarios, one may wish to consider a strategy for data collection rather than taking some arbitrary approach. Here we introduce a two stage algorithm to guide data collection. We assume that each input parameter lies in some known interval and study how data might be optimally collected to minimise input uncertainty within these intervals. In the first stage of the algorithm, data is collected to hone in towards an optimal collection whilst relaying information about the true parameters values. In the second stage, extra data is collected to achieve the proportions that minimise input uncertainty based on the parameter values from the first stage data collection.

Suppose there has been no data collection for a system. Although we have no data from which to estimate the input parameters we shall assume that each input

parameter is known to lie in some interval, $\theta_i^0 \in [l_i, u_i]$, for $i = 1, \ldots, q$, where the lower and upper bounds, l_i and u_i, are known. For example, in a medical practice the number of patients arriving may be known to be between 15-20 per hour, but the exact arrival rate is unknown. The true input parameters could lie anywhere in this q dimensional space, and in order to collect data in a way that minimises input uncertainty, we need to understand how the optimal proportion changes for each input parameter across the space. An intuitive design that can be used to explore the input parameter space is a 2^k factorial design, which is used to study the effects of k factors each at two different levels (usually high and low) by considering every possible combination of factors and levels. Since there are q input parameters and each is known to be between a lower and upper bound, this naturally lends itself to a 2^q factorial design where each factor is an input parameter with low level l_i, and high level u_i.

At each design point we can solve (3.4.2) to find the optimal proportions in which to collect data should the parameters at that design point represent the true parameters. Computing the optimal proportions at each design point will give us an idea of the behaviour of the optimal proportion for each input parameter over the specified parameter space. Rather than studying the effects of each parameter, we are instead interested in the minimum and maximum optimal proportion across the design points for each parameter. We use these to form an approximate interval for the optimal proportion for each parameter at the true parameter values. For example, suppose that $q = 2$ so the parameter vector is $\boldsymbol{\theta} = (\theta_1, \theta_2)$. A 2^q factorial design gives $2^2 = 4$ design points which enumerate every combination of factors and levels. Suppose that a 2^2 factorial design gives the optimal proportions shown in Table 3.5.1. From this, we approximate that the optimal proportions for the true parameters will fall within the following intervals: $r_1 \in [0.3, 0.5]$ and $r_2 \in [0.5, 0.7]$.

Suppose we have a budget C for collecting observations, which is to be allocated amongst all the input distributions. In stage one, we aim to allocate as much of the

Design Point i	θ_1^i	θ_2^i	r_1^i	r_2^i
1	l_1	l_2	0.5	0.5
2	u_1	l_2	0.3	0.7
3	l_1	u_2	0.4	0.6
4	u_1	u_2	0.4	0.6

Table 3.5.1: Example optimal proportions for a 2^2 factorial design.

budget as possible without ruling out the true optimal proportions, which could occur anywhere within the limits of our parameter space. By allocating the budget according to the minimum optimal proportion for each parameter, we can find out information regarding the true parameter values without ruling out any proportions which lie within the approximate intervals. For the example under discussion, this would mean allocating 0.3 of the budget to estimating θ_1 and 0.5 of the budget to estimating θ_2, utilising 0.8 of the budget. Collecting data according to this allocation would give us information about the parameters, whilst ensuring that any proportions within the intervals can still be achieved by allocating the remainder of the budget. Using the data collected in stage one, we can calculate the MLEs and the Fisher information matrix, as well as estimate the gradient vector. We can then solve (3.4.4) to find the optimal proportions according to the parameter estimates gained from the first stage data collection, using the existing data to set the lower bounds. The remaining budget can then be allocated in order to achieve these proportions and guide the second stage data collection. Putting all these steps together gives us the following algorithm.

Algorithm 1: Two Stage Algorithm for Data Collection.

Result: First stage data allocation $m_{\boldsymbol{\theta}_1,1}, m_{\boldsymbol{\theta}_2,1}, \ldots, m_{\boldsymbol{\theta}_L,1}$;

Second stage data allocation $m_{\boldsymbol{\theta}_1,2}, m_{\boldsymbol{\theta}_2,2}, \ldots, m_{\boldsymbol{\theta}_L,2}$;

Initialise two factorial design;

for *each design point* i **do**

Compute $I_0(\boldsymbol{\theta}^i)$ and $\hat{\boldsymbol{\delta}}(\boldsymbol{\theta}^i)$;

Find $r_1^i, r_2^i, \ldots, r_L^i$ by solving (3.4.2);

end

for *each input model* l **do**

$r_{l,\min} = \min_i r_l^i$;

$m_{\boldsymbol{\theta}_l,1} = C \times r_{l,\min}$;

end

Collect data according to first stage allocation;

Compute $\hat{\boldsymbol{\theta}}$, $I_0(\hat{\boldsymbol{\theta}})$, and $\hat{\boldsymbol{\delta}}(\hat{\boldsymbol{\theta}})$;

Find $r_1, r_2, \ldots, r_L$ by solving (3.4.4) using lower bounds $r_{1,\min}, r_{2,\min}, \ldots, r_{L,\min}$;

for *each input model* l **do**

$m_{\boldsymbol{\theta}_l,2} = C \times r_l$;

end

Collect remaining data to achieve second stage allocation;

3.6 Experiments

In this section, we illustrate the algorithm on two examples. We first use an $M/M/1$ queueing model to compare the final allocation of data from the two stage algorithm with the true optimal allocation. Secondly, using a more realistic simulation model, we compare input uncertainty estimates given by the two stage algorithm against two commonly used approaches for data collection.

3.6.1 $M/M/1$ Queueing Model

To experiment with the two stage algorithm we use first use an $M/M/1$ queueing model since closed-form expressions can be found for many performance measures. We measure the mean queueing time and since we are able to derive the gradient measures analytically, we can calculate the true optimal proportions in which to collect data such that input uncertainty is minimised. To evaluate the performance of the two stage algorithm, we can compare the final proportions in which the data is allocated to the true optimal proportions which minimise input uncertainty.

To implement the two stage algorithm, let us assume that the input parameters are known to fall within the following intervals: $\lambda^0 \in [3, 6]$ and $\mu^0 \in [9, 12]$. Note these parameter choices are arbitrary, but ensure the traffic intensity is always below 1. We run the simulation for 1000 time periods, using $n = 1000$ replications to estimate the gradients, and the budget for data collection is set to $C = 1000$ observations. Within this controlled experiment we can set true parameters and use these to generate synthetic data, here we use a Latin hypercube sample with 10 intervals to generate 10 sets of true parameters within the parameter space. For each set of parameters,

Parameter Set	λ^0	μ^0	r_1	r_2
1	5.980	9.364	0.423	0.577
2	4.396	9.038	0.398	0.602
3	3.080	11.516	0.366	0.634
4	3.304	10.015	0.375	0.625
5	3.861	10.602	0.379	0.621
6	5.545	9.780	0.411	0.589
7	5.248	10.856	0.397	0.603
8	4.748	10.401	0.393	0.607
9	4.138	11.732	0.378	0.622
10	5.017	11.223	0.392	0.608

Table 3.6.1: Parameter values and analytically derived true optimal proportions for the $M/M/1$ queueing model experiment.

we can derive the true optimal proportions using the analytical gradient measures (see Appendix A.3 for more detail). Table 3.6.1 shows the values of λ and μ for each

parameter set, along with the analytically derived true optimal proportions.

At each of set of parameters, we run the two stage algorithm 100 times, recording the final recommended proportions in which the data is allocated in each of the experiments. Figure 3.6.1 shows box plots of the final proportion of data allocated to λ for each set of input parameters. The red dots indicate the true optimal proportion for each set of parameters calculated using the analytical gradient measures.

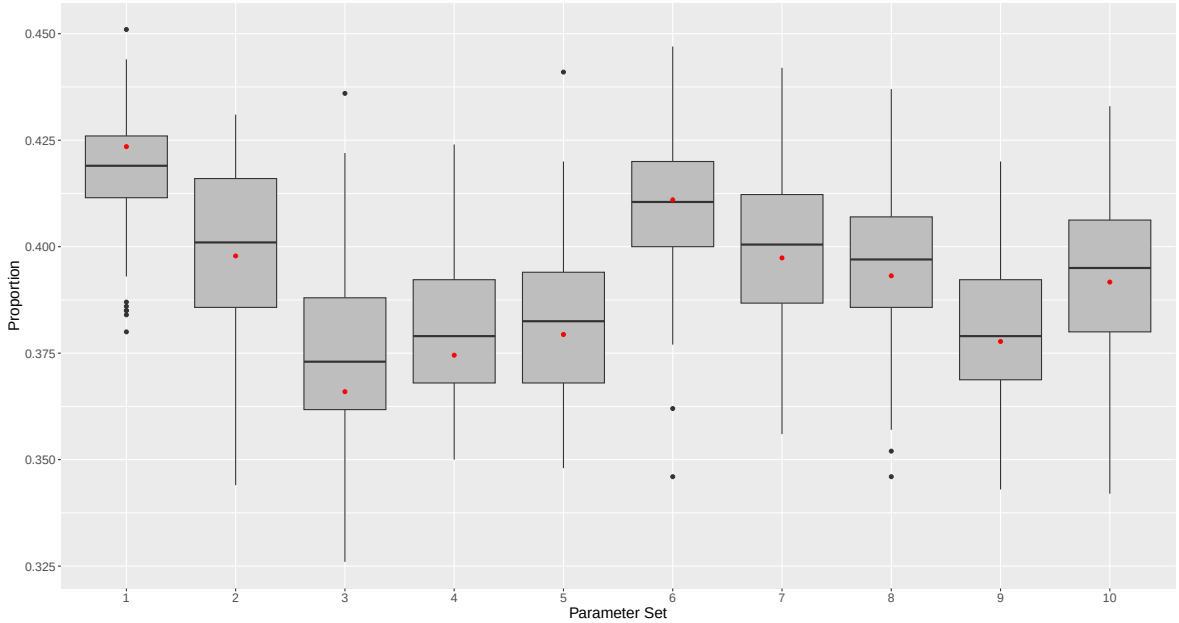

Figure 3.6.1: Box plots showing 100 final proportions for λ from the two stage algorithm, compared to the true optimal proportion, shown in red, at 10 different sets of parameters for the $M/M/1$ queueing model.

For each set of parameters, the box plot of proportions from the two stage algorithm is concentrated on the true optimal proportion, showing that the two stage algorithm is able to hone in towards an optimal collection of data. We expect to see some variation around the true optimal proportions for two reasons. Firstly, the final proportion of data allocated to λ is based upon the parameter estimates from the first stage data collection and these will not to be equivalent to the true parameters since we only have a finite budget. Secondly, the gradient terms are estimated and hence will differ from the true gradient measures. Although variability is evident, these results are promising.

3.6.2 Network Queueing Model

We now consider a more realistic simulation model, a network queueing model consisting of three consecutive multi-server queues. Entities arriving at the system join the queue at node 1. After receiving service at node 1 an entity may leave the system or join the queue at node 2, and similarly after receiving service at node 2 an entity may leave the system or join the queue at node 3. After service at node 3 an entity departs the system. This model could represent a medical centre for example and may be used to solve problems relating to capacity planning and resource allocation. Systems such as this are referred to as operational models of healthcare units in Brailsford (2007).

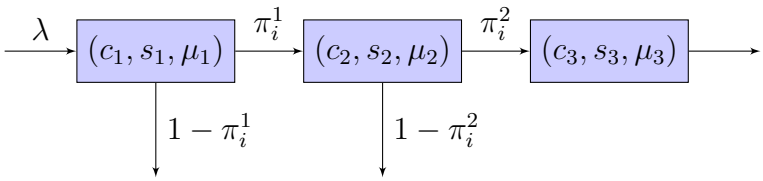

Figure 3.6.2: A graphical representation of the network queueing model.

The network queueing model is set up as follows. At nodes 1, 2 and 3 there are c_1, c_2, and c_3 servers respectively, each of which have a shifted exponential service distribution with parameters (s_1, μ_1), (s_2, μ_2) and (s_3, μ_3). Arrivals to the system follow a stationary Poisson process with rate λ. To represent different demographics of the population the arrivals are split into three different types; 50% of arrivals are of type A, 30% are type B, and 20% are type C. Each type refers to how likely an entity is to travel through the system and so each is defined by a set of probabilities (π_i^1, π_i^2) representing the probability of continuing from node 1 and node 2 respectively. Finally, let us suppose the performance measure of interest is the average queueing time weighted by type. For simplicity, we shall assume the shift parameter of each service distribution and the routing probabilities for each type are known. The unknown input model parameters that require estimation are therefore the arrival rate λ, and the three service rates μ_1, μ_2, μ_3. To implement the two stage algorithm, we assume that the input parameters

are known to fall within the following intervals: $\lambda^0 \in [12, 16]$, $\mu_1^0 \in [18, 22]$, $\mu_2^0 \in [8, 12]$, and $\mu_3^0 \in [6, 10]$. The remaining information about the system is known and is as follows: there are two servers at each node $(c_1, c_2, c_3) = (2, 2, 2)$, the shift parameters for the service distributions are $(s_1, s_2, s_3) = (0.05, 0.05, 0.1)$, and the routing probabilities for types A, B, and C are $(\pi_A^1, \pi_A^2) = (0.4, 0.4)$, $(\pi_B^1, \pi_B^2) = (0.7, 0.7)$, and $(\pi_C^1, \pi_C^2) = (0.9, 0.9)$ respectively.

We now evaluate the performance of the two stage algorithm against other data collection approaches by comparing the input uncertainty passed to the simulation response. One alternative we consider is the equal observations approach, where the same amount of data is collected to estimate each input model. This may be the case in a simple service system where arrivals and services are recorded consecutively. To implement this approach, we can simply split the budget equally amongst the input models and generate observations from each true input distribution. The second alternative we consider is the timed observation approach, where data is collected by observing the true system over some set period of time. An example of this can be found in Griffiths et al. (2005) where an intensive care unit model was developed using data taken over the course of a year. By using the true parameters to run our simulation model for some chosen period of time, we can imitate collecting data from a timed observation of the real-world system.

Within our experiment, we wish to compare input uncertainty estimates given by the three approaches when using the same budget. Since the timed observation approach has no fixed number of observations, we run this first for 250 time periods. The total number of observations gathered from this approach is then used as the budget for the two stage algorithm and for the equal observations approach. We generate 5 random sets of true parameters, so we can compare the approaches across the parameter space when the optimal proportions of the true parameters vary. Table 3.6.2 shows the parameter values for each of the 5 parameter sets. For each of set of parameters, we

Parameter Set	λ^0	μ_1^0	μ_2^0	μ_3^0
1	12.746	19.617	8.127	6.005
2	13.296	18.683	10.523	7.403
3	14.021	18.226	10.049	8.357
4	15.206	20.313	11.236	9.808
5	13.754	19.228	10.928	9.467

Table 3.6.2: Parameter values for the network queueing model experiment.

run the three approaches 100 times. Input uncertainty estimates for each approach are estimated using the same amount of simulation effort and are recorded in the box plots in Figure 3.6.3.

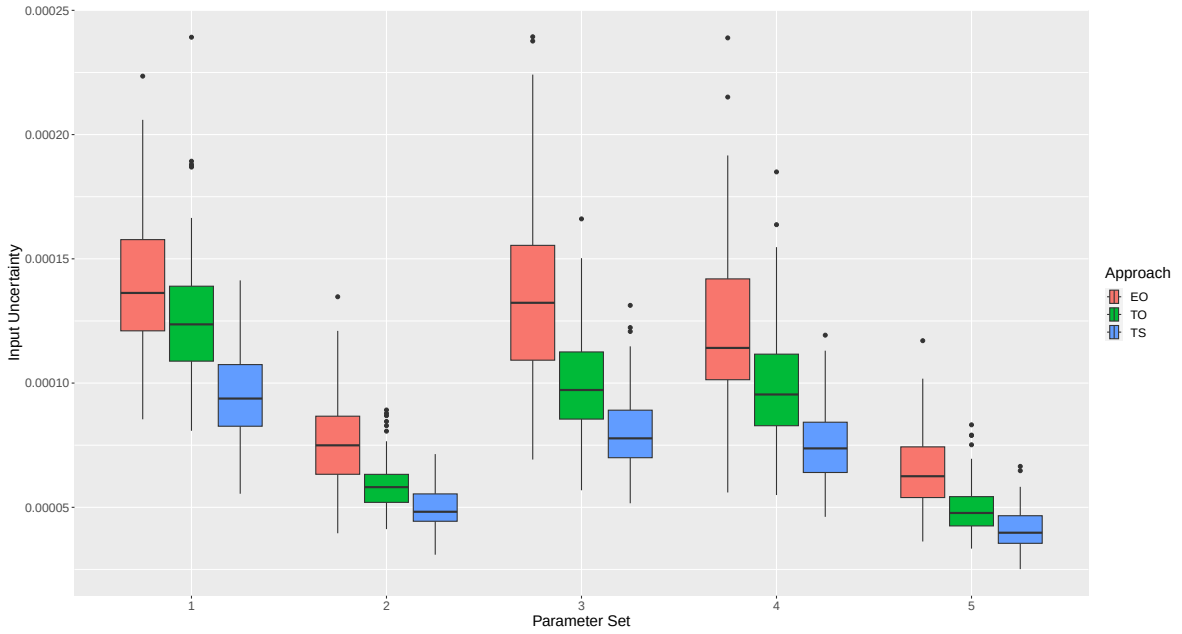

Figure 3.6.3: Box plots showing 100 estimates of input uncertainty from using the two stage algorithm (TS), using an equal observations approach (EO), and using a timed observation approach (TO), at 5 different sets of parameters for the network queueing model.

For each parameter set, the two stage algorithm reduced the mean input uncertainty between 31.44% and 40.81% compared to the equal observation approach, and between 16.05% and 24.23% compared to the timed observation approach. For each set of parameters, the two stage algorithm allocates more observations to the arrival rate compared to the other two approaches, and by doing so reduces input uncertainty

despite using the same overall number of observations. The two stage algorithm shows that by using some knowledge of what values the input parameters might take, we can collect data for our input models in a manner that effectively reduces the input uncertainty of our performance measure. A caveat here is that both the equal observation and timed observation approach require no prior knowledge of input parameter values nor any simulation runs, and can be completed in a single collection.

3.7 Additional Experiments

In this section, we run some additional experiments with the two stage algorithm to help identify areas for future work. We firstly investigate the performance of the algorithm when the true input parameters lie outside the specified intervals, using both the $M/M/1$ queueing model, and the network queueing model. Following this, we illustrate the problem of a small first stage data allocation and a potential pitfall of using the two factorial design, and briefly discuss a potential solution for each of these problems.

3.7.1 External Parameters

The two stage algorithm assumes that each input parameter lies within some known interval, however in reality it is unlikely that such an interval is known with complete confidence. Although experts and practitioners may be able to provide estimates for such intervals, these will only be approximations, and therefore we cannot be certain that the true parameters will lie within the intervals. It is worth noting however that the two stage algorithm works regardless of the values that the true parameters take. The second stage allocation minimises input uncertainty for the MLEs calculated from the first stage collection, regardless of whether the MLE for each parameter lies in its interval or not. The minimisation however is constrained by the first stage data allocation, which is based upon the optimal proportions found from the two factorial

design over the parameter space. The issue with true parameters taking values outside their specified interval is that the optimal proportions for these parameters without any constraints may not be achievable, as the first stage data collection could rule them out. To understand the behaviour and performance of the algorithm in these situations, we run some experiments using both the $M/M/1$ queueing model and the network queueing model.

Firstly, we run some experiments using the $M/M/1$ queueing model. Recall that the initial intervals for the two parameters were given by $\lambda^0 \in [3, 6]$ and $\mu^0 \in [9, 12]$. We wish to test the performance of the algorithm when the true input parameters lie externally to the initially specified parameter space. We experiment with 8 different parameter sets; 4 generated using a two factorial design when the lower and upper limits of each interval are extended by 1, and 4 by repeating the factorial design but extending the limits by 1.25. These parameter values are likely to lead to situations where the true optimal proportions become unobtainable due to the first stage data collection. The parameter values and the analytically derived true optimal proportions are shown in Table 3.7.1.

Parameter Set	λ^0	μ^0	r_1	r_2
1	2.0	8.0	0.364	0.636
2	2.0	13.0	0.351	0.649
3	7.0	8.0	0.471	0.529
4	7.0	13.0	0.406	0.594
5	1.75	7.75	0.360	0.640
6	1.75	13.25	0.349	0.651
7	7.25	7.75	0.484	0.516
8	7.25	13.25	0.408	0.592

Table 3.7.1: Parameter values and analytically derived true optimal proportions for the external parameter experiment using the $M/M/1$ queueing model.

At each set of external parameters, we run the two stage algorithm 100 times, recording the final proportions in which the data is allocated in each of the experiments. Figure 3.7.1 shows box plots of the final proportions of data allocated to λ for each set

of parameters, along with red dots to indicate the true optimal proportions. For the majority of parameter sets, the red dot does not lie close to the median of the box plot.

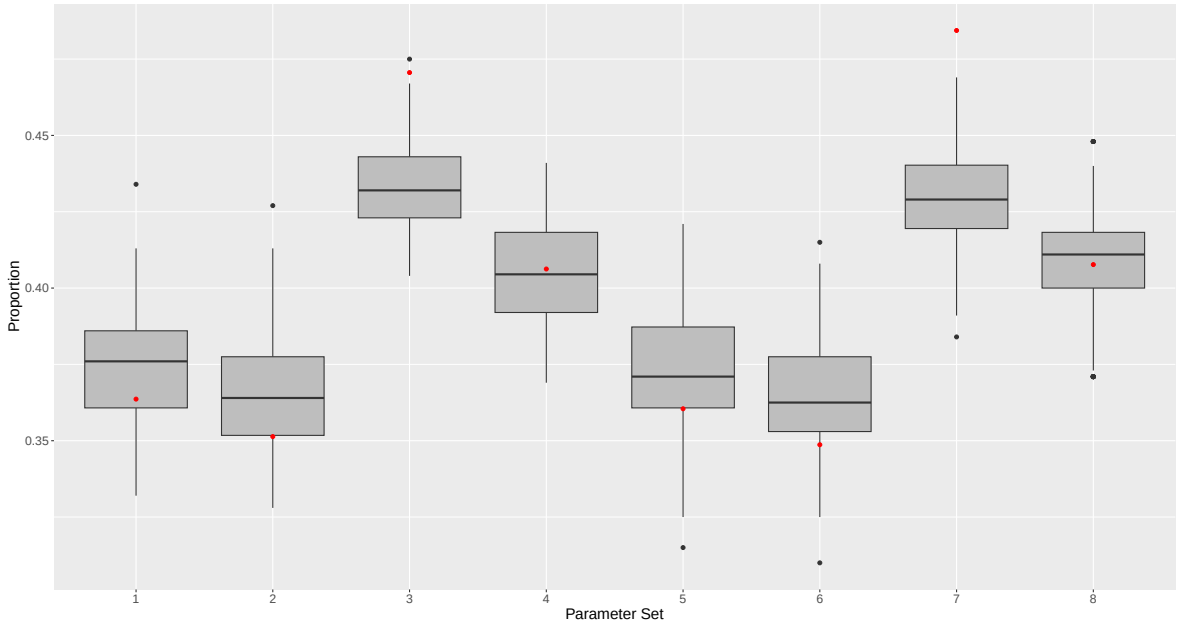

Figure 3.7.1: Box plots showing 100 final proportions for λ from the two stage algorithm, compared to the true optimal proportion, shown in red, at 8 different sets of external parameters for the $M/M/1$ queueing model.

The factorial design used to estimate the proportion intervals in the two stage algorithm uses parameter values from the end points of the specified intervals. In this case, subject to some gradient approximation error, the suggested proportion intervals for λ and μ will approximately be $[0.364, 0.429]$ and $[0.571, 0.636]$ respectively. The first stage data collection is guided based on the premise that the true optimal proportion for each parameter will lie in these proportion intervals. In the cases where the external parameter sets have true optimal proportions that fall in these intervals (parameter sets 4 and 8), the proportions are achievable, demonstrated by the box plots being centred around the true optimal proportions. In the other cases where the external parameter sets have true optimal proportions that lie outside, or at the edge of these intervals, the proportions often become unachievable, and this is seen by the true optimal proportions being located away from the centre of the box plots. However, we can see that in these

scenarios, the final proportions from the algorithm typically lie as close as possible to the optimal, given the suggested proportion intervals from the two factorial design, and ultimately this may still produce a reduced level of input uncertainty than if the algorithm had not been used at all.

We continue our experiments with external parameters, this time using the network queueing model. Recall that the initial parameter intervals were given by $\lambda^0 \in [12, 16]$, $\mu_1^0 \in [18, 22]$, $\mu_2^0 \in [8, 12]$, and $\mu_3^0 \in [6, 10]$. To generate some input parameters that lie outside these specified parameter intervals, we use the parameter values from Table 3.6.2 and add some random noise via a normally distributed random variable with mean 0 and variance 4. This returns the sets of parameters in Table 3.7.2. We use bold text to denote when a parameter lies outside its specified interval, so we can see for sets 1, 2, and 5, two parameters lie outside their specified intervals, whilst for sets 3 and 4, only one parameter lies outside its specified interval.

Parameter Set	λ^0	μ_1^0	μ_2^0	μ_3^0
1	14.059	**24.715**	8.057	**4.666**
2	13.280	**22.237**	8.246	**10.139**
3	**16.680**	18.899	10.062	7.446
4	14.473	21.610	**15.377**	9.501
5	**10.972**	**17.781**	11.445	8.833

Table 3.7.2: Parameter values for the external parameter experiment using the network queueing model.

As we did in the previous network queueing model experiment, we compare input uncertainty estimates given by the two stage algorithm against two other data collection approaches, when using the same total budget for observations. The timed observation approach is run first for 250 time periods, and the total number of observations taken from this approach is then used as the budget for the two stage algorithm and for the equal observations approach. For all the external parameter sets shown in Table 3.7.2, we run each of the three data collection approaches 100 times. We then estimate the input uncertainty introduced by each approach, using a consistent amount of simula-

tion effort for each estimate. Figure 3.7.2 presents box plots of the input uncertainty estimates produced by each data collection approach across each of the parameter sets. Due to the variation in input uncertainty values, the box plots for each parameter set are given on separate scales.

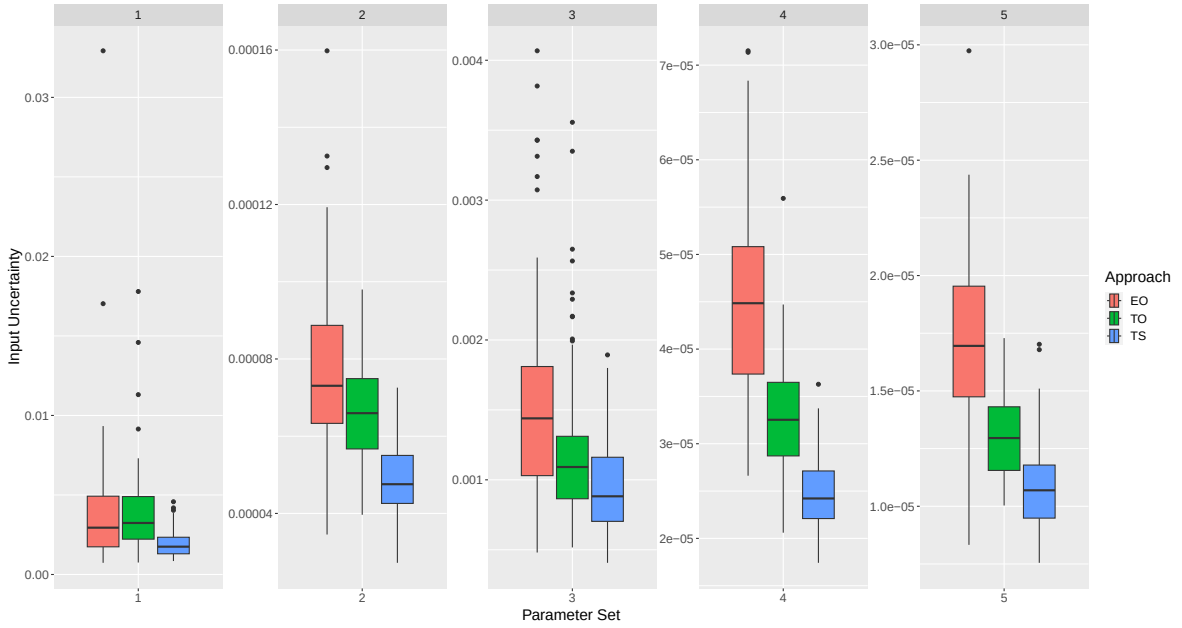

Figure 3.7.2: Box plots showing 100 estimates of input uncertainty from using the two stage algorithm (TS), using an equal observations approach (EO), and using a timed observation approach (TO), at 5 different sets of external parameters for the network queueing model.

Figure 3.7.2 illustrates that in these preliminary experiments where one or two parameters lie outside their specified intervals, the two stage algorithm still gives a reduction in input uncertainty compared to both other data collection approaches. This suggests that having an approximate idea of what value each parameter may take is still effective in guiding data collection to minimise input uncertainty, although this may not always be the case.

Further experimentation could be undertaken here, for example a larger scale study could be conducted on this specific model and setup to see whether the reduction in input uncertainty is still obtained when more than two parameters lie outside their specified intervals. Additionally, similar experiments could also be conducted using different

parameter intervals, as the results are likely to vary depending on the parameter space and the optimal proportions found across the two factorial design. Although empirical results from such experiments may give us confidence that the two stage algorithm can produce a reduced level of input uncertainty when input parameters lie externally to their intervals, such results are likely to be specific to the particular simulation model and parameter intervals.

If some parameters lie far outside their initially specified intervals, then there may be a chance that input uncertainty becomes larger under the two stage algorithm compared to the alternative approaches. However, since the two stage algorithm aims to take into account input uncertainty prior to data collection, we might expect it to return a reduced level of input uncertainty compared to other approaches that do not consider input uncertainty, even when parameters may lie externally.

3.7.2 Small First Stage Allocation

We currently impose no constraint on how large the first stage data allocation needs to be for each parameter. If the minimum optimal proportion for a parameter is small, or the budget multiplied by the minimum optimal proportion is small, then the first stage data allocation will suggest collecting few observations for the parameter, which is likely to lead to an inaccurate parameter estimate. If this is the case then the proportions calculated using the first stage data collection, which are the target proportions, will minimise input uncertainty at a point in the parameter space which may be far away from where the true parameters lie. Consequently, the proportions in which the data is collected may differ greatly from the optimal proportions at the true parameters.

We might expect this to occur when very wide parameter intervals are specified. Although very wide intervals are more likely to contain the true parameter, in queueing style simulation models they may lead to design points in which the system doesn't reach steady state. In general, the design points will represent extreme scenarios in

which certain parameters may hardly impact input uncertainty and hence have very small optimal proportions. Consequently, the minimum optimal proportion for some parameters may be very small, leading to small first stage data allocations and poor parameter estimates. Here, we will demonstrate this problem using a specific example with the network queueing model.

We consider the exact same experimental setup for the network queueing model as previously, where the input parameters were assumed to fall within the following intervals: $\lambda^0 \in [12, 16]$, $\mu_1^0 \in [18, 22]$, $\mu_2^0 \in [8, 12]$, and $\mu_3^0 \in [6, 10]$. Suppose that we run the two stage algorithm, with a data collection budget of $C = 1000$ observations.

First, we use the two factorial design to generate the set of design points. For each design point, we estimate the Fisher information matrix and run a set of replications with the design point parameters to estimate the gradient vector. This gives us the a_l terms in (3.4.2) required to calculate the optimal proportions, at each design point. Table 3.7.3 contains the design point parameter values and their corresponding optimal proportions, shown to four decimal places.

Design Point i	λ^i	μ_1^i	μ_2^i	μ_3^i	r_1^i	r_2^i	r_3^i	r_4^i
1	12	18	8	6	**0.5149**	0.1218	0.1986	0.1647
2	16	18	8	6	0.5576	0.1306	0.1710	0.1408
3	12	22	8	6	0.5226	0.0655	0.2326	0.1793
4	16	22	8	6	0.5631	**0.0606**	0.2039	0.1725
5	12	18	12	6	0.5401	0.1854	0.0639	0.2106
6	16	18	12	6	0.5896	0.1927	0.0249	0.1927
7	12	22	12	6	0.5490	0.0887	0.0716	0.2908
8	16	22	12	6	0.5731	0.1069	**0.0232**	0.2969
9	12	18	8	10	0.5239	0.1795	0.2500	0.0465
10	16	18	8	10	0.5733	0.1814	0.2365	**0.0087**
11	12	22	8	10	0.5201	0.1067	0.3145	0.0586
12	16	22	8	10	0.5593	0.0799	0.3493	0.0115
13	12	18	12	10	0.5697	0.2870	0.0929	0.0503
14	16	18	12	10	0.5789	0.3432	0.0407	0.0373
15	12	22	12	10	0.5673	0.1991	0.1358	0.0978
16	16	22	12	10	0.5821	0.2393	0.1086	0.0700

Table 3.7.3: Example optimal proportions for the 2^4 factorial design in the small first stage allocation experiment.

For each input model parameter, we need to find the minimum optimal proportion across the design points. The bold numbers in each column represent the minimum optimal proportions, and these are used to allocate data in the first stage collection. Noticeably, the minimum optimal proportions vary in magnitude across the four input parameters. For individual parameters, the optimal proportions exhibit different levels of variability across design points. For example, the optimal proportions for λ (r_1^i) appear quite consistent across design points, however the optimal proportions for each of the service rate parameters (r_2^i, r_3^i, r_4^i) are much more variable. Multiplying the minimum optimal proportion for each parameter by the data collection budget gives us the following first stage data allocation

$$m_{\lambda,1} = 515, \quad m_{\mu_1,1} = 61, \quad m_{\mu_2,1} = 23, \quad m_{\mu_3,1} = 9,$$

utilising a total of 608 observations. The small number of observations allocated to estimating μ_2 and μ_3 are concerning, as they are unlikely to give rise to accurate parameter estimates.

We are interested in what happens here when we run the remaining steps of the two stage algorithm, with particular interest in μ_3. Suppose that the true input parameters are given by the midpoints of each interval, $\boldsymbol{\theta}^0 = (14, 20, 10, 8)$. We collect observations according to the first stage allocation and calculate the MLEs, Fisher information matrix, and then estimate the gradient vector. Using this information, we find the final targeted optimal proportions based on the first stage data collection. In two separate cases of the two stage algorithm, we found the following first stage parameter estimate and subsequent optimal proportion for parameter μ_3, based on the 9 observations

$$\hat{\mu}_3 = 15.14 \quad \rightarrow \quad r_4 = 0.013,$$
$$\hat{\mu}_3 = 6.17 \quad \rightarrow \quad r_4 = 0.280.$$

The two parameter estimates are significantly different, and hence so are the optimal proportions. In reality, the true parameter is $\mu_3^0 = 8$, which has an optimal proportion of $r_4 = 0.110$. In the first case, the larger parameter estimate corresponds to a quicker service time at node 3. Since the performance measure relates to queueing time, the gradient of this parameter becomes smaller than at the true parameter value, and hence the algorithm suggests allocating a smaller amount of data to estimate this parameter. Conversely, the smaller parameter estimate corresponds to a slower service time at node 3. This leads to a situation where the simulation response appears more sensitive to the parameter, and hence the gradient of the parameter is greater than at the true parameter. The algorithm therefore suggests allocating a larger amount of data to estimate this parameter. Either way, the final targeted proportion differs drastically from the optimal proportion of the true parameter.

A possible solution to this problem of a small first stage allocation is to introduce a minimum allocation level. This would involve allocating at least a minimum number of observations for each parameter in the first stage data allocation, to ensure that sufficiently accurate parameter estimates are obtained. How to pick this minimum allocation level is not instantly straightforward. We might like to consider whether the minimum allocation level should be fixed across different scenarios, or whether it should somehow be adapted to the specific problem and parameters at hand. For the latter, we might want to investigate whether the minimum allocation level should in some way depend upon the number of input parameters and the data allocation budget, as well as the minimum optimal proportion for each parameter. It is also worth noting that considering the minimum optimal proportion for a parameter ignores the variability of the optimal proportion over the design points. It is possible that all other optimal proportions for the parameter may be much greater than the minimum, or perhaps only slightly greater, and this information would likely affect the choice of the minimum allocation level.

There is a trade-off in introducing a minimum allocation level. For those parameters whose number of observations increases due to the minimum allocation level, although we are more likely to gain an improved parameter estimate we rule out proportions that may give a reduced level of input uncertainty had we collected fewer observations. This might be the case if the simulation response is extremely insensitive to a certain input parameter and therefore an accurate parameter estimate is not necessary to ensure a small value of input uncertainty.

3.7.3 Internal Minimum Optimal Proportion

In the two stage algorithm, we use a two factorial design to explore how the optimal proportion for each parameter varies across the parameter space. We set the lower and upper factor levels for each parameter using the lower and upper bounds of the specified intervals. This means we only study parameter values at the bounds of each interval, and fail to examine any values that fall directly within each interval. This could prove problematic, as we demonstrate briefly in the following experiment.

Consider a simulation model with two parameters $\boldsymbol{\theta} = (\theta_1, \theta_2)$, and suppose that the intervals for each parameter are given by $\theta_1^0 \in [2, 4]$ and $\theta_2^0 \in [2, 4]$. Suppose we initialise the two factorial design and find that the optimal proportions at each of the four design points are given by the results in Table 3.7.4.

Design Point i	θ_1^i	θ_2^i	r_1^i	r_2^i
1	2	2	0.45	0.55
2	4	2	0.45	0.55
3	2	4	0.45	0.55
4	4	4	0.45	0.55

Table 3.7.4: Optimal proportions for the 2^2 factorial design from the internal minimum optimal proportion experiment.

In this case, the optimal proportions are the same across all the design points, and hence the minimum optimal proportion for each parameter occurs at every design

point. Based on these results, the two stage algorithm would allocate 0.45 of the data allocation budget to estimating θ_1, and the remaining 0.55 to estimating θ_2. In this unique case, the algorithm would be completed in a single stage of data collection. However, suppose that the optimal proportion for θ_1 across the parameter space is represented by the heat map shown in Figure 3.7.3.

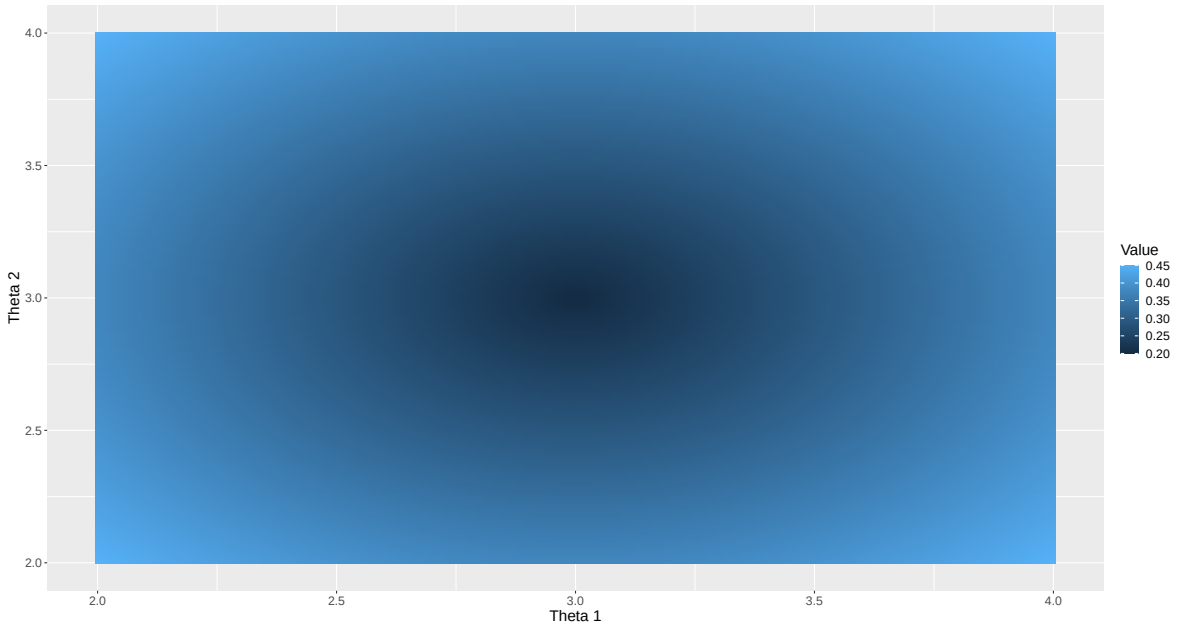

Figure 3.7.3: Heat map showing r_1 (the optimal proportion for θ_1), as both θ_1 and θ_2 vary over their specified parameter intervals.

Across the parameter space, the optimal proportion for θ_1 falls as low as 0.2, and thus the optimal proportion for θ_2 must reach as high as 0.8. These specific values occur when each parameter lies at the centre of their respective interval. They are not picked up by the two factorial design, since this only considers combinations of the interval end points, which are represented by the corners of the heat map. In this example, should the true parameters lie anywhere towards the centre of the parameter space, the two stage algorithm would be unable to get close to the optimal allocation of observations, due to the nature of the results from the two factorial design.

Although this experiment is contrived to illustrate this point, it highlights a potential issue with the two stage algorithm that could occur in practice. There are a multitude of

different ways that the parameter space may be explored in order to understand how the optimal proportions for each parameter vary, rather than via a two factorial design. For example, we could consider using a central composite design, which would include centre points and axial points (Montgomery, 2017), or we could use Latin hypercube sampling (McKay et al., 2000) to generate sets of input parameters. Alternative approaches are likely to perform better or worse in different circumstances, depending on the number of input parameters and how the true optimal proportions vary over the specified intervals. We therefore might like to investigate finding an approach that is robust across different scenarios. Approaches that utilise more design points are more likely to give a better understanding of how the optimal proportion for each parameter changes, and hence more likely to hone in towards an optimal collection of data. However, more design points equates to a larger initial simulation effort since more gradient vectors require estimation, so there is a trade-off to be had here.

3.8 Conclusion

In this chapter, we introduced the novel idea of allocating an initial budget for data collection in a manner that minimises input uncertainty. In particular, we have developed an algorithm that by collecting data in two different stages aims to hone in on an optimal allocation of data across the input models. We use a two stage approach so that we can approximate the parameter values first, and then attempt to collect data optimally. Using an $M/M/1$ queueing model, we have demonstrated that the algorithm achieves an allocation of data that is close to the true optimal allocation. On a more realistic simulation model, we have shown that the two stage algorithm results in a reduced level of input uncertainty compared to two other viable approaches for data collection.

We have started to investigate how the algorithm behaves and performs when the

assumption of parameters lying in the specified intervals is broken, and the preliminary results indicate a reduced level of input uncertainty can still be achieved compared to alternative data collection approaches. We have also illustrated two potential problems with the algorithm. The first is the issue of a small first stage data allocation, which can lead to inaccurate parameter estimates and hence suboptimal proportions. The second is the issue of the two factorial design only exploring the corner points of the parameter space, which could be problematic if the minimum optimal proportion for a parameter lies internal to its specified interval. We have suggested some potential solutions to these problems, introducing a minimum allocation level and utilising different experimental designs to explore the parameter space, however these require further investigation and experimentation. We could also investigate using additional stages in the algorithm, though there is likely to be a trade-off here in terms of the reduction in input uncertainty and the effort required for additional data collection steps.

Chapter 4

Comparing Data Collection Strategies when Simulating Viral Load Profiles

Temporal profiles of viral load have individual variability and are used to determine whether individuals are infected based on some limit of detection. Modelling and simulating viral load profiles allows for the performance of testing policies to be estimated, however viral load behaviour can be very uncertain. We describe an approach for studying the input uncertainty passed to simulated policy performance when viral load profiles are estimated from different data collection strategies. The content of this chapter has been published as Parmar et al. (2021a), with some minor textual changes.

4.1 Introduction

Individuals infected by a virus have a temporal profile of viral load starting from the moment they are infected. If and when they are tested their test sensitivity, the probability that an infected individual will correctly receive a positive test result, is a function of their viral load. There is individual variability in profiles of viral load, and therefore

in test sensitivity too. Viral load can be a chief predictor for the risk of virus transmission (Quinn et al., 2000) and can also be strongly linked to mortality (Pujadas et al., 2020), therefore monitoring and understanding the behaviour of viral load is extremely important. We are interested in modelling the individual variability in temporal profiles of viral load and test sensitivity, and using this model to simulate and compare the performance of different testing policies, where a testing policy is a choice of how many times to test an individual and when. This might assist in suppressing the spread of a virus by for example indicating which testing policies are most effective in terms of successfully identifying the virus. Estimates of the shape of viral load profiles and the individual variability can be very uncertain, since it requires observations of viral load across a group of individuals at different time points. In collecting such data there are choices to be made regarding the number of individuals to observe, the number of times to observe each individual and the time points at which the observations are taken. We shall refer to a particular choice of these as a data collection strategy. Modelling viral load profiles directly from data will introduce a source of uncertainty into the simulation that will propagate through to the outputs. We wish to study how data collection strategies differ in terms of the amount of uncertainty they pass through the simulation to the estimated performance of a testing policy, as if a particular strategy offers a reduced level of uncertainty then this can be used for future data collection.

Here, we outline an approach to compare the uncertainty passed to simulation outputs due to different data collection strategies. We use a nonlinear mixed effects model to allow for individual variability in viral load profiles and test sensitivity, and using this as our input model we are able to simulate the performance of different testing policies. We focus on how different data collection strategies used to estimate the input model parameters compare in terms of the uncertainty that propagates through the simulation model to the output, known as input uncertainty. This comparison allows us to consider whether a particular strategy should be adopted in order to obtain a

reduced level of input uncertainty and thus gain greater insight from the simulated performance of testing policies.

The rest of the chapter is organised as follows. We discuss background literature in Section 4.2 and outline the steps for our general approach in Section 4.3. An experiment using this approach is illustrated in Section 4.4. We discuss the results of our experiment and areas for future work in Section 4.5 before concluding in Section 4.6.

4.2 Background

Lindstrom and Bates (1990) proposed a general nonlinear mixed effects model for data observed from a number of individuals repeatedly under different conditions, otherwise known as repeated measures data. They estimate the model parameters via a two-step iterative procedure that draws on methods developed for nonlinear fixed effects models and linear mixed effects models.

Chen et al. (2021) model respiratory viral load for patients infected with SARS-CoV-2 using a mechanistic model for respiratory virus kinetics based on a system of differential equations. Heterogeneity of viral load is evaluated by fitting Weibull distributions at each time point in which sample data is available. Kucirka et al. (2020) model the time profile of test sensitivity using Bayesian hierarchical logistic regression. By using a nonlinear mixed effects model, we are able to incorporate individual variability in viral load and test sensitivity under a single model.

Quilty et al. (2021) consider how quarantine and testing policies impact transmission of SARS-CoV-2 using an agent-based model to simulate the dynamics of viral load for exposed individuals. Larremore et al. (2021) use temporal profiles of viral load to simulate the effectiveness of certain testing policies and tests with different properties in controlling an epidemic. Rather than aiming to compare different testing policies directly, we are interested in how using different data collection strategies to model

viral load profiles can result in varying levels of uncertainty in simulated responses.

Barton (2012) and Song et al. (2014) give an introduction to input uncertainty, a term which refers to the variance passed to a simulation output due to having estimated the input models via real-world data, and describe techniques to quantify it. For recent advancements, see Lam and Qian (2017, 2018a). Efforts by Nelson et al. (2021) have also been made to reduce input uncertainty using frequentist modelling averaging. Morgan et al. (2019) recently considered detection and quantification of bias caused by input modelling.

4.3 Approach

In this section, we outline the steps in our approach. We introduce our model for viral load and discuss how we simulate the performance of testing policies. We define input uncertainty, a method for quantifying it, and describe how we can compare the impact different data collection strategies have on it.

4.3.1 Modelling Viral Load

To represent the individual variability in temporal profiles of viral load we use a non-linear mixed effects model, as it seems appropriate that all profiles follow the same functional form but with individual variability in some parameters. Let us consider a group of infected individuals who have unique temporal profiles of viral load and suppose we are able to observe, with noise, the viral loads of these individuals at different time points. An example of this can be found in Wölfel et al. (2020), where daily measurements of viral load are taken from different patients using RT-PCR. Let y_{ij} denote the viral load observed in individual i at time t_{ij}, where $i = 1, \ldots, N$, and $j = 1, \ldots, n_i$.

Our nonlinear mixed effects model is as follows

$$
\begin{aligned}
y_{ij} &= v(t_{ij}, \boldsymbol{\theta}_i) + g(t_{ij}, \boldsymbol{\theta}_i, \boldsymbol{\phi})\epsilon_{ij}, \\
\boldsymbol{\theta}_i &\sim \mathcal{N}(\boldsymbol{\theta}_{\text{pop}}, \boldsymbol{\Omega}), \\
\epsilon_{ij} &\sim \mathcal{N}(0, 1),
\end{aligned}
$$

where v is a nonlinear function for the temporal profile of viral load, $\boldsymbol{\theta_i}$ is a parameter vector of length m for the ith individual, g is a function for the residual error, $\boldsymbol{\phi}$ is a parameter vector of length q for the residual error function and ϵ_{ij} is an error term which follows a standard normal distribution. Individual parameter vectors are assumed to be normally distributed with mean $\boldsymbol{\theta}_{\text{pop}}$, a vector of length m representing the fixed population parameters and variance $\boldsymbol{\Omega}$, an $m \times m$ covariance matrix representing the random effects. The residual errors are assumed to be independent for different individuals and independent of each other for the same individual.

The parameters of the model $(\boldsymbol{\theta}_{\text{pop}}, \boldsymbol{\Omega}, \boldsymbol{\phi})$ can be estimated by maximising the log-likelihood function or the restricted log-likelihood function of the observed data. These expressions cannot be evaluated analytically as they have no closed form, however computational methods can be used to obtain parameter estimates $(\hat{\boldsymbol{\theta}}_{\text{pop}}, \hat{\boldsymbol{\Omega}}, \hat{\boldsymbol{\phi}})$.

4.3.2 Simulation

We can use the estimated fixed and random effects to simulate a large number of parameter vectors, where each parameter vector defines the temporal profile of viral load for an individual. This tells us about the behaviour of viral load profiles amongst a population. Moreover, the residual error function and its estimated parameters tell us about the distribution of error that would occur in observing the viral load of an individual at any time point. A test result for an individual at a specific time is based on whether their viral load measurement lies above or below some limit of detection. For

each simulated individual, we can calculate the probability of their observed viral load lying above or below the limit of detection at some given testing time by considering the distribution of error around the true viral load.

Suppose we use the estimated fixed and random effects to simulate K individuals. Each individual k is defined by a parameter vector $\tilde{\boldsymbol{\theta}}_k$, where

$$\tilde{\boldsymbol{\theta}}_k \sim \mathcal{N}(\hat{\boldsymbol{\theta}}_{\text{pop}}, \hat{\boldsymbol{\Omega}}).$$

The true viral load for individual k at some time point t is given by $v(t, \tilde{\boldsymbol{\theta}}_k)$, however we observe the viral load with noise. This noise is distributed according to the residual error function multiplied by a standard normal random variable. The residual error function can be estimated by $g(t, \tilde{\boldsymbol{\theta}}_k, \hat{\boldsymbol{\phi}})$, therefore it follows that the distribution of the observed viral load for individual k at time point t is

$$\mathcal{N}(v(t, \tilde{\boldsymbol{\theta}}_k), g(t, \tilde{\boldsymbol{\theta}}_k, \hat{\boldsymbol{\phi}})^2).$$

For some limit of detection γ, we can use this distribution to measure the probability of an individual correctly or incorrectly returning either a positive or negative test result at any time point. We can use the simulated individuals to study the performance of different testing policies amongst a large population. The performance of a policy, which is a choice of how many times to test each individual and when, could be measured in different ways, for example using the expected number of true positive tests or the average probability of a false negative test.

Let $\hat{\boldsymbol{\Theta}} = \{\hat{\boldsymbol{\theta}}_{\text{pop}}, \hat{\boldsymbol{\Omega}}, \hat{\boldsymbol{\phi}}\}$ denote the estimated parameters of the nonlinear mixed effects model that will drive the simulation. For a testing policy p we denote the simulation output of replication j by

$$Y_j(\hat{\boldsymbol{\Theta}}, p) = \eta(\hat{\boldsymbol{\Theta}}, p) + e_j(\hat{\boldsymbol{\Theta}}, p),$$

where $\eta(\hat{\boldsymbol{\Theta}}, p)$ is the expected value of the simulation output for policy p given the parameters $\hat{\boldsymbol{\Theta}}$, and e is a random variable with mean 0 representing stochastic noise. Note that within a replication, a choice for the number of individuals to simulate must be made. We consider this to be analogous to choosing the run length of replications, and so it is not included in the notation.

We assume that there are unknown but true parameters $\boldsymbol{\Theta}^0 = \{\boldsymbol{\theta}^0_{\text{pop}}, \boldsymbol{\Omega}^0, \boldsymbol{\phi}^0\}$ from which the real-world data is observed and that the goal of the simulation experiment is to estimate $\eta(\boldsymbol{\Theta}^0, p)$, the expected value of the simulation output for policy p given the true parameters $\boldsymbol{\Theta}^0$. We estimate this value by taking the sample mean across n simulation replications using the fitted parameters, that is

$$\begin{aligned}\bar{Y}(\hat{\boldsymbol{\Theta}}, p) &= \frac{1}{n}\sum_{j=1}^{n} Y_j(\hat{\boldsymbol{\Theta}}, p), \\ &= \eta(\hat{\boldsymbol{\Theta}}, p) + \frac{1}{n}\sum_{j=1}^{n} e_j(\hat{\boldsymbol{\Theta}}, p).\end{aligned}$$

The variance of this estimator breaks down into two distinct terms, stochastic estimation error and input uncertainty. Stochastic estimation error arises from the random variates generated in each replication and can be easily estimated via the sample variance.

4.3.3 Input Uncertainty

Input uncertainty refers to the variance in the expected simulation output that arises due to having estimated the input models. In our case input uncertainty simply reduces to parameter uncertainty and is given by

$$\sigma_I^2 = \text{Var}[\eta(\hat{\boldsymbol{\Theta}}, p)].$$

Input uncertainty is not straightforward to estimate since it requires knowledge of the sampling distribution of $\hat{\boldsymbol{\Theta}}$ and the functional form of $\eta(\cdot)$ both of which are usually unknown (Song et al., 2014).

A simple way to estimate input uncertainty is using bootstrapping, which involves sampling with replacement from the input model data, fitting input models according to this bootstrapped data and running replications using the bootstrap fitted input models. Repeating this bootstrapping procedure multiple times and treating the results as a random-effects model allows us to estimate input uncertainty (Nelson, 2013). In the case of repeated measures data bootstrapping can be done by resampling with replacement from entire individuals, this is known as a case or paired bootstrap.

4.3.4 Comparing Data Collection Strategies

By choosing the functions v and g, and selecting some true parameters $(\boldsymbol{\theta}^0_{\text{pop}}, \boldsymbol{\Omega}^0, \boldsymbol{\phi}^0)$ we can simulate viral load observations from the nonlinear mixed effects model. This allows us to generate data sets of observations where we choose the number of individuals observed, as well as the number of observations per individual and the times at which they are observed. Using these data sets we can estimate input parameters, simulate the performance of different testing policies and quantify input uncertainty. This will allow us to investigate how different data collection strategies perform in terms of the uncertainty they propagate to the simulation outputs. If a particular approach clearly returns a reduced level of input uncertainty compared to any alternatives, then using this data collection strategy will provide more insightful simulation results and allow for better comparisons to be made between different testing policies.

4.4 Experiment

In this section, we describe an experiment using the approach outlined in Section 4.3. We select functions and parameters for the nonlinear mixed effects model and compare three different data collection strategies across two sets of testing policies.

4.4.1 Input Model

To provide an example of functions and parameters we use the recent literature on SARS-CoV-2. Note, we are not aiming to accurately model viral load profiles that occurred during the COVID-19 pandemic, rather just choose a nonlinear mixed effects model that is somewhat representative of a real-world problem. We select our model and parameters such that the viral load profiles have properties that match the behaviour found in various sources of literature.

To model the temporal profile of viral load we choose a shifted scaled lognormal function as this has a single peak and an unbounded right tail, with a shape similar to the right-skewed plots of viral load in Wölfel et al. (2020). Individual i is represented by parameter vector $\boldsymbol{\theta}_i = (\mu_i, \sigma_i, \alpha_i)$, where μ_i and σ_i are parameters to the lognormal distribution and α_i is the scale parameter. The viral load of individual i at time t is given by

$$v(t, \boldsymbol{\theta}_i) = \begin{cases} \frac{\alpha_i f(t-s, \mu_i, \sigma_i)}{f(\exp(\mu_i - \sigma_i^2), \mu_i, \sigma_i)}, & t \geq s \\ 0, & t < s \end{cases}$$

where $f(t, \mu_i, \sigma_i)$ is the probability density function of a lognormal distribution with parameters μ_i and σ_i, $s > 0$ represents the shift, and t is interpreted as days since infection. The mode of $f(t, \mu_i, \sigma_i)$ is given by $t = \exp(\mu_i - \sigma_i^2)$, therefore α_i represents the peak of the viral load profile for individual i. Note that v measures viral load in units of $\log_{10}$.

We find fixed and random effects, and a shift value such that the viral load profiles

generated by the nonlinear mixed effects model match properties found in the literature, such as the peak viral load behaviour in Chen et al. (2021) and the time between infection and the peak viral load in Backer et al. (2020). The fixed and random effects are given (to three decimal places) by the following

$$\boldsymbol{\theta}^0_{\text{pop}} = (3.453, 1.415, 7.023), \qquad \boldsymbol{\Omega}^0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.030 & 0.046 \\ 0 & 0.046 & 2.639 \end{bmatrix},$$

with shift $s = 1.5$.

We consider a multiplicative residual error function with parameter vector $\boldsymbol{\phi} = (\beta, \rho)$, where the residual error function for individual i at time t is given by

$$g(t, \boldsymbol{\theta_i}, \boldsymbol{\phi}) = \beta v(t, \boldsymbol{\theta}_i)^\rho.$$

We set the true residual error parameters to be $\boldsymbol{\phi}^0 = (0.25, 1)$, as this means the test sensitivity over time exhibits similar behaviour to that found in Kucirka et al. (2020).

4.4.2 Simulation

In our experiment, we measure policy performance using the average probability of an individual returning at least one positive test. An individual will return a negative test result if their observed viral load is below some limit of detection γ. The probability of a simulated individual k with parameters $\tilde{\boldsymbol{\theta}}_k$ returning a negative test result at time t, where $t > s$, is given by

$$\Phi\left(\frac{\gamma - v(t-s, \tilde{\boldsymbol{\theta}}_k)}{g(t, \tilde{\boldsymbol{\theta}}_k, \hat{\boldsymbol{\phi}})}\right),$$

where Φ denotes the cumulative distribution function of the standard normal distribution. Reported limits of detection of the PCR tests for SARS-CoV-2 have a wide range, we will use $\gamma = 4$ which appears to be the most common. Note that this could be altered

depending on the specific context and requirements of the simulation experiment.

A policy that involves testing every individual at d time points given by $t_1, \ldots, t_d$, is denoted by $p_{t_1,\ldots,t_d}$. For example, p_3 denotes a policy where every individual is tested on day 3, whilst $p_{3,5}$ denotes a policy where every individual is tested on both day 3 and day 5. The probability that an individual returns at least one positive test for a particular policy is equal to 1 minus the probability that the individual returns a negative test result at all time points in the policy. For individual k, the probability of returning at least one positive test under the policy $p_{t_1,\ldots,t_d}$ is given by

$$1 - \prod_{i=1}^{d} \Phi\left(\frac{\gamma - v(t_i - s, \tilde{\boldsymbol{\theta}}_k)}{g(t_i, \tilde{\boldsymbol{\theta}}_k, \hat{\boldsymbol{\phi}})}\right).$$

We can consider how the policy $p_{t_1,\ldots,t_d}$ performs across all simulated individuals by computing the mean probability of an individual returning at least one positive test under the policy, thus our simulation output is

$$\frac{1}{K}\sum_{k=1}^{K}\left[1 - \prod_{i=1}^{d} \Phi\left(\frac{\gamma - v(t_i - s, \tilde{\boldsymbol{\theta}}_k)}{g(t_i, \tilde{\boldsymbol{\theta}}_k, \hat{\boldsymbol{\phi}})}\right)\right].$$

This performance measure might be used to help successfully identify infected individuals and implement effective isolation measures.

We simulate the performance of 22 policies, which are split into two sets. The first set consists of policies $p_{t_1}, p_{t_1,t_2}, p_{t_1,t_2,t_3}$ where $t_2 = t_1 + 2$ and $t_3 = t_1 + 4$ for $t_1 = 3, 4, 5, 6$. These policies might be simulated to study how increased testing affects policy performance, we will refer to them as the multiple testing policies. The second set consists of policies p_3 and p_{3,t_2}, for $t_2 = 4, \ldots, 14$. These policies might be simulated to help decide when to conduct a follow-up test on individual, we will refer to these as the follow-up testing policies.

4.4.3 Data Collection Strategies

We are interested in comparing data collection strategies where the total number of viral load observations are the same, but the observations are split differently in terms of the number of individuals observed and the number of observations per individual. We consider three strategies, all of which use 350 observations of viral load. The first takes 10 observations from 35 individuals, the second takes 7 observations from 50 individuals, and the third takes 5 observations from 70 individuals. These will be referred to as the 35×10, 50×7, and 70×5 strategies respectively. For each strategy, individuals are observed at equally spaced time points, with their first observation taken at a randomised time after s such that their final observation is taken before $21 + s$.

4.4.4 Results

We now have our input model functions and parameters, simulation output, testing policies, and different data collection strategies to compare. For each strategy we generate a data set of viral load observations as appropriate from the true input model and estimate the input model parameters using the R package developed by Pinheiro et al. (2020) which implements the computational methods described in Lindstrom and Bates (1990). These input parameters drive the simulation model in the nominal experiment, where each policy performance is estimated across $n = 100$ replications, each of which simulates $K = 10000$ individuals. We then approximate input uncertainty and stochastic estimation error for each policy performance via a diagnostic experiment where the data is bootstrapped $B = 250$ times, and where for each bootstrap the same number of replications and individuals are used as in the nominal experiment. This gives us an estimate of policy performance, input uncertainty and stochastic estimation error for each of the 22 policies.

Measures of input uncertainty and stochastic estimation error are typically used to construct confidence intervals for simulation outputs. We use the asymptotic nor-

mality theory method described in Cheng and Holland (2004), however we make a slight adjustment. Since the simulation output is a probability, we construct 95% confidence intervals using normality theory on the logarithm of one minus the bootstrapped outputs, and then transform the intervals back to the original scale. The confidence intervals for each policy performance produced by the three data collection strategies are shown in Figure 4.4.1. Note that the stochastic estimation error is very small, so the uncertainty in policy performance is primarily due to input uncertainty.

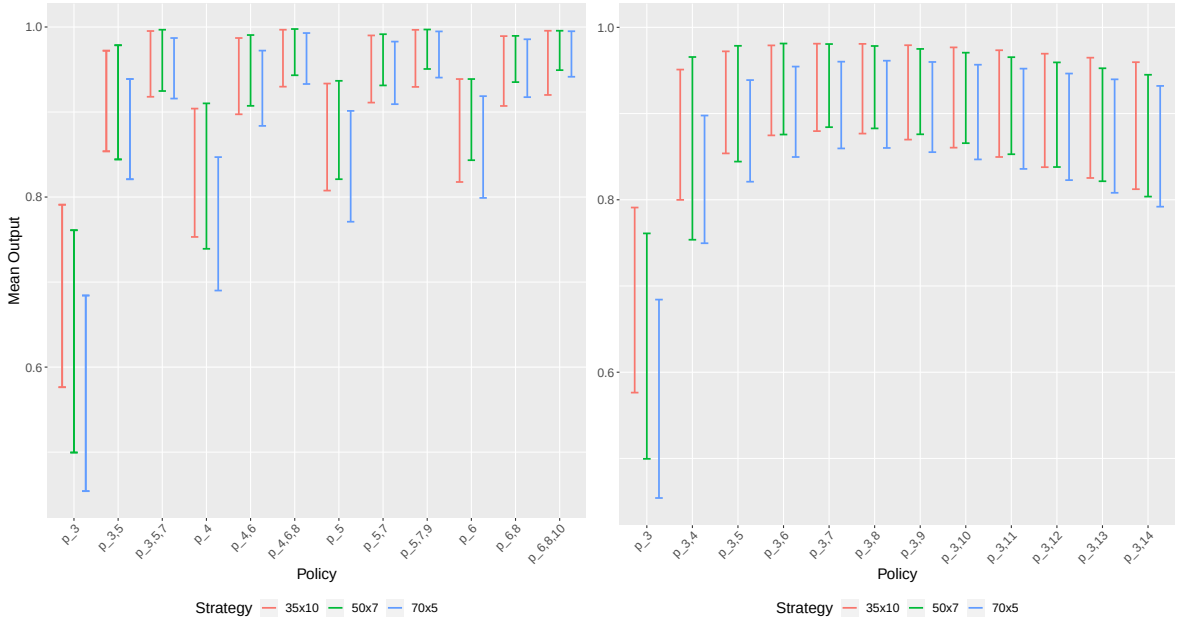

Figure 4.4.1: Interval plot showing the confidence intervals for each data collection strategy across the two sets of testing policies, from a single macro replication.

We can see that the widths of the confidence intervals and their estimates of the performance measure differ for each strategy. In this case, the 35×10 strategy has the smallest input uncertainty for policies p_3, $p_{3,5}$, p_4 and $p_{3,4}$, whilst the 50×7 strategy has the smallest input uncertainty for all other policies. Note that policies that test more frequently have a reduced level of input uncertainty, and this translates to narrower confidence intervals. To understand the behaviour of each strategy more generally, we conduct 100 macro replications, that is, we repeat this experiment 100 times. First, we compare the average input uncertainty estimates for the two sets of policies across the

three data collection strategies.

(a) Multiple testing policies

Policy	35×10	50×7	70×5
p_3	39.7	33.9	32.4
$p_{3,5}$	12.6	10.8	10.7
$p_{3,5,7}$	4.79	3.96	3.74
p_4	21.7	18.6	18.4
$p_{4,6}$	7.35	6.21	6.00
$p_{4,6,8}$	3.60	2.85	2.61
p_5	13.6	11.7	11.6
$p_{5,7}$	5.30	4.28	3.97
$p_{5,7,9}$	3.23	2.42	2.16
p_6	10.8	9.19	9.01
$p_{6,8}$	4.85	3.72	3.33
$p_{6,8,10}$	3.42	2.46	2.15

(b) Follow-up testing policies

Policy	35×10	50×7	70×5
p_3	39.7	33.9	32.4
$p_{3,4}$	21.6	18.6	18.9
$p_{3,5}$	12.6	10.8	10.7
$p_{3,6}$	9.03	7.68	7.43
$p_{3,7}$	7.79	6.50	6.17
$p_{3,8}$	7.55	6.20	5.82
$p_{3,9}$	7.82	6.36	5.93
$p_{3,10}$	8.40	6.79	6.30
$p_{3,11}$	9.19	7.40	6.85
$p_{3,12}$	10.1	8.16	7.54
$p_{3,13}$	11.2	9.03	8.35
$p_{3,14}$	12.3	10.0	9.27

Table 4.4.1: Average input uncertainty ($\times 10^{-4}$) for each data collection strategy across 100 macro replications.

From Table 4.4.1 we see that all the multiple testing policies have the smallest average input uncertainty when using the 70×5 strategy and the largest average input uncertainty when using the 35×10 strategy. The same holds for all the follow-up testing policies, apart from policy $p_{3,4}$ where the 50×7 strategy has the smallest average input uncertainty. Although not shown here, the average stochastic estimation error is of order $\times 10^{-7}$ for p_3 and order $\times 10^{-8}$ for all other policies across each of the data collection strategies, and is therefore relatively small compared to input uncertainty. Since input uncertainty and stochastic estimation error combine to give the total variance of a simulation output, these results suggest that the simulation outputs from using the 70×5 data collection strategy will have the smallest variance and will hence offer the greatest level of insight.

As we have seen in Figure 4.4.1, the confidence intervals produced by each data collection strategy can cover quite different output values. Using the true input model parameters to run simulation replications, we are able to estimate the true performance of each policy, and check whether these lie in the confidence intervals. Table 4.4.2 shows

how many of the 100 macro replications produced confidence intervals that contained the true performance measure for each of the three data collection strategies.

(a) Multiple testing policies

Policy	35×10	50×7	70×5
p_3	87	89	65
$p_{3,5}$	87	94	84
$p_{3,5,7}$	93	94	89
p_4	85	92	80
$p_{4,6}$	91	94	86
$p_{4,6,8}$	94	95	89
p_5	84	95	81
$p_{5,7}$	94	95	86
$p_{5,7,9}$	94	96	91
p_6	91	95	82
$p_{6,8}$	94	94	89
$p_{6,8,10}$	96	94	91

(b) Follow-up testing policies

Policy	35×10	50×7	70×5
p_3	87	89	65
$p_{3,4}$	90	94	82
$p_{3,5}$	87	94	84
$p_{3,6}$	87	94	82
$p_{3,7}$	90	93	84
$p_{3,8}$	91	92	84
$p_{3,9}$	93	91	85
$p_{3,10}$	93	92	87
$p_{3,11}$	93	89	86
$p_{3,12}$	92	86	86
$p_{3,13}$	92	87	86
$p_{3,14}$	92	87	85

Table 4.4.2: The number of macro replications out of 100 that produced confidence intervals containing the true performance measure for each data collection strategy.

For the multiple testing policies, we see that generally the 50×7 strategy produces the most confidence intervals that contain the true performance measure. The 70×5 strategy produces many confidence intervals that do not contain the true measure, particularly for the single day testing policies. For the follow-up testing policies, we again see that the 70×5 strategy produces the least amount of confidence intervals containing the true performance measure. For policies p_3 to $p_{3,8}$ the 50×7 strategy has the highest coverage, whilst for policies $p_{3,9}$ to $p_{3,14}$ the 35×10 strategy has the highest coverage. These results suggest that the 50×7 approach performs best in terms of capturing the true performance measure. The results of Table 4.4.1 and Table 4.4.2 combined suggest that the 70×5 approach is reducing the variance the most but around the incorrect point, possibly indicating some input modelling bias. Note that across all policies, the 35×10 strategy has the largest average interval width. The 70×5 strategy has the smallest average interval width for all policies apart from $p_{3,4}$ and $p_{3,5}$, where the 50×7 strategy has the smallest.

We investigate whether the true performance measure lies above or below the confidence intervals and find that there is a very strong tendency for the performance measure to lie above the confidence intervals across all data collection strategies, particularly for the 50×7 and 70×5 strategy. This suggests that the policy performances estimated by the nominal experiments are underestimating the true performance measures, which we confirm by finding a negative bias for each simulated policy performance across every data collection strategy. Following this, we calculate the bias in the input model parameter estimates. Figure 4.4.2 shows the relative bias induced by each data collection strategy across the 100 macro replications for each of the 8 input parameters. By considering relative bias, we are able to make comparisons across parameters.

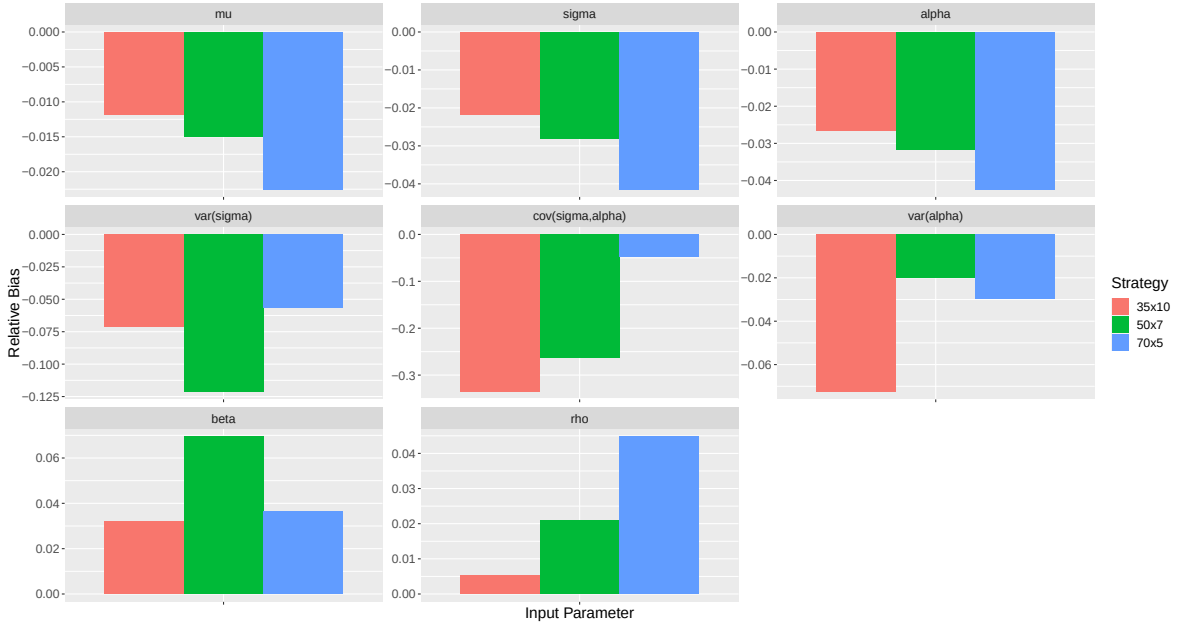

Figure 4.4.2: Bar plots showing the relative bias of each input parameter for the three different data collection strategies, across 100 macro replications

From the top row we see that the relative bias in the population parameters, or the fixed effects, is smallest when using the 35×10 approach and is largest when using the 70×5 approach. Notably, the relative bias is negative for each population parameter and strategy, perhaps indicating a tendency for each of these parameters to be underestimated regardless of how the data is collected. We see that parameters σ and α have a

larger relative bias than μ for each strategy. Conversely the relative bias in estimating the terms of the variance matrix, or the random effects, generally seems to be largest when using the 35×10 strategy and smallest when using the 70×5 strategy. Again, the relative bias is negative for each parameter and strategy, and is particularly large for the covariance of σ and α under the 35×10 and 50×7 strategy. The two terms in the residual error function have the smallest relative bias when using the 35×10 strategy.

Interpreting the results in Figure 4.4.2 is not straightforward, as the relative bias in the estimated parameters occur simultaneously. For example, if we just consider a negatively biased estimate of α then this will clearly result in underestimated simulation outputs since this represents the mean peak height of the viral load profiles. However, a biased estimate of α occurs at the same time as biased estimates of all other input parameters, and we do not know how these interact in terms of affecting the simulation output.

4.5 Discussion

Initially we compared the input uncertainty induced by different data collection strategies and the results in Table 4.4.1 showed that the 70×5 data collection strategy generally had the smallest average input uncertainty across all policies. However, the results in Table 4.4.2 showed that this strategy performs poorly in terms of producing confidence intervals for the simulation outputs that contain the true performance measure, whilst the 50×7 strategy showed the best coverage. We found that in many cases the true performance measures were greater than the upper bounds of the confidence intervals and that the true policy performance was being underestimated by the simulation outputs. Figure 4.4.2 showed that each data collection strategy provided negatively biased estimates of every fixed and random effect of the nonlinear mixed effects model.

Based on the results in this experiment, it appears that the 50×7 strategy is most

suitable, not because it returns a reduced level of input uncertainty across policies compared to the alternative strategies, but because it produces the best coverage of the true performance measure. Clearly, it is inappropriate to solely compare the data collection strategies based on input uncertainty and that the confidence interval coverage should be considered, along with the bias in the simulation outputs. Should the data collection strategies give similar coverage and bias across all policies, then comparing input uncertainty may help to differentiate between them.

With the example discussed, we have seen negatively biased estimates of the fixed and random effects in our input model. Whether this is due to the amount of data being used to fit the input parameters or due to an inherent trait of the model fitting methodology remains to be seen and requires further investigation. Regardless of this, considering how the input parameter estimates produced by each data collection strategy compare to the true input parameters may help us to understand the differences between strategies in terms of replicating the true input model behaviour. In addition, an input parameter sensitivity analysis would indicate which of the input model parameters is most influential in estimating each policy performance, as well as revealing how the simulation outputs are affected by interactions between the parameters.

More generally, there are alternative methods for bootstrapping that could be used, such as the parametric bootstrap as well as bootstrap methods specific to nonlinear mixed effects models. Comparing these methods is difficult however, as the true value of input uncertainty is unknown. There are also alternative techniques for quantifying input uncertainty that could be implemented, which may offer a computational advantage over a bootstrapping procedure. We have seen issues with the simulation outputs behaving differently to the true performance measures, demonstrated by the poor confidence interval coverage shown by some of the data collection strategies. It therefore might be suitable to compare the strategies via the mean squared error due to input modelling, which incorporates input uncertainty and squared input model bias.

4.6 Conclusion

In this chapter, we outline an approach for comparing different data collection strategies via input uncertainty when modelling temporal profiles of viral load and simulating the performance of different testing policies. In particular, we use a nonlinear mixed effects model to represent individual variability in temporal profiles of viral load and use the residual error function to model measurement error. By selecting the functions and parameters of the model, we can generate different sets of viral load observations that represent different strategies for collecting data. For each strategy we use the observations to estimate the model parameters which are used to simulate the performance of different testing policies. The different data collection strategies are compared by the impact of input uncertainty in the simulation outputs.

We demonstrate this approach using a nonlinear mixed effects model, which aims to replicate the behaviour of viral load profiles seen during the COVID-19 pandemic. We simulate the probability of an individual returning at least one positive test across two sets of testing policies and compare data collection strategies which use the same number of total observations but split differently between the number of individuals and the number of observations per individual. In this example, we find that comparing the strategies solely based on input uncertainty is not sensible, as the strategies perform differently in terms of producing confidence intervals for the simulation output that contain the true performance measure. We find a tendency for the true performance measures to be underestimated by the simulation and although this can be linked to negatively biased parameter estimates it requires further investigation through some sensitivity analysis. Future research includes considering alternative techniques for quantifying input uncertainty and incorporating confidence interval coverage and simulation output bias in the approach.

Chapter 5

Input Uncertainty Quantification for Quantiles

Input models that drive stochastic simulations are often estimated from real-world samples of data. This leads to uncertainty in the input models that propagates through to the simulation outputs. Input uncertainty typically refers to the variance of the output performance measure due to the estimated input models. Many methods exist for quantifying input uncertainty when the performance measure is the sample mean of the simulation outputs, however quantiles that are frequently used to evaluate simulation output risk cannot be incorporated into this framework. Here, we adapt two input uncertainty quantification techniques for when the performance measure is a quantile of the simulation outputs rather than the sample mean. The content of this chapter has been published as Parmar et al. (2022), with some minor textual additions made for clarity and consistency with other chapters, as well as some notational adjustment also to maintain consistency with other chapters.

5.1 Introduction

The randomness in stochastic simulation models comes from input models that are typically represented by some probability distributions or processes. Often, these input models are fit using samples of data taken from the real-world system. Since the samples are necessarily finite, the fitted input models are never truly representative of reality. This introduces a source of uncertainty into the simulation model that will propagate through to the outputs. If this uncertainty is not considered in simulation output analysis, then important decisions are at risk of being made with misleading levels of confidence. Input uncertainty broadly refers to the impact of input model uncertainty on simulation outputs. More specifically, input uncertainty quantification aims to quantify the variance in the performance measure due to having estimated the input models. These methods often consider the performance measure to be the sample mean of the simulation outputs, however alternative performance measures might be helpful, for example to learn about the distributional properties of the simulation output or to identify different features between two systems that have similar sample means.

Quantiles are useful for assessing risk. They are particularly common in financial portfolio management, where they are referred to as value at risk. Quantiles can be estimated from simulation outputs using the empirical cumulative distribution function. Existing work on quantile uncertainty quantification accounts for the uncertainty in the quantile estimate due to the finite number of outputs. If the simulation model is driven by input models fitted using real-world data, then there is input model uncertainty which will propagate through the model to the quantile estimate. Current methods do not account for this additional source of error. We are interested in quantifying the effect of input model uncertainty on quantile estimates calculated from simulation outputs. That is, this work aims to quantify input uncertainty when considering a quantile of the simulation outputs, rather than the sample mean.

The rest of the chapter is organised as follows. In Section 5.2 we discuss the input

uncertainty quantification problem for the mean and describe how this changes for quantiles. In Section 5.3, we describe two input uncertainty quantification methods that apply to the mean, and adapt them specifically for the case of quantiles. We run some experiments in Section 5.4 to illustrate the performance of the methods, before concluding in Section 5.5.

5.2 Input Uncertainty

Consider a simulation model driven by L independent input distributions denoted by $\boldsymbol{G} = \{G_1, \ldots, G_L\}$, each of which could be parametric or nonparametric. We represent the output of replication j as

$$Y_j(\boldsymbol{G}) = \eta(\boldsymbol{G}) + \epsilon_j(\boldsymbol{G}),$$

where $\eta(\boldsymbol{G}) = \mathrm{E}[Y_j(\boldsymbol{G})]$ is the expected value of the simulation output random variable and $\epsilon_j(\boldsymbol{G})$ is a random variable with mean 0 representing stochastic noise. Assume there are true input distributions denoted by $\boldsymbol{G}^0 = \{G_1^0, \ldots, G_L^0\}$. Suppose the true input distributions are unknown, but real-world data can be collected from each distribution. Suppose we take a sample of m_l observations from the lth input distribution and compute a collection of fitted input distributions denoted by $\hat{\boldsymbol{G}} = (\hat{G}_1, \ldots, \hat{G}_L)$. A nominal experiment consists of running n i.i.d. simulation replications using the fitted input distributions to obtain outputs $Y_1(\hat{\boldsymbol{G}}), Y_2(\hat{\boldsymbol{G}}), \ldots, Y_n(\hat{\boldsymbol{G}})$.

5.2.1 Input Uncertainty Quantification for the Mean

The original input uncertainty problem assumes the goal of the experiment is to estimate $\eta(\boldsymbol{G}^0)$, the expected value of the simulation output random variable under the true input

distributions. We estimate this via the sample mean of the simulation outputs

$$\bar{Y}(\hat{\boldsymbol{G}}) = \frac{1}{n}\sum_{j=1}^{n} Y_j(\hat{\boldsymbol{G}}) = \eta(\hat{\boldsymbol{G}}) + \frac{1}{n}\sum_{j=1}^{n} \epsilon_j(\hat{\boldsymbol{G}}).$$

The variance of this point estimator is

$$\mathrm{Var}[\bar{Y}(\hat{\boldsymbol{G}})] = \frac{1}{n}\mathrm{E}\big[\mathrm{Var}(\epsilon_1(\hat{\boldsymbol{G}}) \mid \hat{\boldsymbol{G}})\big] + \mathrm{Var}\big[\eta(\hat{\boldsymbol{G}})\big], \tag{5.2.1}$$

where the outer expectation and variance on the right-hand side of the equation are with respect to the sampling distribution of $\hat{\boldsymbol{G}}$. The first term in (5.2.1) measures the expected variability due to the stochastic noise, given the fitted input distributions. We shall define this as *stochastic uncertainty for the mean* and denote with $\sigma^2_{S,M}$

$$\sigma^2_{S,M} = \frac{1}{n}\mathrm{E}\big[\mathrm{Var}(\epsilon_1(\hat{\boldsymbol{G}}) \mid \hat{\boldsymbol{G}})\big]. \tag{5.2.2}$$

This can be driven towards 0 by increasing the number of replications. The second term in (5.2.1) measures the variance in the system mean due to having estimated the input distributions. We shall define this as *input uncertainty for the mean* and denote with $\sigma^2_{I,M}$

$$\sigma^2_{I,M} = \mathrm{Var}\big[\eta(\hat{\boldsymbol{G}})\big]. \tag{5.2.3}$$

This depends on the sample sizes of real-world data used to fit the input distributions, as well as the structure of $\eta(\cdot)$, which is usually unknown. The aim of input uncertainty quantification is to estimate this term.

Various methods have been proposed to quantify input uncertainty for the mean of simulation outputs, for an overview see Song et al. (2014) or Lam (2016). There are two existing methods that we will adapt in this chapter. The first, from Nelson (2013) (Section 7.2), uses bootstrapping to capture the variability of the input distributions and simulation to propagate input model uncertainty. Input uncertainty can be estimated

by subtracting the stochastic uncertainty from the total variability of the bootstrapped outputs. The second method, by Cheng and Holland (1997), considers the case of parametric input distributions. Input uncertainty is modelled using a first-order Taylor series approximation and requires estimates of the parameter variance and the gradient of $\eta(\cdot)$. We choose these two methods as they cover both parametric and nonparametric input models, and provide the foundation for more complex methodology.

5.2.2 Input Uncertainty Quantification for Quantiles

Suppose that instead of estimating the expected value of the simulation output random variable under the true input distributions, we estimate a quantile of the simulation output random variable under the true input distributions. For a random variable Y with a strictly increasing cumulative distribution function (CDF) F, the p-quantile, for $0 < p < 1$, is defined as the constant $\xi_p = F^{-1}(p) = \inf\{y : F(y) \geq p\}$. A common example of a quantile is the median, which is the 0.5-quantile, $\xi_{0.5}$.

Denote the simulation output random variable under the true input distributions by $Y(\boldsymbol{G}^0)$, with CDF F_0. The p-quantile of F_0, for $0 < p < 1$, is given by $\xi_p(\boldsymbol{G}^0) = F_0^{-1}(p) = \inf\{y : F_0(y) \geq p\}$. Assume that F_0 is strictly increasing, differentiable at $\xi_p(\boldsymbol{G}^0)$, and that $f(\xi_p(\boldsymbol{G}^0)) > 0$, where f is the derivative of F. Given n i.i.d. outputs, we can construct an empirical CDF, which can be inverted to obtain a quantile point estimate (Serfling, 2009) (Section 2.3). We can estimate F_0 via the empirical CDF $\hat{F}_n$, defined by

$$\hat{F}_n(y) = \frac{1}{n}\sum_{j=1}^{n} I(Y_j(\hat{\boldsymbol{G}}) \leq y),$$

where $I(\cdot)$ denotes the indicator function. The p-quantile estimator from our nominal experiment is given by $\xi_{p,n}(\hat{\boldsymbol{G}}) = \hat{F}_n^{-1}(p)$. This is equivalent to $\xi_{p,n}(\hat{\boldsymbol{G}}) = Y_{(\lceil np \rceil)}(\hat{\boldsymbol{G}})$, where $Y_{(1)}(\hat{\boldsymbol{G}}) \leq Y_{(2)}(\hat{\boldsymbol{G}}) \leq \cdots \leq Y_{(n)}(\hat{\boldsymbol{G}})$ are the order statistics of the outputs and $\lceil \cdot \rceil$ represents the ceiling function.

Applying the law of total variance to the quantile estimator gives

$$\text{Var}\left[\xi_{p,n}(\hat{\boldsymbol{G}})\right] = \text{E}\left[\text{Var}(\xi_{p,n}(\hat{\boldsymbol{G}})|\hat{\boldsymbol{G}})\right] + \text{Var}\left[\text{E}(\xi_{p,n}(\hat{\boldsymbol{G}})|\hat{\boldsymbol{G}})\right], \tag{5.2.4}$$

where the outer expectation and variance on the right-hand side of the equation are with respect to the sampling distribution of $\hat{\boldsymbol{G}}$. The first term in (5.2.4) measures the expected variability of the quantile estimate given the fitted input distributions. We shall define this as *stochastic uncertainty for the quantile* and denote with $\sigma^2_{S,Q}$

$$\sigma^2_{S,Q} = \text{E}\left[\text{Var}(\xi_{p,n}(\hat{\boldsymbol{G}})|\hat{\boldsymbol{G}})\right]. \tag{5.2.5}$$

This is the uncertainty due to having estimated the quantile via simulation. This will tend towards 0 as the number of replications increases, since given the fitted input distributions, the simulated CDF will approximate the true CDF. The second term in (5.2.4) measures the variance of the expected value of the quantile estimate given the fitted input distributions. We shall define this as *input uncertainty for the quantile* and denote with $\sigma^2_{I,Q}$

$$\sigma^2_{I,Q} = \text{Var}\left[\text{E}(\xi_{p,n}(\hat{\boldsymbol{G}})|\hat{\boldsymbol{G}})\right]. \tag{5.2.6}$$

This measures the uncertainty in the expectation of the quantile estimate due to having estimated the input distributions. This is complex and depends upon the sample sizes used to estimate the input distributions as well as the CDF of the simulation output random variable at the fitted input distributions, which is usually unknown. As this term is different to input uncertainty for the mean in (5.2.3), we will require different methods to quantify it. Note that since the quantile estimator is asymptotically unbiased (Nakayama, 2014), then for large enough n it follows that

$$\text{E}(\xi_{p,n}(\hat{\boldsymbol{G}})|\hat{\boldsymbol{G}}) \approx \xi_p(\hat{\boldsymbol{G}}), \tag{5.2.7}$$

where $\xi_p(\hat{\boldsymbol{G}})$ is the p-quantile of the simulation output random variable under the estimated input distributions.

There is little literature on input uncertainty quantification for quantiles. Zhu et al. (2020) define and provide estimators to quantiles of the mean performance measure under input uncertainty. Xie et al. (2018) develop a Bayesian framework to quantify both the stochastic uncertainty and input uncertainty of percentiles of simulation outputs. Our work aims to quantify input uncertainty for quantiles from a frequentist perspective.

5.3 Methods

We have outlined the original input uncertainty problem and discussed how this changes for quantiles. We now develop two input uncertainty quantification techniques for quantiles. We consider a bootstrapping approach and a Taylor series approximation, the latter of which is restricted to the case of parametric input distributions. For each method we describe its application to the mean followed by our adaptation for quantiles. When we refer to just input uncertainty or stochastic uncertainty in this section, this will be specific to the mean or quantile, depending on the subsection.

5.3.1 Bootstrapping for the Mean

Here we describe the bootstrapping method from Nelson (2013) (Section 7.2) which is used to estimate input uncertainty for the mean. Bootstrapping approximates the sampling distribution of the fitted input models. A single bootstrap consists of three parts. Firstly, for each input distribution, we sample with replacement m_l observations from each set of m_l initial observations. Secondly, these samples are used to estimate bootstrap fitted input distributions. Thirdly, the bootstrap fitted input distributions are used to run simulation replications. Suppose we use B bootstraps. We denote the

bootstrap fitted input distributions by $\hat{\boldsymbol{G}}_k$ for $k = 1, \ldots, B$, and we denote the outputs from the kth bootstrap by $Y_1(\hat{\boldsymbol{G}}_k), \ldots, Y_n(\hat{\boldsymbol{G}}_k)$. This diagnostic experiment requires a total of Bn replications.

Let the mean of the outputs from the kth bootstrap be denoted by $\bar{Y}(\hat{\boldsymbol{G}}_k) = \sum_{j=1}^{n} Y_j(\hat{\boldsymbol{G}}_k)/n$, and let the mean of these means be denoted by $\bar{\bar{Y}} = \sum_{k=1}^{B} \bar{Y}(\hat{\boldsymbol{G}}_k)/B$. The total variance of the mean performance measure from the nominal experiment is estimated by the sample variance of the bootstrapped means

$$\hat{\sigma}^2_{T,M} = \frac{1}{B-1} \sum_{k=1}^{B} (\bar{Y}(\hat{\boldsymbol{G}}_k) - \bar{\bar{Y}})^2.$$

This term approximately measures both input uncertainty and stochastic uncertainty. Stochastic uncertainty is approximated by calculating the sample variance of the outputs in each bootstrap, averaging these across bootstraps, and dividing by a factor of n

$$\hat{\sigma}^2_{S,M} = \frac{1}{n} \left(\frac{1}{B} \sum_{k=1}^{B} \left(\frac{1}{(n-1)} \sum_{j=1}^{n} (Y_j(\hat{\boldsymbol{G}}_k) - \bar{Y}(\hat{\boldsymbol{G}}_k))^2 \right) \right).$$

To estimate input uncertainty we subtract stochastic uncertainty from the total variance

$$\hat{\sigma}^2_{I,M} = \hat{\sigma}^2_{T,M} - \hat{\sigma}^2_{S,M}.$$

Note that this could return a negative estimate of input uncertainty, which is interpreted as meaning that the effect of input uncertainty is relatively small compared to stochastic uncertainty.

5.3.2 Bootstrapping for Quantiles

We now adapt the bootstrapping method for quantiles. For the kth bootstrap, we can compute an empirical CDF from the outputs of the n replications

$$\hat{F}_{n,k}(y) = \frac{1}{n}\sum_{j=1}^{n} I(Y_j(\hat{\boldsymbol{G}}_k) \leq y),$$

and a quantile estimate $\xi_{p,n}(\hat{\boldsymbol{G}}_k) = \hat{F}^{-1}_{n,k}(p)$. Let $\bar{\xi}_{p,n,B} = \sum_{k=1}^{B} \xi_{p,n}(\hat{\boldsymbol{G}}_k)/B$ denote the average of the quantile estimates across bootstraps. The total variance of the quantile estimate from the nominal experiment is approximated by the sample variance of the bootstrapped quantile estimates

$$\hat{\sigma}^2_{T,Q} = \frac{1}{B-1}\sum_{k=1}^{B}(\xi_{p,n}(\hat{\boldsymbol{G}}_k) - \bar{\xi}_{p,n,B})^2.$$

This term approximately measures both input uncertainty and stochastic uncertainty. As previously, we estimate input uncertainty by subtracting an estimate of stochastic uncertainty from the total variance, however we cannot approximate stochastic uncertainty for the quantile in the same way as for the mean. Recall that stochastic uncertainty for the quantile is the expectation of the variance of the quantile estimate with respect to the sampling distribution of $\hat{\boldsymbol{G}}$. The sample variance of the simulation outputs does not provide an approximation to the variance of the quantile estimate, so we require a different method here.

There are myriad ways to approximate the variance of the quantile estimator. The quantile estimator satisfies a central limit theorem (Serfling, 2009) (Section 2.3.3), however the asymptotic variance contains the density function, which is typically unknown. Computing a consistent estimator of the asymptotic variance is non-trivial (Nakayama, 2014). Although finite differences can be used to estimate the density (Serfling, 2009) (Section 2.6.2), this requires specification of a suitable bandwidth parameter. Alter-

natively, bootstrapping methods can be used to directly estimate the variance. The conventional unsmoothed bootstrap is shown to have high relative error (Hall and Martin, 1988), which can be reduced by using a smoothed bootstrap based on a kernel density estimate (Hall et al., 1989). However, this requires stronger smoothness conditions on the density and a suitable choice of smoothing bandwidth. Cheung and Lee (2005) estimate the variance of the quantile estimator using a modification of the bootstrap known as the m out of n bootstrap. Although this requires a choice for m it seems to be less crucial than the choice of the smoothing bandwidth in terms of the sensitivity and stability of the mean squared error of each estimator. Shao and Wu (1989) show that the jackknife estimator with d observations removed gives consistent and asymptotically unbiased estimates of the quantile estimator variance for suitable choices of d.

Alternatively, we can approximate the variance of the quantile estimator by applying batching or sectioning (Asmussen and Glynn, 2007) (Section III.5a). These methods avoid the complication of consistently estimating the density function and are less computationally intensive than bootstrapping procedures, since they only use the results from the nominal experiment. Both involve dividing the outputs into batches and taking quantile estimates from each batch. Batching utilises the variance of the batch quantile estimates, whilst sectioning replaces the sample mean in the variance calculation with the quantile estimator from all the outputs. The batching and sectioning variance divided by the number of batches provides an approximation to the quantile estimator variance.

Whichever method is used, suppose that $\hat{\sigma}_k^2$ represents the variance of the quantile estimate from the kth bootstrap. We estimate stochastic uncertainty by averaging these variance estimates across bootstraps

$$\hat{\sigma}_{S,Q}^2 = \frac{1}{B}\sum_{k=1}^{B}\hat{\sigma}_k^2.$$

To estimate input uncertainty we subtract stochastic uncertainty from the total variance

$$\hat{\sigma}^2_{I,Q} = \hat{\sigma}^2_{T,Q} - \hat{\sigma}^2_{S,Q}.$$

Again note that this could return a negative estimate of input uncertainty which we interpret similar to previously. We now describe the Taylor series approximation for the mean.

5.3.3 Taylor Series Approximation for the Mean

Suppose that the L input distributions follow known parametric distributions. In this case, input model uncertainty becomes input parameter uncertainty, so the input models can be denoted by a set of parameters $\boldsymbol{G} = \boldsymbol{\theta} = (\theta_1, \ldots, \theta_q)$, where $q \geq L$. The true input models are given by the true parameters of the distributions, denoted by $\boldsymbol{G}^0 = \boldsymbol{\theta}^0 = (\theta_1^0, \ldots, \theta_q^0)$. We suppose the parameters are estimated via maximum likelihood estimators (MLEs), which we denote by $\hat{\boldsymbol{\theta}} = (\hat{\theta}_1, \ldots, \hat{\theta}_q)$.

Cheng and Holland (1997) use a Taylor series approximation to estimate input uncertainty when taking the sample mean of simulation outputs. Input uncertainty can be approximated by

$$\hat{\sigma}^2_{I,M} = \nabla\eta(\boldsymbol{\theta}^0)\mathrm{Var}(\hat{\boldsymbol{\theta}})\nabla\eta(\boldsymbol{\theta}^0)^\top,$$

where $\nabla\eta(\boldsymbol{\theta}^0)$ is the gradient of the expected value of the simulation output with respect to the input parameters $\boldsymbol{\theta}$, evaluated at $\boldsymbol{\theta}^0$. This approximation of input uncertainty combines the sensitivity of the expected simulation output with respect to the input parameters, with how accurately the input parameters have been estimated. To use this approximation, we need to estimate both the parameter variance and the gradient of the expected value of the simulation output.

Since the input parameters are estimated via maximum likelihood, we can approxi-

mate the parameter variance by the inverse Fisher information matrix evaluated at the MLEs

$$\widehat{\mathrm{Var}}(\hat{\boldsymbol{\theta}}) = I(\hat{\boldsymbol{\theta}})^{-1}.$$

This follows since the asymptotic distribution of the MLEs is multivariate normal with covariance matrix $I(\boldsymbol{\theta}^0)^{-1}$, which can be consistently estimated by $I(\hat{\boldsymbol{\theta}})^{-1}$. Lin et al. (2015) note that MLEs are not required for the Taylor series approximation method, only that the covariance matrix of the parameter estimates can be approximated, which is most easily done when using MLEs.

There are many ways to estimate the gradient of the expected value of the simulation output. Cheng and Holland (1997) describe the delta method, which employs finite forward differences and requires computational effort that increases linearly with the number of parameters. To improve upon this, they develop the two-point method (Cheng and Holland, 1998), which utilises the delta method but makes most simulation replications at just two settings of parameter values. Lin et al. (2015) provide a method for estimating the gradient that requires no diagnostic experiment, only the replications from the nominal experiment.

Outside the input uncertainty literature, Fu (2006) provides an overview of gradient estimation techniques. Approaches for gradient estimation are divided into two main categories, direct and indirect. Direct approaches aim to estimate the true gradient via some analysis of the underlying mechanism of the simulation model. Methods include perturbation analysis, the likelihood ratio method, and weak derivatives (also known as measure-valued differentiation). Indirect approaches are characterised by two features; they estimate an approximation to the true gradient, and they only utilise evaluations of the simulation model. Methods include finite differences and simultaneous perturbations. Generally, indirect gradient estimators are more widely applicable since direct estimators can involve analysis specific to the problem and may also require some changes to how the simulation model runs. However, direct estimators usually

give unbiased estimators and eliminate the choice of a suitable perturbation parameter.

The gradient estimate can be combined with the parameter variance estimate to employ the Taylor series approximation of input uncertainty. Lin et al. (2015) note that the approximation also provides estimates of the contribution made to input uncertainty by each input distribution. Let $\boldsymbol{\theta}_l$ denote the parameters belonging to the lth input distribution, note that this could a scalar or a vector depending on the distribution. Under the initial assumption that the input distributions are independent, it follows that

$$\nabla\eta(\boldsymbol{\theta}^0)\mathrm{Var}(\hat{\boldsymbol{\theta}})\nabla\eta(\boldsymbol{\theta}^0)^\top = \sum_{l=1}^{L} \nabla\eta(\boldsymbol{\theta}_l^0)\mathrm{Var}(\hat{\boldsymbol{\theta}}_l)\nabla\eta(\boldsymbol{\theta}_l^0)^\top,$$

where $\mathrm{Var}(\hat{\boldsymbol{\theta}}_l)$ is the covariance matrix of $\hat{\boldsymbol{\theta}}_l$. Each term in the summation represents the contribution to input uncertainty from the lth input distribution. These measures can be useful when input uncertainty is large, as we can identify which distributions it would be most beneficial to collect additional data from in order to reduce input uncertainty. Typically, input distributions with the largest contributions would be the best target for additional data collection.

5.3.4 Taylor Series Approximation for Quantiles

We now adapt the Taylor series approximation for quantiles. Applying (5.2.6) and (5.2.7) in the context of known parametric input distributions, for large enough n it follows that

$$\sigma_{I,Q}^2 \approx \mathrm{Var}\left[\xi_p(\hat{\boldsymbol{\theta}})\right],$$

where $\xi_p(\hat{\boldsymbol{\theta}})$ is the p-quantile of the simulation output random variable at the estimated parameters. Subsequently, under the same regularity conditions as stated in Cheng and Holland (1997), we have that

$$\hat{\sigma}_{I,Q}^2 = \nabla\xi_p(\boldsymbol{\theta}^0)\mathrm{Var}(\hat{\boldsymbol{\theta}})\nabla\xi_p(\boldsymbol{\theta}^0)^\top,$$

where $\nabla\xi_p(\boldsymbol{\theta}^0)$ is the gradient of the p-quantile with respect to the input parameters $\boldsymbol{\theta}$, evaluated at $\boldsymbol{\theta}^0$. This approximation of input uncertainty combines the sensitivity of the quantile with respect to the input parameters, with how accurately the input parameters have been estimated. To use this approximation, we need to estimate both the parameter variance and the gradient of the p-quantile. As in the mean case, we can estimate the parameter variance using the inverse Fisher information matrix, however the gradient estimation requires a bit more thought.

A point of difference between the mean and quantiles is the number of estimates we obtain. For the mean, each of the n outputs provides an estimate to the expectation of the simulation output random variable. However, for quantiles, we only obtain a single estimate from a set of n outputs. Consequently, the gradient estimation methods described by Cheng and Holland (1997), Cheng and Holland (1998), and Lin et al. (2015), which can be used for the mean are not applicable for quantiles. Neither are the direct gradient estimators described in Fu (2006), since they are specifically derived for performance measures that are expectations. The indirect gradient estimators described in Fu (2006) however, can be applied when the performance measure is a quantile.

There is a small amount of fairly recent literature on direct gradient estimators for quantiles. Hong (2009) proposes a consistent estimator by combining infinitesimal perturbation analysis with batching. Alternatively, Heidergott and Volk-Makarewicz (2009) present a quantile gradient estimate based on measure-valued differentiation. Liu and Hong (2009) describe a kernel estimator that is consistent and more efficient than Hong (2009). Fu et al. (2009) use conditional Monte Carlo to derive a consistent estimator that does not require batching. More recently, Lei et al. (2018) applied a generalised likelihood ratio method to develop an estimator that also does not require batching. Since these methods all fall under the category of direct gradient estimation they typically require some problem-specific analysis, and consequently they are not applicable to a broad range of simulation models.

Note, we do not advocate for any particular method to be used to estimate the quantile gradient, but in the experiments that follow we use the symmetric difference estimator described in Fu (2006) (Section 3.1). This works by perturbing the value of each parameter in turn, whilst keeping the remaining parameters at their nominal values. In particular, the symmetric difference estimator perturbs each parameter by both a positive and negative difference, and the gradient is estimated by the difference in mean simulation outputs from the perturbed parameter values, divided by twice the size of the difference parameter. Once we have an estimate for the quantile gradient, we can combine this with the parameter variance estimate to employ the Taylor series approximation of input uncertainty. Again this naturally yields the approximate contribution to input uncertainty by each input distribution, where each term in the summation represents the contribution to input uncertainty from the lth input distribution

$$\nabla\xi_p(\boldsymbol{\theta}^0)\mathrm{Var}(\hat{\boldsymbol{\theta}})\nabla\xi_p(\boldsymbol{\theta}^0)^\top = \sum_{l=1}^{L} \nabla\xi_p(\boldsymbol{\theta}_l^0)\mathrm{Var}(\hat{\boldsymbol{\theta}}_l)\nabla\xi_p(\boldsymbol{\theta}_l^0)^\top.$$

5.4 Experiments

We now implement both input uncertainty methods for quantiles on two different examples. In the remainder of this chapter, when we refer to just input uncertainty or stochastic uncertainty, this will be for the quantile. Firstly, we derive an analytical example. This allows us to illustrate the problem of input uncertainty and compare input uncertainty estimates produced by each method against an approximation of the true value. Secondly, we use a stochastic activity network that utilises more input distributions than the analytical example. This allows us to consider more interesting results for the estimated contributions to input uncertainty.

5.4.1 Analytical Example

To illustrate the problem of input uncertainty, we shall create an analytical example. This is contrived to enable input uncertainty to be derived analytically and is not meant to represent a realistic simulation problem. To create an analytical example, we need to be able to compute both $\mathrm{E}(\xi_{p,n}(\hat{\boldsymbol{G}})|\hat{\boldsymbol{G}})$ and $\mathrm{Var}(\xi_{p,n}(\hat{\boldsymbol{G}})|\hat{\boldsymbol{G}})$. If we know the CDF of the simulation output random variable and can derive the inverse CDF, then we can write $\xi_p(\hat{\boldsymbol{G}})$ explicitly. Since the quantile estimator is asymptotically unbiased, this will give us an approximation to $\mathrm{E}(\xi_{p,n}(\hat{\boldsymbol{G}})|\hat{\boldsymbol{G}})$. We can approximate the variance of the quantile estimator given the fitted input models using the well-known asymptotic distribution of the sample quantile (Nelson, 2013) (Section 7.1). Using the asymptotic variance it follows that

$$\mathrm{Var}(\xi_{p,n}(\hat{\boldsymbol{G}})|\hat{\boldsymbol{G}}) \approx \frac{p(1-p)}{nf(\xi_p(\hat{\boldsymbol{G}}))^2}, \tag{5.4.1}$$

where $f(\xi_p(\hat{\boldsymbol{G}}))$ is the probability density function evaluated at the p-quantile under the fitted input models.

Suppose a simulation model has two input models which are known to follow exponential distributions with unknown parameters. The input models are defined by two parameters $\boldsymbol{G} = (\mu, \beta)$ both of which are to be estimated from real-world observations. In this case, input model uncertainty can be thought of as input parameter uncertainty. Suppose we observe $m_1 = m_2 = m$, i.i.d. observations from each distribution. Observations $Z_{1,1}, \ldots, Z_{1,m}$ are used to estimate μ and observations $Z_{2,1}, \ldots, Z_{2,m}$ are used to estimate β. The fitted input models are then given by the estimated parameters $\hat{\boldsymbol{G}} = (\hat{\mu}, \hat{\beta})$. The parameters can be estimated by their MLEs

$$\hat{\mu} = \left(\frac{1}{m}\sum_{i=1}^{m} Z_{1,i}\right)^{-1}, \qquad \hat{\beta} = \left(\frac{1}{m}\sum_{i=1}^{m} Z_{2,i}\right)^{-1}.$$

Suppose that these parameters drive the simulation model for n i.i.d. replications and the distribution of the simulation output random variable is given by $Y \sim \mathrm{Gumbel}(\hat{\mu}, \hat{\beta})$.

Using (5.2.5), (5.4.1), and the probability density function of the Gumbel distribution, stochastic uncertainty for the quantile is approximately

$$\begin{aligned}\sigma^2_{S,Q} &\approx \mathrm{E}\left[p(1-p)n^{-1}\left(e^{-(z+e^{-z})}\hat{\beta}^{-1}\right)^{-2}\right], \\ &\approx p(1-p)n^{-1}\mathrm{E}\left[\left(e^{-(z+e^{-z})}\hat{\beta}^{-1}\right)^{-2}\right], \end{aligned} \tag{5.4.2}$$

where $z = (\xi_p(\hat{\mu},\hat{\beta}) - \hat{\mu})/\hat{\beta}$. Although we cannot compute this expectation analytically, it can be approximated via numerical integration. The expectation term in (5.4.2) will not depend upon the number of outputs n, and therefore stochastic uncertainty will tend towards 0 as n increases. Using (5.2.6), (5.2.7), and the inverse CDF of the Gumbel distribution, input uncertainty for the quantile is approximately

$$\begin{aligned}\sigma^2_{I,Q} &\approx \mathrm{Var}\left[\hat{\mu} - \hat{\beta}\ln(-\ln(p))\right], \\ &\approx \frac{m^2\mu^2}{(m-1)^2(m-2)} + \left(\ln(-\ln(p))\right)^2\frac{m^2\beta^2}{(m-1)^2(m-2)}. \end{aligned} \tag{5.4.3}$$

This follows since if observations $a_1,\ldots,a_m$ are i.i.d. from an exponential distribution with rate θ and $X = \sum_{i=1}^m a_i$, then $1/X \sim$ Inv-Gamma(m,θ) and hence $\mathrm{Var}[1/X] = \theta^2/((m-1)^2(m-2))$. Note that (5.4.3) does not depend on the number of outputs n, but does depend on the number of observations m used to fit the input parameters.

To illustrate the importance of input uncertainty quantification, we will consider the following experiment. Let $\mu = 2$, $\beta = 3$ and $m = 250$. Suppose we use a nominal experiment of $n = 10000$ replications, from which we estimate the 0.95-quantile. If input uncertainty is not considered, then we approximate the variance of our quantile estimate by applying any of the methods described in Section 5.3.2. Suppose we use sectioning. This involves dividing the outputs into batches and calculating the sum of squared errors between the batch quantile estimates and the overall quantile estimate. The variance is then given by the sum of squared errors divided by the number of batches.

We run 1000 macro replications of our nominal experiment, and each time we compute the variance of the quantile estimate using sectioning with 20 batches (Asmussen and Glynn (2007) suggest choosing 30 batches or fewer). The average variance across the macro replications is approximately 0.01812.

The variance of our quantile estimate from the nominal experiment should be given by the sum of stochastic uncertainty and input uncertainty. Using (5.4.2) and (5.4.3), these are approximately

$$\sigma^2_{S,Q} \approx 0.01794, \qquad \sigma^2_{I,Q} \approx 0.3390,$$

where we have used 1×10^5 samples of $(\hat{\mu}, \hat{\beta})$ to estimate the expectation term in (5.4.2). Sectioning provides an approximation of stochastic uncertainty, but does not capture input uncertainty. Input uncertainty is almost 19 times larger, so ignoring it could have serious practical consequences. Although we can derive an analytical approximation of input uncertainty in this particular example, for most realistic simulation problems this is not the case. This motivates the need for methods to quantify input uncertainty.

We use this analytical example to test the accuracy of our two methods. If a nominal experiment uses n replications, and we estimate input uncertainty via B bootstraps, then this diagnostic experiment requires a total of Bn replications. For a fair comparison between the bootstrapping method and the Taylor series approximation method we use the same number of total replications to estimate input uncertainty for each method. Since estimating the parameter variance requires no replications, we utilise Bn replications to estimate the quantile gradient.

We keep $\mu = 2$, $\beta = 3$, $n = 10000$, and compare estimates of input uncertainty for quantiles $p = (0.8, 0.95)$ and input sample sizes $m = (250, 1000)$. For the bootstrapping method we use $B = 10000$ bootstraps and apply sectioning with 10 batches. For the Taylor series approximation, we estimate the quantile gradient using the symmetric difference gradient estimator described in Fu (2006) (Section 3.1), with $c = (0.1, 0.1)$.

This requires simulation runs at 4 sets of parameters, so for each set we use 2.5×10^7 replications. The results from each approach, averaged across 1000 macro replications, are shown in Table 5.4.1, along with the analytical approximation of input uncertainty computed using (5.4.3).

m	Method	$p = 0.8$		$p = 0.95$	
		Mean	Std. Error	Mean	Std. Error
250	Bootstrapping	9.967×10^{-2}	1.564×10^{-2}	3.432×10^{-1}	6.002×10^{-2}
	TSA	9.853×10^{-2}	1.081×10^{-2}	3.389×10^{-1}	4.176×10^{-2}
	Analytical	9.856×10^{-2}	-	3.390×10^{-1}	-
1000	Bootstrapping	2.433×10^{-2}	1.853×10^{-3}	8.356×10^{-2}	7.074×10^{-3}
	TSA	2.433×10^{-2}	1.321×10^{-3}	8.361×10^{-2}	5.068×10^{-3}
	Analytical	2.435×10^{-2}	-	8.373×10^{-2}	-

Table 5.4.1: Mean and standard errors of input uncertainty estimates for the analytical example using bootstrapping and the Taylor series approximation (TSA), compared to an analytical approximation.

For $m = 250$, the Taylor series approximation estimates have a more accurate mean and a smaller standard error than the bootstrapping estimates for both values of p. For $m = 1000$, the bootstrapping estimates and the Taylor series approximation estimates return similarly accurate means for both values of p, although the Taylor series approximation estimates have a smaller standard error. These results show us that both methods are accurately estimating the approximate true value.

5.4.2 Stochastic Activity Network

Additionally, we run experiments using a stochastic activity network, which models the completion time of a project using a group of activities with precedence constraints and random durations. Such models can aid with project planning problems. Specifically, we use the stochastic activity network described in Dong and Nakayama (2014) and Nelson (2013) (Section 3.4). The model consists of $L = 5$ random processes, each of which models the completion time of an activity. The completion times of the activities follow independent exponential distributions, therefore we have $q = 5$ parameters,

denoted by $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5)$. The activities form 3 paths in the network, denoted by $P_1 = \{1, 2\}$, $P_2 = \{1, 3, 5\}$ and $P_3 = \{4, 5\}$. The simulation output is the time to complete the project, which is measured by the longest path in the network. Using A_j to denote the duration of the jth activity, for $1 \leq j \leq 5$, the simulation output is given by $Y = \max\{A_1 + A_2, A_1 + A_3 + A_5, A_4 + A_5\}$. Project planners may be interested in a quantile of the simulation outputs to estimate the probability of completing the project within a certain timeframe.

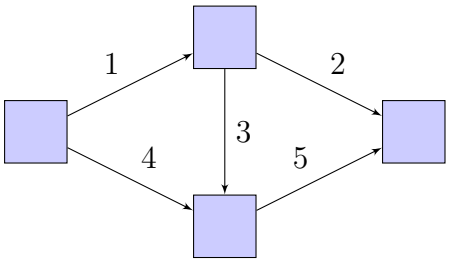

Figure 5.4.1: A graphical representation of the stochastic activity network.

Suppose the true parameters are given by $\boldsymbol{\theta}^0 = (1, 1, 1, 1, 1)$ and each parameter is estimated using the same number of observations, that is $m_l = m$, for $1 \leq l \leq 5$. Again, our nominal experiment consists of $n = 10000$ replications, and we compare estimates of input uncertainty for $p = (0.8, 0.95)$ and $m = (250, 1000)$. For the bootstrapping method we use $B = 10000$ bootstraps and apply sectioning with $b = 10$ batches. For the Taylor series approximation, we use the symmetric difference gradient estimator with $c_l = 0.1$, for $1 \leq l \leq 5$. This requires simulation runs at 10 sets of parameters, so for each set we use 1×10^7 replications. The results from each approach, averaged across 1000 macro replications, are shown in Table 5.4.2.

For 3 of the 4 combinations of m and p, the mean estimates of input uncertainty from the bootstrapping and Taylor series approximation match to 2 decimal places. Although we cannot approximate the true values of input uncertainty, it is reassuring that both methods are returning similar mean estimates. Across all 4 combinations, we see that the Taylor series approximation has a smaller standard error than bootstrap-

m	Method	$p = 0.8$		$p = 0.95$	
		Mean	Std. Error	Mean	Std. Error
250	Bootstrapping	2.044×10^{-2}	2.047×10^{-3}	4.375×10^{-2}	4.924×10^{-3}
	TSA	2.015×10^{-2}	1.549×10^{-3}	4.296×10^{-2}	3.934×10^{-3}
1000	Bootstrapping	4.994×10^{-3}	2.690×10^{-4}	1.060×10^{-2}	6.592×10^{-4}
	TSA	4.971×10^{-3}	1.891×10^{-4}	1.055×10^{-2}	4.831×10^{-4}

Table 5.4.2: Mean and standard errors of input uncertainty estimates for the stochastic activity network using bootstrapping and the Taylor series approximation (TSA).

ping. Although we see this smaller standard error across both experiments, there are too many variables to conclude whether we would expect to see this generally.

Using the results from the Taylor series approximation, we can look at the contributions made to input uncertainty by each input distribution. We calculate the average normalised contribution to input uncertainty by each input distribution across the 1000 macro replications. Table 5.4.3 shows the results for $p = (0.8, 0.95)$ when $m = 1000$.

p	θ_1	θ_2	θ_3	θ_4	θ_5
0.80	32.0	6.4	23.4	6.4	31.8
0.95	33.8	4.1	24.5	4.1	33.5

Table 5.4.3: Average normalised contributions to input uncertainty (%) made by each input parameter in the stochastic activity network, as estimated by the Taylor series approximation.

Firstly, note that for each quantile the normalised contributions to input uncertainty are approximately the same for parameters θ_1 and θ_5, and also for θ_2 and θ_4. We would expect to see this due to the symmetry of the stochastic activity network. We could switch the labels of these activities and the simulation model would remain the same. Secondly, for both quantiles, we see that θ_1 and θ_5 make the largest contributions. We would expect to see this since all activities are identically distributed and both these parameters represent activities that feature in 2 out of the 3 paths in the network. Both also feature in the path with the highest number of activities, which is likely to be the longest path. The second-largest contributions to input uncertainty for both quantiles comes from θ_3, whilst θ_2 and θ_4 return the smallest contributions. Moving from the 0.8-

quantile to the 0.95-quantile, the contributions made by θ_1, θ_3 and θ_5 increase, whilst the contributions made by θ_2 and θ_4 decrease. Parameters θ_1 and θ_5 should be targeted for additional data collection since these make the largest normalised contribution for both quantiles.

5.5 Conclusion

In this work, we considered input uncertainty quantification for quantile performance measures of simulation outputs. This allows us to identify a source of uncertainty in quantile estimates that may previously have been ignored, enabling simulation practitioners to make better-informed decisions.

We focused on the case where input models follow independent distributions and input modelling is done from a frequentist perspective. We adapt two methods of quantifying input uncertainty for the mean, a bootstrapping approach and a Taylor series approximation. The latter is only appropriate for parametric input distributions. We applied both methods to an analytical example, which shows they accurately estimate an analytical approximation of the true value of input uncertainty. We also applied both methods to a stochastic activity network, where they returned similar mean estimates of input uncertainty.

In the future, we should consider how to construct asymptotically valid confidence intervals for the quantile estimator, that account for both stochastic uncertainty and input uncertainty. This will help with the interpretation of input uncertainty for quantiles. We could also consider how other input uncertainty quantification techniques for the mean, which may offer benefits over both methods used here, could be adapted for quantiles. We should also investigate how input uncertainty estimates are impacted when using a smaller number of replications, which would violate the asymptotic relationship in (5.2.7).

Chapter 6

Input Uncertainty in Simulation Analytics

Increasingly, within the field of simulation analytics, predictive machine learning models are being fit to simulation sample path data. The motivation here is to discover factors that drive dynamic performance and enable real-time prediction, both of which can provide insight into and aid with system management and control. These approaches typically treat the simulation input models as fixed, neglecting any possible input model uncertainty, however input models are frequently estimated from real-world samples of data and thus are uncertain. Here, we begin to investigate how such a predictive model might be impacted by input uncertainty.

6.1 Introduction

Simulation output analysis regularly focuses on mean performance measures, or long-run averages, which can mask variable and dynamic behaviour. Simulation analytics aims to utilise the sample path data generated within simulation replications to discover novel insights into the performance of the simulation model, and to assist with real-time decision-making. This usually involves the application of machine learning methods to

the dynamic sample path data to enable a deeper and more nuanced analysis of the simulation model.

Often, the input models used to drive simulation replications are estimated from real-world samples of data. Uncertainty arises in the input models, as they are never truly representative of the random processes of the real-world system. Current approaches that fit machine learning models to simulation sample path data generally treat the input models as fixed. We are interested in investigating how these types of approaches are impacted when estimated input models drive the simulation. We wish to study the impact that input model uncertainty has on the performance of a predictive machine learning model, with a view to learning which uncertain input models most affect the model performance.

The rest of this chapter is organised as follows. In Section 6.2 we discuss some background and motivation and in Section 6.3 we introduce notation and outline the research problem. In Section 6.4 we run an experiment to explore different aspects of the research problem. We conclude the chapter in Section 6.5, providing a summary, a discussion on extensions to the experiment, and outlining areas for future work.

6.2 Background and Motivation

Recently, there has been an interest in using data generated within stochastic simulations, often referred to as state data, to develop a deeper understanding of the simulated system. For example, in a queueing network, the state data might measure the queue size and number of servers in use at each node, at a specific time. There has been a particular focus on fitting classification and regression models to pairs of state and output data from the simulation. For example, Baldwa et al. (2020) fit various classification and regression models to data generated by a neurosurgery ward simulation, to enable prediction of patient admission outcomes and wait times, whilst Jiang et al.

(2020) apply logistic regression to data generated by financial portfolio simulations, to allow for prediction of portfolio risk measures. These types of prediction models can be used to provide interpretation about the system performance, in addition to facilitating real-time predictions which can assist with system control.

Formally, these approaches typically consider observations of system state $\boldsymbol{x}$, paired with a system response y. Across a set of n replications, or by collating data across a single or multiple replications of a steady-state simulation, n pairs of data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$, are obtained. These form the data set from which various predictive models can be trained and validated, using standard techniques from machine learning such as data set transformations, hyperparameter tuning, and cross-validation. The models are evaluated using some selection of scoring metrics, allowing for their performances to be compared and contrasted. The best performing or most appropriate model can then be selected to be used for interpretation and prediction.

Running a simulation model requires specification of the input models, to represent the randomness in the simulation. For example, queueing models require arrival rate and service rate distributions. These are often estimated from real-world data, and hence carry some level of uncertainty that propagates through the simulation model. In many of the examples of simulation analytics reviewed, the input distributions and input parameters are assumed to be known. For example, the wafer fabrication model discussed in Laidler et al. (2020) features 11 stations which have log-normally distributed processing times with known parameters. The experiments on queueing models considered in Ouyang and Nelson (2017) utilise a Poisson arrival process and exponentially distributed service times, the parameters of which are all assumed to be known.

The neurosurgery ward described in Baldwa et al. (2020) uses input distributions to model the arrival rates and length of stays for each patient type. Poisson arrivals are assumed, and these rates are calculated from real-world admissions data. Goodness of fit tests indicate no suitable parametric distribution for the real-world length of stay

data, hence these are modelled using empirical distribution functions. In this case, machine learning models are fitted to the simulation data driven by uncertain input models, however the possible impact of any input model or input parameter uncertainty is not considered.

We are interested in exploring how input uncertainty impacts the performance of a predictive model. In particular, we want to study whether a model fitted to simulation data generated using input models estimated from real-world data has the same level of performance when applied to the real-world system. If the performance of the predictive model on the real-world system is not as expected, then the predictive model may not be applied or utilised in the same way as initially planned.

We are also interested in considering how the performance of the predictive model changes when taking into account the uncertainty of the input models, with a view to learning which input models the predictive model performance is most and least sensitive to. If the performance of the predictive model is very sensitive to an input model that is highly uncertain, then the practitioner may place less confidence in any inference from the predictive model had this analysis not been conducted.

6.3 Research Problem

For simplicity, let us consider a simulation model that has L input models, all of which follow known parametric distributions. Then a parameter vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_p)$, where $p \geq L$, must be specified to run the simulation model. Additionally, let us ignore any kind of controllable inputs (e.g. number of servers) that might be used for experimentation or optimisation. Let the true but unknown input parameters be denoted by $\boldsymbol{\theta}^0 = (\theta_1^0, \ldots, \theta_p^0)$, and suppose that these are estimated from real-world data. Let the estimated input parameters be denoted by $\hat{\boldsymbol{\theta}} = (\hat{\theta}_1, \ldots, \hat{\theta}_p)$, and suppose these are used to run the simulation model.

The n pairs of state-output data generated by the simulation model depend upon the input parameter vector specified in the experiment. Therefore, the following pairs of data are generated $\{(\boldsymbol{x}_i(\hat{\boldsymbol{\theta}}), y_i(\hat{\boldsymbol{\theta}}))\}_{i=1}^n$, where $\boldsymbol{x}_i$ represents the system state information and y_i represents the system responses. These form the data set from which machine learning techniques can be applied, in order to extract insights and understand system behaviour, as part of the goal of simulation analytics. Suppose that various machine learning algorithms are trained and validated on this data set, with the objective of finding the best performing model that can subsequently be used for interpretation and prediction.

Let us consider how the performance of the predictive models are measured. Common practice is to split the data into two sets, where one is used to train the models and the other is used to test the performance of the models. This is done to avoid overfitting the models to the data. The various machine learning procedures can be compared via a particular choice of metrics and scores on the test data set. For example, the performance of classification models might be measured via precision and recall, whilst regression models might use R-squared and mean squared error. A final predictive model can be selected based upon this comparison.

Since the input parameters are uncertain, we want to consider the impact of applying the predictive model to simulation data generated using different input parameter values. Suppose that a different set of parameter values, $\tilde{\boldsymbol{\theta}} = (\tilde{\theta}_1, \ldots, \tilde{\theta}_p)$, are used to run the simulation model. Then n different pairs of state-output data are obtained, and these can be denoted by $\{(\boldsymbol{x}_i(\tilde{\boldsymbol{\theta}}), y_i(\tilde{\boldsymbol{\theta}}))\}_{i=1}^n$. We can measure the performance of the final predictive model, which is fitted to $\{(\boldsymbol{x}_i(\hat{\boldsymbol{\theta}}), y_i(\hat{\boldsymbol{\theta}}))\}_{i=1}^n$, on this new data set by feeding the model instance the state data $\boldsymbol{x}_i(\tilde{\boldsymbol{\theta}})$. We can then compare the predictions against the system responses via the same choice of metrics used to select the final predictive model. Using these results, we can begin to understand how the performance of the predictive model changes, when the input parameters take different values.

We are interested in learning whether the performance of the predictive model significantly changes under the true input parameters, that is, when $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta}^0$, as this is equivalent to applying the predictive model to the real-world system. Although we can test this within a controlled experiment, in reality the true parameters of the simulation model are unknown. We are therefore interested in learning whether the performance of the predictive model significantly changes over the uncertain input parameter space, and whether we can determine the changes to particular input parameters which have driven the changes in predictive performance. We can achieve this by experimenting with various choices of parameter values $\tilde{\boldsymbol{\theta}}$, as informed by the variability of the estimated input parameters, and selecting these appropriately will provide information to aid our understanding of which input parameters the predictive model performance is most sensitive to.

6.4 Experiment

In this section, we run an experiment to explore some of the ideas outlined above. We first fit a classification model to the state-output data from a network queueing model, where the simulation is run using estimated input parameters. We then compare the performance of the model on simulation data from the estimated input parameters versus simulation data from the true input parameters, in a controlled experiment where everything is known to us. Since in reality, the true parameters would be unknown, we consider how an experimental design could be used to examine the impact that different input parameter values have on the predictive performance of the model. Finally, we consider how the comparison against the true parameters and the experimental design results differ, when an increased amount of data is used to estimate the input parameters.

6.4.1 Simulation Analytics on a Network Queueing Model

Here, we experiment with a network queueing model consisting of three consecutive queues, each with a single server. Arrivals to the queue follow a Poisson process, whilst the service times at each queue follow exponential distributions. There are periods of time when the server at the final queue stops serving, and for ease, both the intervals between these periods, and the periods themselves, are constant. The model requires specification of four input parameters, an arrival rate and three service rates, which we denote by $\boldsymbol{\theta} = (\lambda, \mu_1, \mu_2, \mu_3)$. Although simplistic, this type of simulation model could be representative of a manufacturing process, with the final server representing a machine that requires regular maintenance.

For each arrival, we measure the time in the system, and are interested in whether this time falls above or below some threshold. In the context of a manufacturing process, this might define whether jobs or products are completed early or late, for example, see Laidler et al. (2020). To describe the system state for each arrival into the system, we consider the observable components of the system, which consists of whether each server is busy, the number in each queue, and whether the final server is out of action. The data set generated by the simulation model $\{(\boldsymbol{x}_i(\boldsymbol{\theta}), y_i(\boldsymbol{\theta}))\}_{i=1}^n$, is therefore made up of a 7-dimensional measure of system state $\boldsymbol{x}_i$, with an associated binary response y_i, which can be used to fit a classification model.

We set the true input parameters to be $\boldsymbol{\theta}^0 = (4, 6, 8, 6)$, and estimate these by applying maximum likelihood estimation to $m = 500$ observations from each distribution. This gives us a set of estimated input parameters $\hat{\boldsymbol{\theta}} = (4.29, 5.75, 8.16, 5.80)$. We run the simulation model using the estimated input parameters for 2000 units of time, where the third and final server is out of action every 100 time periods, for a length of 5 time periods. We utilise a threshold of 2.5 to classify the time in the system, which results in approximately 28.2% of values falling above the threshold, after discarding a warm-up period from the data set to focus on the steady-state behaviour.

We fit a logistic regression model to the simulation data, using 5-fold cross-validation to evaluate the estimator performance. We evaluate the performance using the F1 score, a popular classification metric which measures the harmonic mean of the precision and recall, that is

$$\text{F1} = 2 \times \frac{precision \times recall}{precision + recall}.$$

Precision measures the number of true positives out of the total positive predictions, and recall (sensitivity) measures the number of true positives out of the total positive results. Positive here refers to points of class 1. For the logistic regression model, we find an average F1 score of 0.8280.

Note that we could extend this classification model fitting process by tuning hyperparameters, fitting more complex models, or by creating additional features for the model to take into account, however this is not the purpose of our experimentation, and thus we have not pursued it further. This would still lead to a similar point where we have a final predictive model with an estimated measure of performance. This is typically where the existing analysis would stop, and the classification model would be used to make real-time predictions, or offer some interpretation about the simulation model.

6.4.2 Model Performance using the True Input Parameters

We now wish to measure the performance of the classification model on simulation data generated using the true input parameters, as this is akin to applying the model to the real-world system. Since we are running a controlled experiment where everything is known to us, this is possible here.

We first generate a sample of F1 scores to summarise the performance of the model on data generated by the estimated input parameters. We run 100 replications of the simulation model with the estimated input parameters, apply the classification model to each of these data sets, and measure the F1 scores. We then repeat this process but

running the simulation model with the true input parameter values. The histogram in Figure 6.4.1 compares the performance of the classification model on simulation data generated using the estimated input parameters and simulation data generated using the true parameters.

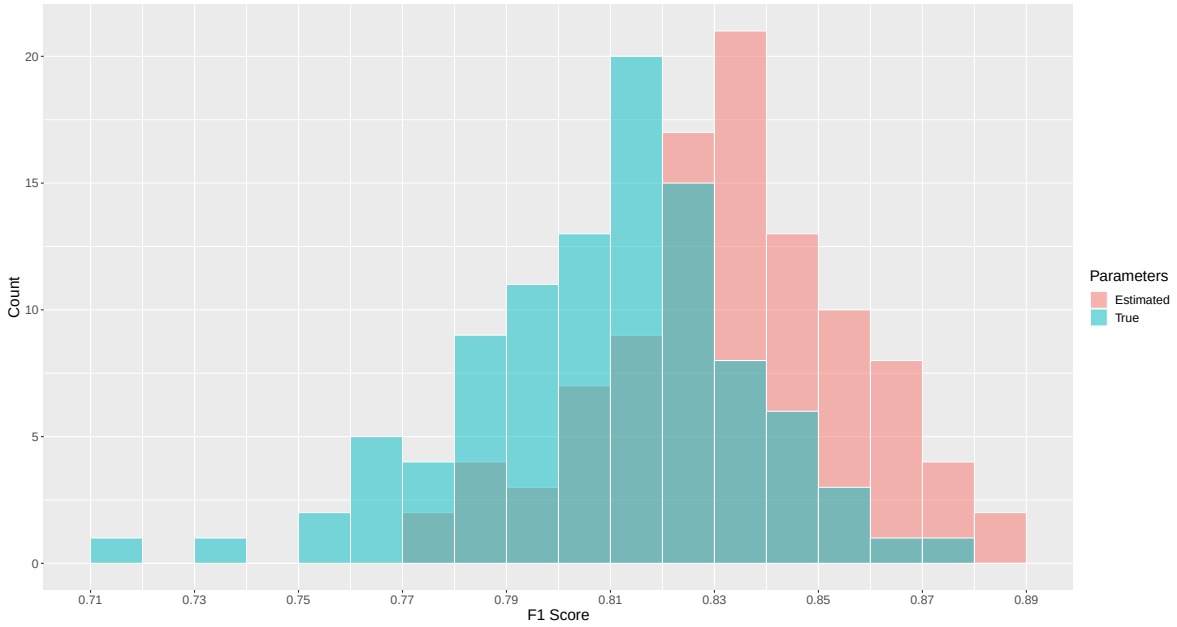

Figure 6.4.1: Histogram comparing the F1 scores from the classification model applied to estimated input parameter and true input parameter simulation data.

We see that the distribution of F1 scores from each set of parameters have a similar shape, each is approximately normal, however the centres of the distribution are different. The distribution of F1 scores from the true input parameters lies to the left of the distribution from the estimated input parameters. This indicates a potential difference in the performance of the classification model on simulation data from the two different sets of parameter values.

To investigate this further, we use a hypothesis test to measure whether there is a significant difference in the performance of the classification model when applied to simulation data driven by the two different sets of input parameter values. Let $(a_1, a_2, \ldots, a_r)$ be the sample of F1 scores from applying the classification model to simulation data generated using the estimated input parameters, and let $(b_1, b_2, \ldots, b_r)$

be the sample of F1 scores from applying the classification model to simulation data generated by the true input parameters.

We want to compare the means of the two independent samples. Our null hypothesis is that the two independent samples have the same mean, $H_0 : \mu_a = \mu_b$, and our alternative hypothesis is that the two samples do not have the same mean, $H_1 : \mu_a \neq \mu_b$. Although the population variances are unknown, the number of samples should be large enough to for normality to approximately hold via the central limit theorem, and this is evidenced by Figure 6.4.1. We can therefore use a two-sample independent z-test, with the test statistic given by

$$z = \frac{(\bar{a} - \bar{b})}{\sqrt{\frac{\sigma_a^2}{r} + \frac{\sigma_b^2}{r}}},$$

where r is the sample size of the F1 scores, and σ_a^2 and σ_b^2 are the population variances that can be approximated by the sample variances.

Applying this hypothesis test to our samples of F1 scores, we find that $z = 6.647$, which returns a p-value of 2.987×10^{-11}. This very small p-value provides strong evidence that the means of the two samples are different, indicating that the performance of the model changes. When applied to simulation data generated by the true input parameters, i.e. the real-world system, the performance of the classification model is significantly different compared to what would be anticipated based on applying it to data generated by the estimated input parameters. If this information were known, then the classification model may not be treated and used for predictions or interpretation in the usual way. This highlights a potential issue with applying simulation analytics techniques to simulation data generated using uncertain input parameters, and suggests that it is worth investigating how the model performs on simulation data generated by various different input parameter values.

6.4.3 Designed Experiment for Input Parameter Uncertainty and Sensitivity

In reality, the true parameters of the simulation are unknown. We therefore would not be able to run such an experiment to learn whether there is a difference in the performance of the predictive model when applied to simulation data generated by the true input parameters. Instead, what we can consider, is how the performance of the model changes when applied to simulation data generated by a range of plausible input parameter values, as determined by the input data. We can apply the same hypothesis test to learn whether there is a difference in model performance for each of the alternative input parameter values compared to the estimated input parameters. If hypothesis tests indicate that the performance of the predictive model frequently changes, then this might decrease the level of confidence placed in any inference from the model. In conjunction with this, we might also want to consider how changes to each input parameter impact the performance of the model.

To understand the impact of changes to each input parameter on the predictive model performance, we need to experiment by running the simulation model with different choices of input parameter values $\tilde{\boldsymbol{\theta}}$. To select these input parameter values, we can utilise techniques from design of experiments (Montgomery, 2017). Using an experimental design will help us to select the parameter values in such a way that we are able to separate the effects of each input parameter, as well as study the impact of any possible interactions between parameters. Moreover, such techniques are often extremely efficient, allowing us to identify the impact of changes to the input parameters whilst using minimal computational effort.

Since the input parameters in our experiment are estimated using maximum likelihood estimation, we can approximate the variance of each parameter by computing the inverse observed Fisher information. This captures the uncertainty of each parameter, so can be used to inform the choice of parameter values to explore in the experimental

design. We decide to use a full factorial design across the four input parameters, where the upper and lower level of each parameter is given by the MLE plus and minus one standard deviation. This gives us a feasible space around the MLEs that the true input parameters are likely to lie in. Each of the 16 design points from the factorial design corresponds to a unique set of input parameter values with which we run the simulation model for 100 replications.

For each of the state-output data sets generated by running the simulation using the design point input parameters, we apply the classification model trained on the estimated input parameters, and measure the performance of the predictions via the F1 score. This gives us a sample of 100 F1 scores for each design point. We apply the hypothesis test to the sample of F1 scores from each design point, to see whether there is evidence that the performance of the model at the design point input parameters is different compared to the performance of the model at the estimated input parameters. Table 6.4.1 shows the design point input parameters for the 16 design points, along with the mean and standard deviation of the F1 scores. Note that the mean F1 score from using the estimated input parameters is 0.833. We also include the p-values from each of the hypothesis tests, shown to three decimal places, comparing the sample of F1 scores from each set of design point input parameters against the sample of F1 scores from using the estimated input parameters. The design points are ordered by their mean F1 score.

It is interesting to see that in some instances, the classification model performs better on simulation data generated using different input parameter values, compared to simulation data generated using the estimated input parameters, since the model is fitted to data from the latter. The mean and standard deviations of the F1 scores exhibit variability across the design points, however they are all around a similar magnitude. The p-values provide evidence that the performance of the model changes for 14 of the 16 design points, indicating that the performance of the model frequently differs

$\tilde{\lambda}$	$\tilde{\mu}_1$	$\tilde{\mu}_2$	$\tilde{\mu}_3$	Mean F1	SD F1	p-value
4.486	5.492	8.525	5.536	0.882	0.018	0.000
4.486	5.492	7.795	5.536	0.880	0.019	0.000
4.486	6.006	8.525	5.536	0.865	0.018	0.000
4.486	6.006	7.795	5.536	0.860	0.020	0.000
4.486	5.492	8.525	6.054	0.857	0.019	0.000
4.486	5.492	7.795	6.054	0.855	0.019	0.000
4.486	6.006	8.525	6.054	0.834	0.021	0.747
4.486	6.006	7.795	6.054	0.829	0.019	0.272
4.101	5.492	8.525	5.536	0.820	0.020	0.000
4.101	6.006	8.525	5.536	0.818	0.024	0.000
4.101	5.492	7.795	5.536	0.817	0.021	0.000
4.101	6.006	7.795	5.536	0.816	0.023	0.000
4.101	6.006	8.525	6.054	0.812	0.022	0.000
4.101	5.492	7.795	6.054	0.811	0.019	0.000
4.101	5.492	8.525	6.054	0.810	0.020	0.000
4.101	6.006	7.795	6.054	0.809	0.026	0.000

Table 6.4.1: Average and standard deviation of the F1 scores from the classification model applied to simulation data generated by each set of design point input parameters, along with the p-value from the hypothesis test, when $m = 500$.

from what we would expect. Knowing this would surely raise a concern that the model might not perform as anticipated on the real-world system. Additionally, we would not be aware of whether the true input parameter values lie within the space captured by the design points or not. These results might suggest that more input data should be collected, or perhaps that the performance of classification models discarded during the model selection process should be tested over the uncertain parameter space to look for more consistent performance.

Note that the design points with the largest mean F1 scores all have $\tilde{\lambda}$ at the upper factor level, suggesting that an increase in this parameter value improves the predictive performance of the model, whilst a decrease in this parameter value worsens the performance. To aid our understanding of the impact of the input parameters on the model performance, we fit a linear regression model using the design point input parameters as the predictor variables, and the mean F1 scores as the response variable. We include the interaction terms in the linear regression model, which returns an R-

squared score of 0.998. Note that we normalise the variables to ensure the regression coefficients are all on the same scale. The model coefficients associated with each input parameter and their interactions are shown in Figure 6.4.2.

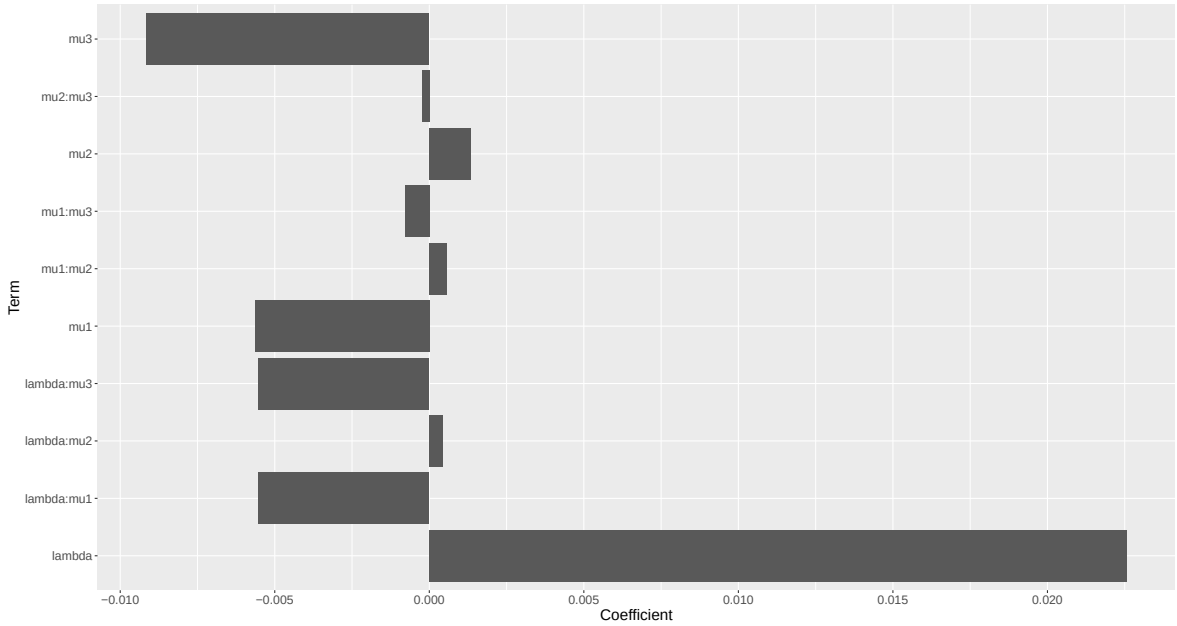

Figure 6.4.2: Bar plot of linear regression coefficients representing the impact of each input parameter and their interactions on the mean F1 score of the classification model.

We see that λ has the coefficient of the largest magnitude, indicating it has the biggest impact on the performance of the predictive classification model. The coefficient is positive, indicating that an increase to this input parameter value positively impacts the F1 score of the model, suggesting an increase in the predictive ability. Conversely, a decrease to λ negatively impacts the F1 score of the model, suggesting a decrease in the predictive ability. The term with the coefficient of the next largest magnitude is μ_3, which is around two fifths the size of the coefficient of λ. The coefficient is negative here, indicating that an increase to μ_3 causes a decrease in the performance of the predictive model. After this, there are three terms, all with negative coefficients of around the same size, these are μ_1, and the interactions between λ and μ_1, and λ and μ_3. Again, an increase in any of these terms suggests a decrease in the predictive model performance. All the terms discussed are statistically significant at a 0.001 significance level, whilst

the remainder of the terms are not.

In the context of the network queueing model, an increase to λ equates to an increase in the number of arrivals to the system, which will result in more congestion. Subsequently, arrivals are likely to spend more time in the system, which may result in a greater number of responses falling above the threshold, thus making it easier to predict such behaviour. An increase to μ_3 equates to a decrease in the service time at the third queue, which is where the largest queues form, resulting in less congestion. This may result in a smaller number of responses falling above the threshold. These results potentially indicate a relationship between the model performance and the class balance (proportion of 0 and 1 binary responses) of the data sets.

We briefly compare these results to those from a traditional sensitivity analysis, which would study the impact of the input parameters on the mean simulation output. We might expect there to be some similarities between how the input parameters affect the simulation output, and how the input parameters affect the performance of the predictive model. Using the simulation results from the experimental design, we measure the sensitivity of the mean time in the system with respect to each input parameter. We fit a linear regression model using the design point input parameters as the predictor variables, again including interaction terms, and the mean time in the system as the response variable. This model produces an R-squared score of 0.999 and the bar plot in Figure 6.4.3 shows the coefficients associated with each term.

We see that the mean simulation output is most sensitive to λ, which aligns with the predictive model F1 score sensitivity results. As previously, this is followed by μ_3, and then μ_1, however the interaction terms that seemed important to the performance of the predictive model seem to have less impact on the mean simulation output. This is an intriguing result, and shows that the impact that the input parameters have on the simulation output is not necessarily the same as the impact they have on the performance of the predictive model. In this case, all the individual parameters are

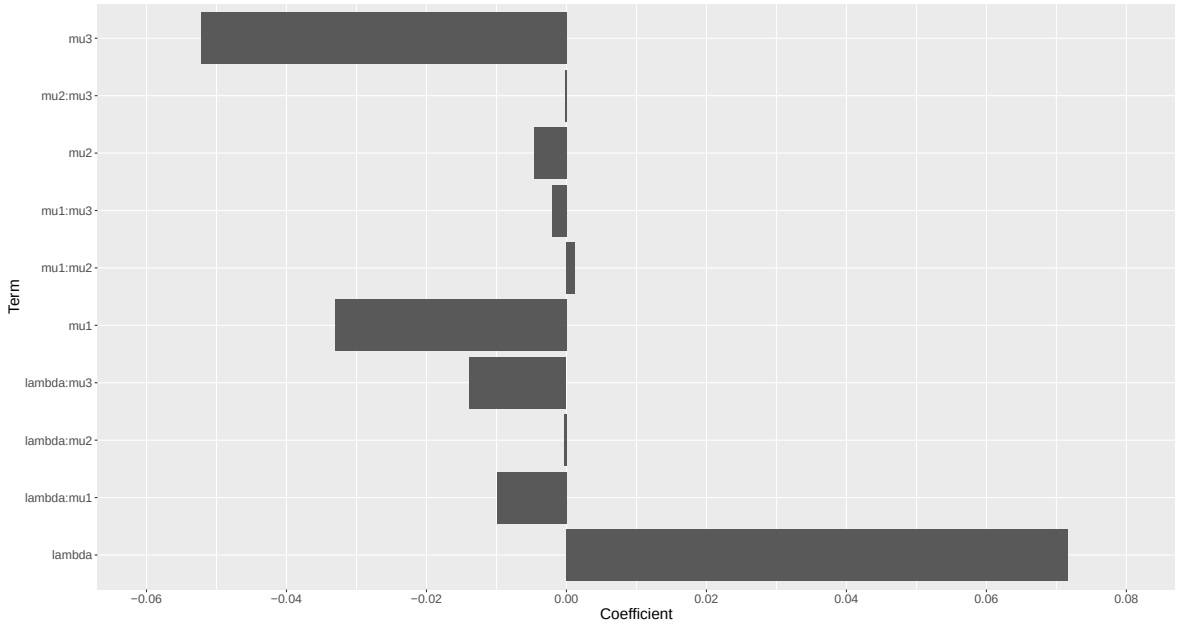

Figure 6.4.3: Bar plot of linear regression coefficients representing the impact of each input parameter on the mean output of the simulation model.

statistically significant at a 0.001 significance level, as well as the interactions between λ and μ_1, and λ and μ_3.

6.4.4 An Increase in Input Data

We now repeat the previous experiment, but using a larger amount of input data. Specifically, we increase the number of observations used to estimate each input parameter from $m = 500$, to $m = 10000$. In this case, we obtain the following set of estimated input parameters $\hat{\boldsymbol{\theta}} = (3.98, 5.91, 8.09, 6.03)$. We run the simulation model using these estimated input parameters, and as previously, fit a logistic regression model to the state-output data.

We apply this classification model to 100 simulation data sets generated by the estimated input parameters, and 100 simulation data sets generated by the true input parameters, measuring the F1 score each time. We then apply the hypothesis test to the two samples to look for a significant difference in performance. In this instance, we find that $z = 0.0158$, which returns a p-value of 0.9874. This provides strong evidence that

the mean F1 score from each sample is equivalent, indicating no significant difference in the performance of the model. This highlights that if enough input data is used to run the simulation, then the predictive model performance in the real-world will be as anticipated based on the simulation results.

We also run the designed experiment as previously. Again, we use a full factorial design across the four input parameters, setting the upper and lower levels by deviating the MLEs by a single standard deviation. Table 6.4.2 shows the design point input parameter values for the 16 design points, along with the mean and standard deviation of the F1 scores, and the p-values from the hypothesis tests. Recall that the hypothesis test compares the performance of the model at the estimated input parameters versus the performance of the model at the design point input parameters. Note that the mean F1 score from using the estimated input parameters is 0.806. Again, we order the design points by their mean F1 score.

$\tilde{\lambda}$	$\tilde{\mu}_1$	$\tilde{\mu}_2$	$\tilde{\mu}_3$	Mean F1	SD F1	p-value
4.020	5.974	8.170	5.978	0.811	0.026	0.143
4.020	5.974	8.170	6.099	0.810	0.024	0.155
4.020	5.856	8.170	6.099	0.809	0.023	0.375
4.020	5.856	8.170	5.978	0.808	0.023	0.449
4.020	5.856	8.008	6.099	0.807	0.022	0.573
3.940	5.974	8.170	5.978	0.807	0.025	0.772
4.020	5.856	8.008	5.978	0.807	0.026	0.786
3.940	5.856	8.170	5.978	0.805	0.023	0.858
3.940	5.856	8.008	6.099	0.805	0.022	0.745
4.020	5.974	8.008	5.978	0.805	0.024	0.732
3.940	5.974	8.008	5.978	0.804	0.026	0.707
3.940	5.974	8.170	6.099	0.804	0.026	0.667
4.020	5.974	8.008	6.099	0.803	0.023	0.493
3.940	5.974	8.008	6.099	0.803	0.023	0.465
3.940	5.856	8.008	5.978	0.802	0.028	0.315
3.940	5.856	8.170	6.099	0.798	0.025	0.032

Table 6.4.2: Average and standard deviation of the F1 scores from the classification model applied to simulation data generated by each set of design point input parameters, along with the p-value from the hypothesis test, when $m = 10000$.

Since the input parameters are estimated from a greater amount of data, the param-

eter variances are much smaller and therefore the design points are much closer than previously. Consequently, the mean F1 scores are less variable, and the performance of the predictive model is much more consistent across the design points. The large p-values provide evidence that the performance of the model at the design point input parameters is equivalent to the performance at the estimated input parameters. That is, we are much more certain about the performance of the model over the uncertain parameter space. This provides some assurance that the performance of the model is likely to be similar to as anticipated when applied to the real-world system.

6.5 Conclusion

We end this chapter by summarising the experiment, methods, and findings. Following this, we provide a discussion around aspects of the experiment that could be extended or given further consideration, and identify some areas that could be investigated in future work.

6.5.1 Summary

In this chapter, we have taken a first step in considering how input uncertainty impacts predictive models fit to simulation data generated by estimated input parameters. We experimented with a network queueing model, and fitted a logistic regression model to classify whether the time in the system was above a certain threshold, using observable state data. We applied this model to data sets generated by the estimated input parameters, to understand the variability in the model performance, as measured by the F1 score. We then applied the model to simulation data generated using the true input parameters, to mimic the application of the classification model to the real-world system. A hypothesis test provided evidence that the performance of the model significantly changed when applied to the true parameter simulation data. When we

repeated this experiment using a greater amount of input data, the hypothesis test provided evidence that there was no such difference in the model performance.

We used an experimental design, to generate sets of input parameters that would explore the parameter space according to their individual levels of uncertainty. By running the simulation model at these sets of input parameters, applying the classification model, and running the hypothesis test, we found that the performance of the model frequently changed over the uncertain parameter space. We used linear regression to model the relationship between the input parameters and the performance of the predictive model, which allowed us to identify which parameters and interactions impacted the model performance the most. We also repeated the experiment using a greater amount of input data, and found that the predictive model performance was much more consistent over the smaller uncertain parameter space.

To summarise, we have shown that a predictive model fitted to simulation data generated by estimated input parameters may not perform as anticipated when applied to the real-world system. We have proposed a hypothesis test as a potential method for identifying whether the performance of the model changes under different input parameter values. We have considered how an experimental design may be used to explore the uncertain parameter space and in conjunction with the hypothesis testing, be used to ascertain whether the performance of the model is as expected or not.

6.5.2 Discussion and Further Work

Although we use the F1 score in our experiment to evaluate the predictive ability of the classification model, we could conduct the same analysis using a different scoring metric, such as the balanced accuracy, or the area under the receiver operating characteristic curve. The hypothesis testing, designed experiment, and sensitivity analysis could be applied in the same way to look for differences in model performance. Note that the approach we have described is invariant to the choice of classification model and

additionally, it could be applied to study the impact of input parameter uncertainty on a predictive regression model, utilising a different scoring metric.

We saw from the F1 score input parameter sensitivity analysis, that the change in model performance might be related to the class balance of the data sets, that is, the proportions of 0 and 1 binary responses. We could investigate this further by measuring the correlation between the class balances of the data sets and the corresponding F1 scores of the classification model. If there is a strong correlation, then the hypothesis testing may be identifying changes to class balance, rather than model performance. In this case, it might be interesting to compare the performance of other classification models, that would be generated during a typical model selection process, over the uncertain parameter space, to see whether they vary in the same way as the logistic regression model.

Predictive models fitted to simulation state-output data might also be used for interpretation. For example, the coefficients of the logistic regression model would inform us how each state variable impacts the probability of being above or below the threshold for time in the system. If decision-making is based on, or influenced by, the model interpretation, then the impact of input parameter uncertainty on the model interpretation should be considered too. Within the existing experiment, we could fit the logistic regression model to state-output data sets from each of the design point input parameters, to compare the model coefficients against those from the original model. This might give us some understanding of how variable the model interpretation is with respect to the uncertain input parameters.

It might be interesting to consider using the regression coefficients to find approximately which input parameter values cause a change in the classification model performance. To do this, we could look at constructing parameter sets, within the parameter space of the experiment, to identify the parameter values where the estimated F1 score of the classification model drops below some threshold, where the threshold may be

motivated by the hypothesis test statistic.

In our experiment, the R-squared score of the linear regression model was close to 1, indicating that the changes to the input parameters explained a large proportion of the variance in the mean F1 scores. If the linear regression model returned a low R-squared score, this would indicate a poor goodness-of-fit. Subsequently, the regression coefficients would offer less explanation and interpretation about the impact of the input parameters on the performance of the predictive model. We could consider including higher-order terms to improve the goodness-of-fit, however the interpretation of the coefficients becomes more complex, and an appropriate experimental design would be required to ensure the effects are not confounded.

We have experimented using a small simulation model, however a more realistic large-scale simulation model with an increased number of input parameters may present some different challenges. The number of design points in a full factorial design grows exponentially with the number of factors, therefore a model with more uncertain input parameters would require substantially more simulation effort to explore all the parameter value combinations. In such an instance, one might choose to reduce the number of design points by using an alternative experimental design, such as a fractional factorial design. These utilise a fraction of the design points from a full factorial design, whilst still obtaining information about the main effects and lower-order interactions (Montgomery, 2017). Furthermore, with an increase in input parameters, fitting and interpreting the subsequent linear regression model with interaction terms becomes difficult. This could be combated by applying the lasso method (Tibshirani, 1996), which performs both variable selection and regularisation to balance model accuracy with simplicity.

In the experiment, we run multiple replications of the simulation model using the estimated input parameters. This is to create state-output data sets on which we can test the classification model, to better gauge the model performance. Creating a

sample of performance metrics is usually not considered, however it is useful for us to understand the variability of the metric due to the stochastic nature of the simulation. It also enables us to compare the model performance for different input parameters via hypothesis testing. It would make sense to utilise this additional simulation data from the estimated input parameters to train a better classification model, rather than solely using it to measure the performance of the existing model. However, this would prevent us from using the data to produce samples of the performance metric required for the hypothesis tests. We should investigate how to use the additional data to fit a better model, whilst still being able to run hypothesis tests to compare the model performance across different input parameter values. Additionally, we might also wish to consider how the simulation data driven by the design point input parameters could be used to build a predictive model that accounts for input parameter uncertainty.

Chapter 7

Conclusion

In this chapter, we conclude the thesis. Firstly, we summarise the contributions made to input uncertainty and data collection problems within the field of stochastic simulation. Secondly, we discuss some generalisations and limitations of our work, as well as the practical implications. Finally, we consider how our research might be extended by identifying some areas for further work.

7.1 Summary of Contributions

In Chapter 3 we addressed our first research problem of whether input uncertainty can be considered prior to any input data collection. Specifically, we presented a two stage algorithm to guide input data collection in a manner that minimises input uncertainty. To the best of our knowledge, this is the first procedure that considers the input uncertainty problem, prior to any type of initial data collection. By collecting data in two stages and assuming the input parameter values fall within some specific intervals, the algorithm aims to hone in towards an allocation of data amongst input distributions that minimises the Taylor series approximation to input uncertainty. Experiments on a simple $M/M/1$ queueing model, from which we could analytically derive the true optimal proportions, indicated that the algorithm was able to achieve close to an op-

timal allocation of input data. Further experiments on a more sophisticated network queueing model demonstrated that the algorithm returned a smaller measure of input uncertainty compared to two other approaches for collecting data.

Chapter 4 addressed our second research problem of developing an approach to compare data collection strategies for viral load simulations. We presented an approach for comparing the input uncertainty passed to the simulated performance of different testing policies, when viral load profiles are estimated using different data collection strategies. Specifically, we used a nonlinear mixed effects model to represent viral load, and used the residual error function to evaluate the probabilities of returning correct or incorrect test results, based on some limit of detection. In our experiment, we found that data collection strategies differed substantially in terms of input uncertainty, however they also varied in confidence interval coverage of the true performance measure. We found that the simulation output was frequently underestimated, so subsequently investigated the input parameter bias. We provided a discussion of our methodology and experiment, and suggested some adaptations and areas for future work that would improve the general approach. This work was motivated by the COVID-19 pandemic, and would potentially be useful for future pandemic responses.

In Chapter 5 we addressed our third research problem of developing frequentist input uncertainty quantification methods for a quantile of the simulation response, as quantiles are often used to evaluate risk. We adapted two existing input uncertainty quantification techniques to consider a quantile of the simulation outputs, rather than the sample mean. Whilst the Taylor series approach is restricted to parametric input distributions, the bootstrapping approach can be used for both parametric and non-parametric input distributions. We constructed a contrived simulation example, from which we could derive an analytical approximation of input uncertainty for the quantile. We used this example to illustrate the problem if input uncertainty for the quantile is ignored, and we also used it to show that the two adapted methods accurately estimate

the correct quantity. Finally, we also compared both approaches on a stochastic activity network, where the Taylor series approximation consistently returned a smaller standard error across estimates of input uncertainty compared to the bootstrapping approach. Since we addressed this research problem by adapting two existing methods from the mean to the quantile, there may be scope to continue this approach of adapting methods to help further address the research problem. This is discussed further in Section 7.3.2.

Chapter 6 began to address our fourth research problem of developing an understanding of the impact of input uncertainty within simulation analytics. Using hypothesis testing, we showed that the performance of a predictive classification model trained on simulation data generated using estimated input parameters, can perform differently than expected when applied to the real-world system. We used a two factorial design to explore the uncertain parameter space and found that the performance of the classification model, as measured by the F1 score, changed frequently compared to the performance on simulation data generated by the estimated input parameters. We also used these results to measure the sensitivity of the predictive model performance with respect to the input parameters. Finally, we found that when an increased amount of input data was used, the performance of the classification model was more consistent over the uncertain parameter space. Since simulation analytics is a growing area, there is more work that can be done to address wider aspects of this new research problem.

7.2 Generalisations, Limitations, and Implications for Practice

In this section, we discuss the generalisability and limitations of the methods developed in this thesis, as well as the implications for practice.

The two stage algorithm developed in Chapter 3 requires specification of intervals

for each of the input parameter values. In practice, this may not always be straightforward, especially if the parameter has no clear interpretation. In this case, graphical representations of the input distribution may help to select the parameter intervals, and it is worth reiterating that the algorithm works regardless of whether the parameters fall in their intervals or not. The solution to two optimisation problems form the basis of the algorithm developed, however these solutions are generalisable in that they may be utilised in alternative ways to help reduce input uncertainty. For example, the solution to (3.4.4) may be used solely for an additional data collection problem. The practical implication of the algorithm, that allocates an initial collection of input data to minimise input uncertainty, is that more precise simulation results can be generated compared to alternative data collection approaches.

The approach developed in Chapter 4 is generalisable, in that it could be adapted and utilised for novel viruses by selecting appropriate functions and parameters. However, it is limited currently without incorporating bias into the data collection strategy comparisons. In developing this approach we found that existing methodologies for collecting data to reduce input uncertainty do not account for complex input models like the nonlinear mixed effects model used, where there is a temporal component to data collection. This work could therefore give rise to new research on such methodologies to cover these particular cases. In practice, this approach should be able to aid with the understanding of which data collection strategy is most valuable for generating insightful results from the simulation of different testing policies. However, in reality, we may find that the data collected does not conform exactly to the input model process selected, in which case the input data could be utilised to derive a more appropriate choice of input model.

The methods developed in Chapter 5 enable input uncertainty quantification for the quantile of a simulation response under frequentist input modelling. Though the Taylor series approach is limited to parametric input distributions, the bootstrapping approach

can be used for both parametric and nonparametric input distributions, however the latter approach can be computationally expensive. In practice, these methods can help practitioners to understand the input uncertainty associated with any quantiles estimated from simulation outputs, an error which would previously be ignored. This improved understanding of the total uncertainty of the quantile estimate should ensure decisions are made with appropriate levels of confidence.

Chapter 6 focused on investigating the impact of input uncertainty on predictive models, fit under the goal of simulation analytics. Our exploratory approach focused on the F1 score of a logistic regression model, but is applicable to any other scoring metric and predictive model. The approach we have described is limited in that it requires the use of parametric input distributions, and becomes computationally expensive with a greater number of input parameters. The results here suggest that in practice, simulation users should take caution when applying a predictive model to the real-world system when the model is fitted using simulation data generated by estimated input parameters, and that they should seek to understand the impact of uncertain input parameters on the performance of the model beforehand.

7.3 Further Work

We now outline possible areas for further work, and describe some additional experiments that may assist with these. We first discuss extending the two stage algorithm to account for multiple simulation outputs and then discuss adapting more input uncertainty quantification techniques for the case of quantiles.

7.3.1 Guiding Data Collection to Minimise Input Uncertainty: Multiple Simulation Outputs

In the two stage algorithm for data collection developed in Chapter 3, the objective is to minimise input uncertainty for a single simulation output, however when running a simulation model, there may be interest in multiple simulation outputs. For example, even in a simple queueing model, there might be interest in both the waiting times and the queue lengths. Generally, more complex simulation models are likely to exhibit lots of interesting behaviour, and thus a greater number of simulation outputs may be recorded. Each output will have a different input uncertainty measure associated with it, and these will be impacted by data collection in contrasting ways. We therefore might like to consider how the two stage algorithm could be extended to include multiple simulation outputs.

To account for the input uncertainties of multiple simulation outputs, we could investigate updating the objective function coefficients in the optimisation problems (3.4.2) and (3.4.4). Currently, these minimise the input uncertainty of a single simulation output. Intuitively, we might alter them to minimise the sum of the input uncertainties associated with each simulation output. An issue we might anticipate here is that no consideration is given to the size of input uncertainty relative to the simulation output, and therefore the optimal proportions are likely to focus on minimising the largest values of input uncertainty. However, smaller values of input uncertainty may be a greater cause for concern if they are large relative to their corresponding simulation output. Therefore, it may be more suitable to minimise the sum of the input uncertainties relative to their simulation output.

We could test out this idea by including extra simulation outputs within our existing examples. With the $M/M/1$ experiment, we should again be able to analytically derive the true optimal proportion for any additional outputs. So, for example, if we decided to consider the mean number in the system, in addition to the mean queueing time, it

would be interesting to see how the analytically derived true optimal proportions that minimise input uncertainty for each of these simulation outputs compare. We could also analytically derive the true optimal proportions that minimise the sum of input uncertainties, as well as those that minimise the sum of input uncertainties relative to their simulation output, to see how these compare. Furthermore, we could then run the two stage algorithm using the updated optimisation problems to see if it is able to provide the trade-off in terms of reaching the optimal allocation of input data to minimise both input uncertainty values.

With the network queueing example, it would be interesting to consider additional simulation outputs within the existing experiment. For example, as well as measuring the average queueing time weighted by type, we could also measure outputs such as the average queueing time of each arrival type, the average queue length at each node, or the server utilisation at each node, to name a few. More specifically, it would be interesting to compare the input uncertainties of these additional simulation outputs under the different data collection approaches, where the two stage algorithm is used to minimise the input uncertainty of the average queueing time weighted by type. For these other simulation outputs which the two stage algorithm does not take into account, we could produce box plots similar to Figure 3.6.3, to see how the algorithm compares against the two other data collection approaches. We could then update the two stage algorithm to account for the multiple simulation outputs and repeat the experiment. This experiment would be fruitful for two reasons. Firstly, to see how the results differ when considering a single output versus multiple outputs, and secondly, to see whether the two stage algorithm is able to produce a reduced level of input uncertainty compared to the other data collection approaches across multiple simulation outputs.

7.3.2 Input Uncertainty Quantification for Quantiles: Adapt More Techniques

Chapter 5 studied input uncertainty quantification for a quantile of the simulation outputs, as opposed to the typical input uncertainty problem which considers the mean. We adapted two existing techniques for input uncertainty quantification of the mean simulation output, to work for a quantile of the simulation outputs.

In the usual input uncertainty context, each of the approaches we have adapted, bootstrapping and the Taylor series approximation, has certain disadvantages. For example, bootstrapping involves a two-layer sampling scheme, which can result in computationally expensive simulation experiments, whilst the Taylor series approach is only a first-order approximation, and requires the use of parametric input models. For these reasons, amongst others, many alternative techniques for quantifying input uncertainty have been developed, each with their own unique set of advantages and disadvantages. We therefore might want to adapt some more techniques so that they can be applied to a quantile of the simulation outputs.

One technique we might be interested in adapting is the approach of Song and Nelson (2015). Here the expected simulation output is modelled as a linear combination of the means and variances of each input distribution, and bootstrapping with empirical cumulative distribution functions is used to estimate the metamodel parameters. This approach is appealing for two reasons. Firstly, it places no restrictions on the types of fitted input distributions used in the nominal simulation experiment. Secondly, from a single diagnostic experiment, the approach is able to estimate input uncertainty, the contribution made to input uncertainty by each input distribution, and the sensitivity of input uncertainty to increases in the sample size of data used to estimate each input distribution. The challenges with adapting this approach would be checking whether the variance decomposition of the model would hold for the case of a quantile, and re-considering the validity of the bootstrap approximation.

Prior to considering which methods to adapt, we could run further experiments with the two existing adapted techniques. Applying these techniques over a wider range of quantiles and input data sizes, as well as utilising a more complex simulation model with an increased number of input models, would be valuable for a couple of reasons. Firstly, it would help us to understand the behaviour of the approaches in more depth, and in particular, we could learn when and why the input uncertainty estimates produced by each approach may suffer. This could enable us to identify certain properties or characteristics that we would like any new methodologies to have, which in turn may inform which existing input uncertainty quantification techniques we would choose to adapt. Secondly, it would give us a benchmark against which we could compare any new methods, to learn whether they do indeed outperform the existing approaches in the scenarios identified, and to see how the methods compare more generally.

Appendix A

Appendix to Chapter 3

A.1 Derivation of Optimal Proportions

We wish to solve the following optimisation problem

$$\{\min \sum_{l=1}^{L} \frac{a_l}{r_l} \quad \text{s.t.} \quad \sum_{l=1}^{L} r_l = 1 \quad \text{and} \quad r_l > 0, \text{ for } l = 1, \ldots, L\},$$

where $a_l = \hat{\boldsymbol{\delta}}(\boldsymbol{\theta}_l)\mathbf{I}_0(\boldsymbol{\theta}_l)^{-1}\hat{\boldsymbol{\delta}}(\boldsymbol{\theta}_l)^\top$ and r_l are the proportions to be optimised. We can write this problem as

$$\{\min \sum_{l=1}^{L} \frac{a_l}{r_l} \quad \text{s.t.} \quad \sum_{l=1}^{L} r_l \leq 1 \quad \text{and} \quad r_l \geq 0, \text{ for } l = 1, \ldots, L\}.$$

We do not need to specify that the proportions sum to 1 or that each proportion is positive, as this will occur at the optimal solution anyway. This problem takes the form of an inequality-constrained nonlinear programme and can be solved to optimality by studying the first-order KKT conditions proved by Karush (1939) and Kuhn and Tucker (1951).

Let μ be the dual multiplier for the first constraint, and λ_l be the dual multiplier

for the following L constraints. The Lagrangian is given by

$$L(\mu, \boldsymbol{\lambda}) = \sum_{l=1}^{L} \frac{a_l}{r_l} - \mu \left(1 - \sum_{l=1}^{L} r_l\right) - \sum_{l=1}^{L} \lambda_l r_l,$$

and the derivative with respect to proportion l is given by

$$\frac{\partial L}{\partial r_l} = \frac{-a_l}{r_l^2} + \mu - \lambda_l.$$

An optimal solution will satisfy, for $l = 1, \ldots, L$, the following four conditions

- *primal feasibility*: $\sum_{l=1}^{L} r_l \leq 1$ and $r_l \geq 0$,
- *dual feasibility*: $\mu \geq 0$ and $\lambda_l \geq 0$,
- *stationarity*: $\partial L / \partial r_l = 0$,
- *complementary slackness*: $\mu \left(1 - \sum_{l=1}^{L} r_l\right) = 0$ and $\lambda_l r_l = 0$.

At an optimal solution, we have $r_l > 0,$ for $l = 1, \ldots, L$. By complementary slackness it follows that $\lambda_l = 0,$ for $l = 1, \ldots, L$. Applying this result to the stationarity conditions we find

$$r_l = \sqrt{\frac{a_l}{\mu}}.$$

Since the sum of proportions will equal 1, it follows that

$$\mu^* = \left(\sum_{l=1}^{L} \sqrt{a_l}\right)^2,$$

and hence the optimal proportions are given by

$$r_l = \sqrt{\frac{a_l}{\left(\sum_{l=1}^{L} \sqrt{a_l}\right)^2}}.$$

A.2 Derivation of Optimal Proportions with Prior Data Collection

We wish to solve the following optimisation problem

$$\{\min \sum_{l=1}^{L} \frac{a_l}{r_l} \quad \text{s.t.} \quad \sum_{l=1}^{L} r_l = 1 \quad \text{and} \quad r_l \geq b_l, \text{ for } l = 1, \ldots, L\},$$

where $a_l = \hat{\boldsymbol{\delta}}(\hat{\boldsymbol{\theta}}_l)\mathbf{I}_0(\hat{\boldsymbol{\theta}}_l)^{-1}\hat{\boldsymbol{\delta}}(\hat{\boldsymbol{\theta}}_l)^\top$, $b_l = m_l/(m+B)$ and r_l are the proportions to be optimised. Similarly to Appendix A.1, we can write this as

$$\{\min \sum_{l=1}^{L} \frac{a_l}{r_l} \quad \text{s.t.} \quad \sum_{l=1}^{L} r_l \leq 1 \quad \text{and} \quad r_l \geq b_l, \text{ for } l = 1, \ldots, L\}.$$

Let μ be the dual multiplier for the first constraint, and λ_l be the dual multiplier for the following L constraints. The Lagrangian is given by

$$L(\mu, \boldsymbol{\lambda}) = \sum_{l=1}^{L} \frac{a_l}{r_l} - \mu \left(1 - \sum_{l=1}^{L} r_l\right) - \sum_{l=1}^{L} \lambda_l (r_l - b_l),$$

and the derivative with respect to proportion l is given by

$$\frac{\partial L}{\partial r_l} = \frac{-a_l}{r_l^2} + \mu - \lambda_l.$$

An optimal solution will satisfy, for $l = 1, \ldots, L$, the following four conditions

- *primal feasibility*: $\sum_{l=1}^{L} r_l \leq 1$ and $r_l \geq b_l$,
- *dual feasibility*: $\mu \geq 0$ and $\lambda_l \geq 0$,
- *stationarity*: $\partial L/\partial r_l = 0$,
- *complementary slackness*: $\mu \left(1 - \sum_{l=1}^{L} r_l\right) = 0$ and $\lambda_l(r_l - b_l) = 0$.

Here we do not know that $r_l \geq b_l$ is not active at the optimal solution, so we cannot say that $\lambda_l = 0$, for $l = 1, \ldots, L$. Instead, let $\{r_l : l \in I\}$ be the proportions for which $r_l > b_l$ in the optimal solution, and let $\{r_l : l \in J\}$ be the proportions for which $r_l = b_l$ in the optimal solution. By stationarity it follows that

$$r_l = \begin{cases} \sqrt{\frac{a_l}{\mu}}, & \text{for } l \in I, \\ \sqrt{\frac{a_l}{\mu - \lambda_l}}, & \text{for } l \in J. \end{cases} \tag{A.2.1}$$

Since $r_l = b_l, \text{ for } l \in J$, by complementary slackness it follows that

$$\lambda_l = \begin{cases} 0, & \text{for } l \in I, \\ \mu - \frac{a_l}{b_l^2}, & \text{for } l \in J. \end{cases} \tag{A.2.2}$$

Since the sum of proportions will equal 1, it follows that

$$\sum_{l \in I} \sqrt{\frac{a_l}{\mu}} + \sum_{l \in J} b_l = 1,$$

and this yields

$$\mu = \left(\frac{\sum_{l \in I} \sqrt{a_l}}{1 - \sum_{l \in J} b_l} \right)^2. \tag{A.2.3}$$

To find an optimal solution, we need to find a partition of the proportions that returns a solution that is both primal and dual feasible. So for each partition (I, J) of $\{1, \ldots, L\}$, calculate

- μ, via (A.2.3),
- λ_l, for $l \in J$, via (A.2.2),
- r_l, for $l = 1, \ldots, L$, via (A.2.1).

If the solution is both primal and dual feasible, then the solution should be optimal.

A.3 Analytical Optimal Proportions for the $M/M/1$ Experiment

The mean queueing (waiting) time of customers in an $M/M/1$ queueing model can be derived using Little's law, and is given by

$$\mathrm{E}[W] = \frac{\lambda}{\mu(\mu - \lambda)}.$$

The gradient of the mean queueing time with respect to each parameter is therefore

$$\frac{\partial \mathrm{E}[W]}{\partial \lambda} = \frac{\mu^2}{(\mu^2 - \mu\lambda)^2}, \qquad \frac{\partial \mathrm{E}[W]}{\partial \mu} = \frac{\lambda^2 - 2\mu\lambda}{(\mu^2 - \mu\lambda)^2}. \tag{A.3.1}$$

Since λ and μ are both estimated using maximum likelihood estimation on sets of exponentially distributed data, their Fisher information is given by

$$I_0(\lambda) = \frac{1}{\lambda^2}, \qquad I_0(\mu) = \frac{1}{\mu^2}. \tag{A.3.2}$$

Input uncertainty can be approximated by combining the gradients in (A.3.1) with the Fisher information for each parameter in (A.3.2), via (3.4.1). This gives

$$\sigma_I^2 \approx \frac{1}{m}\left(\left(\frac{\mu^2}{(\mu^2 - \mu\lambda)^2}\right)^2 \frac{\lambda^2}{r_1} + \left(\frac{\lambda^2 - 2\mu\lambda}{(\mu^2 - \mu\lambda)^2}\right)^2 \frac{\mu^2}{r_2}\right),$$

where m is the overall number of observations and r_1 and r_2 represent the proportions of observations used to estimate λ and μ respectively. For any given values of λ and μ, we can solve for the optimal proportions by using (3.4.3), with

$$a_1 = \lambda^2\left(\frac{\mu^2}{(\mu^2 - \mu\lambda)^2}\right)^2, \qquad a_2 = \mu^2\left(\frac{\lambda^2 - 2\mu\lambda}{(\mu^2 - \mu\lambda)^2}\right)^2.$$

Bibliography

Ankenman, B. E. and Nelson, B. L. (2012). A quick assessment of input uncertainty. In *Proceedings of the 2012 Winter Simulation Conference*, pages 1–10. IEEE.

Asmussen, S. and Glynn, P. W. (2007). *Stochastic simulation: Algorithms and analysis*. Springer Science & Business Media.

Backer, J. A., Klinkenberg, D., and Wallinga, J. (2020). Incubation period of 2019 novel coronavirus (2019-nCoV) infections among travellers from Wuhan, China, 20–28 January 2020. *Eurosurveillance*, 25(5).

Baldwa, V., Sehgal, S., Ramamohan, V., and Tandon, V. (2020). A combined simulation and machine learning approach for real-time delay prediction for waitlisted neurosurgery candidates. In *Proceedings of the 2020 Winter Simulation Conference*, pages 956–967. IEEE.

Barton, R. R. (2012). Tutorial: Input uncertainty in outout analysis. In *Proceedings of the 2012 Winter Simulation Conference*, pages 1–12. IEEE.

Barton, R. R., Lam, H., and Song, E. (2018). Revisiting direct bootstrap resampling for input model uncertainty. In *Proceedings of the 2018 Winter Simulation Conference*, pages 1635–1645. IEEE.

Barton, R. R., Lam, H., and Song, E. (2022). Input uncertainty in stochastic simulation. In *The Palgrave Handbook of Operations Research*, pages 573–620. Springer.

Barton, R. R., Nelson, B. L., and Xie, W. (2010). A framework for input uncertainty analysis. In *Proceedings of the 2010 Winter Simulation Conference*, pages 1189–1198. IEEE.

Barton, R. R., Nelson, B. L., and Xie, W. (2014). Quantifying input uncertainty via simulation confidence intervals. *INFORMS Journal on Computing*, 26(1):74–87.

Barton, R. R. and Schruben, L. W. (1993). Uniform and bootstrap resampling of empirical distributions. In *Proceedings of the 1993 Winter Simulation Conference*, pages 503–508. IEEE.

Barton, R. R. and Schruben, L. W. (2001). Resampling methods for input modeling. In *Proceedings of the 2001 Winter Simulation Conference*, pages 372–378. IEEE.

Biller, B. and Corlu, C. G. (2011). Accounting for parameter uncertainty in large-scale stochastic simulations with correlated inputs. *Operations Research*, 59(3):661–673.

Biller, B. and Gunes, C. (2010). Introduction to simulation input modeling. In *Proceedings of the 2010 Winter Simulation Conference*, pages 49–58. IEEE.

Brailsford, S. C. (2007). Tutorial: Advances and challenges in healthcare simulation modeling. In *Proceedings of the 2007 Winter Simulation Conference*, pages 1436–1448. IEEE.

Chen, M., Liu, Z., and Lam, H. (2022). Distributional input uncertainty. In *Proceedings of the 2022 Winter Simulation Conference*, pages 2617–2628. IEEE.

Chen, P. Z., Bobrovitz, N., Premji, Z., Koopmans, M., Fisman, D. N., and Gu, F. X. (2021). Heterogeneity in transmissibility and shedding SARS-CoV-2 via droplets and aerosols. *Elife*, 10:e65774.

Cheng, R. C. (1994). Selecting input models. In *Proceedings of the 1994 Winter Simulation Conference*, pages 184–191. IEEE.

Cheng, R. C. (1995). Bootstrap methods in computer simulation experiments. In *Proceedings of the 1995 Winter Simulation Conference*, pages 171–177. IEEE.

Cheng, R. C. and Holland, W. (1997). Sensitivity of computer simulation experiments to errors in input data. *Journal of Statistical Computation and Simulation*, 57(1-4):219–241.

Cheng, R. C. and Holland, W. (1998). Two-point methods for assessing variability in simulation output. *Journal of Statistical Computation Simulation*, 60(3):183–205.

Cheng, R. C. and Holland, W. (2004). Calculation of confidence intervals for simulation output. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 14(4):344–362.

Cheung, K. Y. and Lee, S. M. S. (2005). Variance estimation for sample quantiles using the m out of n bootstrap. *Annals of the Institute of Statistical Mathematics*, 57(2):279–290.

Chick, S. E. (2001). Input distribution selection for simulation experiments: Accounting for input uncertainty. *Operations Research*, 49(5):744–758.

Cioppa, T. M., Lucas, T. W., and Sanchez, S. M. (2004). Military applications of agent-based simulations. In *Proceedings of the 2004 Winter Simulation Conference*, pages 171–180. IEEE.

Corlu, C. G., Akcay, A., and Xie, W. (2020). Stochastic simulation under input uncertainty: A review. *Operations Research Perspectives*, 7:100–162.

Currie, C. S., Fowler, J. W., Kotiadis, K., Monks, T., Onggo, B. S., Robertson, D. A., and Tako, A. A. (2020). How simulation modelling can help reduce the impact of covid-19. *Journal of Simulation*, 14(2):83–97.

Dantas, J. P. A., Costa, A. N., Medeiros, F. L. L., Geraldo, D., Maximo, M. R. O. A., and Yoneyama, T. (2022). Supervised machine learning for effective missile launch based on beyond visual range air combat simulations. In *Proceedings of the 2022 Winter Simulation Conference*, pages 1990–2001. IEEE.

Dong, H. and Nakayama, M. K. (2014). Constructing confidence intervals for a quantile using batching and sectioning when applying Latin hypercube sampling. In *Proceedings of the 2014 Winter Simulation Conference*, pages 640–651. IEEE.

Efron, B. and Tibshirani, R. (1986). Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Statistical Science*, 1(1):54–75.

Feng, B. M. and Song, E. (2019). Efficient input uncertainty quantification via green simulation using sample path likelihood ratios. In *Proceedings of the 2019 Winter Simulation Conference*, pages 3693–3704. IEEE.

Freimer, M. and Schruben, L. (2002). Collecting data and estimating parameters for input distributions. In *Proceedings of the 2002 Winter Simulation Conference*, pages 392–399. IEEE.

Fu, M. C. (2006). Gradient estimation. *Handbooks in Operations Research and Management Science*, 13:575–616.

Fu, M. C., Hong, L. J., and Hu, J.-Q. (2009). Conditional Monte Carlo estimation of quantile sensitivities. *Management Science*, 55(12):2019–2027.

Glynn, P. W. and Lam, H. (2018). Constructing simulation output intervals under input uncertainty via data sectioning. In *Proceedings of the 2018 Winter Simulation Conference*, pages 1551–1562. IEEE.

Griffiths, J. D., Price-Lloyd, N., Smithies, M., and Williams, J. E. (2005). Modelling the requirement for supplementary nurses in an intensive care unit. *Journal of the Operational Research Society*, 56(2):126–133.

Günal, M. M. and Pidd, M. (2010). Discrete event simulation for performance modelling in health care: A review of the literature. *Journal of Simulation*, 4:42–51.

Hall, P., DiCiccio, T. J., and Romano, J. P. (1989). On smoothing and the bootstrap. *The Annals of Statistics*, pages 692–704.

Hall, P. and Martin, M. A. (1988). Exact convergence rate of bootstrap quantile variance estimator. *Probability Theory and Related Fields*, 80(2):261–268.

Heidergott, B. and Volk-Makarewicz, W. (2009). Quantile sensitivity estimation. In *International Conference on Network Control and Optimization*, pages 16–29. Springer.

Henderson, S. G. (2003). Input model uncertainty: Why do we care and what should we do about it? In *Proceedings of the 2003 Winter Simulation Conference*, pages 90–100. IEEE.

Hong, L. J. (2009). Estimating quantile sensitivities. *Operations Research*, 57(1):118–130.

Jahangirian, M., Eldabi, T., Naseer, A., Stergioulas, L. K., and Young, T. (2010). Simulation in manufacturing and business: A review. *European Journal of Operational Research*, 203(1):1–13.

Jiang, G., Hong, L. J., and Nelson, B. L. (2020). Online risk monitoring using offline simulation. *INFORMS Journal on Computing*, 32(2):356–375.

Karush, W. (1939). Minima of functions of several variables with inequalities as side constraints. *MSc Dissertation. Dept. of Mathematics, Univ. of Chicago.*

Kim, T. and Song, E. (2022). Optimizing input data acquisition for ranking and selection: A view through the most probable best. In *Proceedings of the 2022 Winter Simulation Conference*, pages 2258–2269. IEEE.

Kucirka, L. M., Lauer, S. A., Laeyendecker, O., Boon, D., and Lessler, J. (2020). Variation in false-negative rate of reverse transcriptase polymerase chain reaction–based SARS-CoV-2 tests by time since exposure. *Annals of Internal Medicine*, 173(4):262–267.

Kuhn, H. W. and Tucker, A. W. (1951). Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*. University of California Press, Berkeley.

Laidler, G., Morgan, L. E., Nelson, B. L., and Pavlidis, N. G. (2020). Metric learning for simulation analytics. In *Proceedings of the 2020 Winter Simulation Conference*, pages 349–360. IEEE.

Lam, H. (2016). Advanced tutorial: Input uncertainty and robust analysis in stochastic simulation. In *Proceedings of the 2016 Winter Simulation Conference*, pages 178–192. IEEE.

Lam, H. (2022). Cheap bootstrap for input uncertainty quantification. In *Proceedings of the 2022 Winter Simulation Conference*, pages 2318–2329. IEEE.

Lam, H. and Qian, H. (2016). The empirical likelihood approach to simulation input uncertainty. In *Proceedings of the 2016 Winter Simulation Conference*, pages 791–802. IEEE.

Lam, H. and Qian, H. (2017). Optimization-based quantification of simulation input uncertainty via empirical likelihood. https://arxiv.org/pdf/1707.05917.pdf, accessed 26th April 2021.

Lam, H. and Qian, H. (2018a). Subsampling to enhance efficiency in input uncertainty quantification. https://arxiv.org/pdf/1811.04500.pdf, accessed 26th April 2021.

Lam, H. and Qian, H. (2018b). Subsampling variance for input uncertainty quantification. In *Proceedings of the 2018 Winter Simulation Conference*, pages 1611–1622. IEEE.

Lam, H. and Qian, H. (2019). Random perturbation and bagging to quantify input uncertainty. In *Proceedings of the 2019 Winter Simulation Conference*, pages 320–331. IEEE.

Lam, H. and Qian, H. (2022). Subsampling to enhance efficiency in input uncertainty quantification. *Operations Research*, 70(3):1891–1913.

Larremore, D. B., Wilder, B., Lester, E., Shehata, S., Burke, J. M., Hay, J. A., Tambe, M., Mina, M. J., and Parker, R. (2021). Test sensitivity is secondary to frequency and turnaround time for COVID-19 screening. *Science Advances*, 7(1):eabd5393.

Lei, L., Peng, Y., Fu, M. C., and Hu, J.-Q. (2018). Applications of generalized likelihood ratio method to distribution sensitivities and steady-state simulation. *Discrete Event Dynamic Systems*, 28(1):109–125.

Lin, Y., Song, E., and Nelson, B. L. (2015). Single-experiment input uncertainty. *Journal of Simulation*, 9(3):249–259.

Lindstrom, M. J. and Bates, D. M. (1990). Nonlinear mixed effects models for repeated measures data. *Biometrics*, pages 673–687.

Liu, G. and Hong, L. J. (2009). Kernel estimation of quantile sensitivities. *Naval Research Logistics (NRL)*, 56(6):511–525.

Lucas, T. W., Kelton, W. D., Sánchez, P. J., Sanchez, S. M., and Anderson, B. L. (2015). Changing the paradigm: Simulation, now a method of first resort. *Naval Research Logistics (NRL)*, 62(4):293–303.

McKay, M. D., Beckman, R. J., and Conover, W. J. (2000). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1):55–61.

Montgomery, D. C. (2017). *Design and analysis of experiments*. John Wiley & Sons.

Morgan, L. E., Nelson, B. L., Titman, A. C., and Worthington, D. J. (2019). Detecting bias due to input modelling in computer simulation. *European Journal of Operational Research*, 279(3):869–881.

Morgan, L. E., Nelson, B. L., Titman, A. C., and Worthington, D. J. (2023). A spline function method for modelling and generating a nonhomogeneous Poisson process. *Journal of Simulation*, pages 1–12.

Morgan, L. E., Titman, A. C., Worthington, D. J., and Nelson, B. L. (2016). Input uncertainty quantification for simulation models with piecewise-constant non-stationary Poisson arrival processes. In *Proceedings of the 2016 Winter Simulation Conference*, pages 370–381. IEEE.

Nakayama, M. K. (2014). Confidence intervals for quantiles using sectioning when applying variance-reduction techniques. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 24(4):1–21.

Nelson, B. (2013). *Foundations and methods of stochastic simulation: A first course*. Springer Science & Business Media.

Nelson, B. L. (2016). 'Some tactical problems in digital simulation' for the next 10 years. *Journal of Simulation*, 10(1):2–11.

Nelson, B. L., Wan, A. T., Zou, G., Zhang, X., and Jiang, X. (2021). Reducing simulation input-model risk via input model averaging. *INFORMS Journal on Computing*, 33(2):672–684.

Ng, S. H. and Chick, S. E. (2001). Reducing input parameter uncertainty for simulations. In *Proceedings of the 2001 Winter Simulation Conference*, pages 364–371. IEEE.

Ng, S. H. and Chick, S. E. (2006). Reducing parameter uncertainty for stochastic systems. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 16(1):26–51.

Oakley, J. E. and O'Hagan, A. (2004). Probabilistic sensitivity analysis of complex models: A Bayesian approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66(3):751–769.

Ouyang, H. and Nelson, B. L. (2017). Simulation-based predictive analytics for dynamic queueing systems. In *Proceedings of the 2017 Winter Simulation Conference*, pages 1716–1727. IEEE.

Parmar, D., Morgan, L. E., Titman, A. C., Regnier, E. D., and Sanchez, S. M. (2021a). Comparing data collection strategies via input uncertainty when simulating testing policies using viral load profiles. In *Proceedings of the 2021 Winter Simulation Conference*, pages 1–12. IEEE.

Parmar, D., Morgan, L. E., Titman, A. C., Williams, R. A., and Sanchez, S. M. (2021b). A two stage algorithm for guiding data collection towards minimising input uncertainty. In *Proceedings of the Operational Research Society Simulation Workshop 2021*, pages 127–136. Operational Research Society.

Parmar, D., Morgan, L. E., Titman, A. C., Williams, R. A., and Sanchez, S. M. (2022). Input uncertainty quantification for quantiles. In *Proceedings of the 2022 Winter Simulation Conference*, pages 97–108. IEEE.

Pinheiro, J., Bates, D., DebRoy, S., Sarkar, D., and R Core Team (2020). *nlme:*

Linear and nonlinear mixed effects models. R package version 3.1-144, https://CRAN.R-project.org/package=nlme.

Pujadas, E., Chaudhry, F., McBride, R., Richter, F., Zhao, S., Wajnberg, A., Nadkarni, G., Glicksberg, B. S., Houldsworth, J., and Cordon-Cardo, C. (2020). SARS-CoV-2 viral load predicts COVID-19 mortality. *The Lancet. Respiratory Medicine*, 8(9):e70.

Quilty, B. J., Clifford, S., Hellewell, J., Russell, T. W., Kucharski, A. J., Flasche, S., Edmunds, W. J., Atkins, K. E., Foss, A. M., Waterlow, N. R., et al. (2021). Quarantine and testing strategies in contact tracing for SARS-CoV-2: A modelling study. *The Lancet Public Health*, 6(3):e175–e183.

Quinn, T. C., Wawer, M. J., Sewankambo, N., Serwadda, D., Li, C., Wabwire-Mangen, F., Meehan, M. O., Lutalo, T., and Gray, R. H. (2000). Viral load and heterosexual transmission of human immunodeficiency virus type 1. *New England Journal of Medicine*, 342(13):921–929.

Rhodes-Leader, L., Worthington, D. J., Nelson, B. L., and Onggo, B. S. (2018). Multi-fidelity simulation optimisation for airline disruption management. In *Proceedings of the 2018 Winter Simulation Conference*, pages 2179–2190. IEEE.

Sanchez, P. J. and Sanchez, S. M. (2015). A scalable discrete event stochastic agent-based model of infectious disease propagation. In *Proceedings of the 2015 Winter Simulation Conference*, pages 151–158. IEEE.

Sanchez, S. M. (2020). Data farming: Methods for the present, opportunities for the future. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 30(4):1–30.

Sanchez, S. M., Sanchez, P. J., and Wan, H. (2020). Work smarter, not harder: A tutorial on designing and conducting simulation experiments. In *Proceedings of the 2020 Winter Simulation Conference*, pages 1128–1142. IEEE.

Sargent, R. G. (2010). Verification and validation of simulation models. In *Proceedings of the 2010 Winter Simulation Conference*, pages 166–183. IEEE.

Serfling, R. J. (2009). *Approximation theorems of mathematical statistics*. John Wiley & Sons.

Shao, J. and Wu, C. J. (1989). A general theory for jackknife variance estimation. *The Annals of Statistics*, pages 1176–1197.

Song, E. and Nelson, B. L. (2013). A quicker assessment of input uncertainty. In *Proceedings of the 2013 Winter Simulation Conference*, pages 474–485. IEEE.

Song, E. and Nelson, B. L. (2015). Quickly assessing contributions to input uncertainty. *IIE Transactions*, 47(9):893–909.

Song, E., Nelson, B. L., and Pegden, C. D. (2014). Advanced tutorial: Input uncertainty quantification. In *Proceedings of the 2014 Winter Simulation Conference*, pages 162–176. IEEE.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288.

Ungredda, J., Pearce, M., and Branke, J. (2022). Bayesian optimisation vs. input uncertainty reduction. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 32(3):1–26.

von Rueden, L., Mayer, S., Sifa, R., Bauckhage, C., and Garcke, J. (2020). Combining machine learning and simulation to a hybrid modelling approach: Current and future directions. In *International Symposium on Intelligent Data Analysis*, pages 548–560. Springer.

Wieland, J. R. and Schmeiser, B. W. (2006). Stochastic gradient estimation using a

single design point. In *Proceedings of the 2006 Winter Simulation Conference*, pages 390–397. IEEE.

Wölfel, R., Corman, V. M., Guggemos, W., Seilmaier, M., Zange, S., Müller, M. A., Niemeyer, D., Jones, T. C., Vollmar, P., Rothe, C., et al. (2020). Virological assessment of hospitalized patients with COVID-2019. *Nature*, 581(7809):465–469.

Xie, W., Nelson, B. L., and Barton, R. R. (2014a). A Bayesian framework for quantifying uncertainty in stochastic simulation. *Operations Research*, 62(6):1439–1452.

Xie, W., Nelson, B. L., and Barton, R. R. (2014b). Statistical uncertainty analysis for stochastic simulation with dependent input models. In *Proceedings of the 2014 Winter Simulation Conference*, pages 674—685. IEEE.

Xie, W., Nelson, B. L., and Barton, R. R. (2016). Multivariate input uncertainty in output analysis for stochastic simulation. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 27(1):1–22.

Xie, W., Wang, B., and Zhang, P. (2019). Metamodel-assisted sensitivity analysis for controlling the impact of input uncertainty. In *Proceedings of the 2019 Winter Simulation Conference*, pages 3681–3692. IEEE.

Xie, W., Wang, B., and Zhang, Q. (2018). Metamodel-assisted risk analysis for stochastic simulation with input uncertainty. In *Proceedings of the 2018 Winter Simulation Conference*, pages 1766–1777. IEEE.

Xu, J., Zheng, Z., and Glynn, P. W. (2020). Joint resource allocation for input data collection and simulation. In *Proceedings of the 2020 Winter Simulation Conference*, pages 2126–2137. IEEE.

Yi, Y. and Xie, W. (2017). An efficient budget allocation approach for quantifying the impact of input uncertainty in stochastic simulation. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 27(4):1–23.

Zhou, E. and Liu, T. (2018). Online quantification of input uncertainty for parametric models. In *Proceedings of the 2018 Winter Simulation Conference*, pages 1587–1598. IEEE.

Zhu, H., Liu, T., and Zhou, E. (2020). Risk quantification in stochastic simulation under input uncertainty. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 30(1):1–24.

Zouaoui, F. and Wilson, J. (2001a). Accounting for input model and parameter uncertainty in simulation. In *Proceedings of the 2001 Winter Simulation Conference*, pages 290–299. IEEE.

Zouaoui, F. and Wilson, J. (2001b). Accounting for parameter uncertainty in simulation input modeling. In *Proceedings of the 2001 Winter Simulation Conference*, pages 354–363. IEEE.

Zouaoui, F. and Wilson, J. R. (2003). Accounting for parameter uncertainty in simulation input modeling. *IIE Transactions*, 35(9):781–792.

Zouaoui, F. and Wilson, J. R. (2004). Accounting for input-model and input-parameter uncertainties in simulation. *IIE Transactions*, 36(11):1135–1151.