NORTHWESTERN UNIVERSITY

Reducing and Measuring Input Model Risk in Stochastic Simulation

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Industrial Engineering & Management Sciences

By

Xi Jiang

EVANSTON, ILLINOIS

September 2020

Acknowledgments

Foremost, I would like to express my deep and sincere gratitude to my advisor Professor Barry L. Nelson for the continuous support and invaluable guidance throughout my Ph.D study and research. His patience, vision, motivation, and immense knowledge have deeply inspired me. It was a great privilege and honor to work and study under his guidance. I would also like to thank him for his friendship, empathy, and great sense of humor. I could not have imagined a better advisor and mentor for my Ph.D study.

Besides my advisor, I would like to thank the rest of my thesis committee: Professor Bruce E. Ankenman and Doctor Bahar Biller, for their insightful comments and encouragement.

My sincere thanks also goes to Professor Jeff L. Hong and SAS Institute for offering me the summer internship opportunities which incented me to widen my research on sensitivity analysis. In addition, I thank Art Owen for suggesting the idea of fitting in the quantile space for frequentist model averaging.

I also would like to thank all the professors who have taught me at Northwestern University and all my colleagues for their support and help over the past years of my Ph.D study.

Last but no least, I am extremely grateful to my family and friends for their love, prayers, caring, understanding, and support for completing this research work and my life in general.

My research is supported by National Science Foundation under Grant Number CMMI-1634982.

Abstract

Simulation studies are virtually all motivated by decision making. Because simulation output is stochastic and input models are never perfect, all decisions should include an accounting for risks. Input model risk refers to the exposure due to imperfect simulation input models that are estimated from real-world data, involving both the level of uncertainty about the input itself and the propagation of uncertainty in the input to the output. This dissertation addresses both issues: reducing and measuring input model risk, with a focus on the latter. For reducing the input model risk, we propose model averaging, which is a weighted average of the candidate distributions in a given set with the weights tuned by cross-validation, and extend the implementation in the probability space to the quantile space that emphasizes the tail behavior. For measuring the input model risk, we propose a family of solutions to measure the local sensitivity of an output property to an input property, focusing on the output mean or variance with respect to the input mean or variance. We extend existing stochastic gradient methods to identify the point and error estimators for any member of the family from the nominal simulation experiment only. Based on this basic framework, we create a local sensitivity analysis technique for the clinical trial enrollment simulation at SAS Institute and demonstrate it on a realistic case for the U.S.

TABLE OF CONTENTS

List of Figures

List of Tables

Chapter 1

Introduction

"Analysis methodology" (AM) is the stochastic simulation research area that addresses all of the statistical aspects of the simulation experiment that remain given a valid simulation model (Nelson, 2016). The key AM problems originally identified in Conway (1963) are: (i) establishing equilibrium (determining the starting and stopping condition of the simulation model to obtain a steady-state performance measure), and (ii) consideration of variability of simulation output and sample size. These issues have been the central topics of research in simulation AM ever since.

However, new tactical problems arise with the dramatic change in the applications of interest to simulation users and in the state of simulation and computing, based on which Nelson (2016) identifies key AM problems for the next 10 years. The AM problems can be categorized into three main aspects: (i) simulation analytics, (ii) parallel simulation, and (iii) simulation to support decisions. Simulation analytics is driven by the dramatically increased data storage, which makes it possible to retain the entire simulation sample path. Simulation analytics treats traditional simulation as data analytics and aims to add the capability of system prediction and revealing key drivers of the system behavior through investigating the retained sample path. The AM problems in parallel simulation address how to deploy multiple processors to assist different tasks of stochastic simulation, e.g., checking feasibility, optimization, etc. In category (iii), the AM problems target

supporting decision making. Simulation studies are virtually all motivated by decision making. Because simulation output is stochastic and input models are never perfect, all decisions should include an accounting for risks. One central topic of interest in this aspect is reduction in and evaluation of model risk.

Model risk refers to the exposure due to an imperfect simulation model. At a high level, a stochastic simulation consists of two parts, inputs and logic, and both affect the validity of a simulation model. The form of risk that is caused by estimating input models from real-world data is referred as input uncertainty. An input model is a fully specified stochastic model that drives the simulation and the process of choosing such models is referred as input modeling.

A simulation model maps its inputs into outputs via a collection of rules and algorithms that mimic the features of the target system. The output, which can be regarded as a function of the inputs, is also affected by the uncertainty in the inputs. Therefore, input uncertainty involves both the level of uncertainty about the input itself, and also how the uncertainty in the input propagates to the output. In other words, the sensitivity of the output to the input. This dissertation addresses both issues: (i) reducing input model risk via better input modeling, and (ii) measuring local sensitivity of the output with respect to the input models. A brief summary of these two research contributions is presented below.

1.1 Reducing Input Model Risk

We address situations when the input models are fit to a relevant sample of real-world data. Despite many advances in input modeling, basic univariate input modeling in practice has not advanced much in decades. The standard practice is fitting a collection of plausible distributions via methods such as maximum likelihood estimation (MLE) and selecting a single best-fit distribution from a candidate set based on some ranking (e.g., goodness-of-fit test) or graphical assessment. Yet under almost all circumstances the real-world data should not be expected to follow any of the candidate

distributions exactly.

Rather than the standard practice described above, frequentist model averaging (FMA) forms a mixture distribution that is a weighted average of the candidate distributions with the weights tuned by cross-validation. Nelson et al. (2020) showed theoretically and empirically that FMA in probability space leads to higher fidelity input distributions. In Chapter 2 we show that FMA can also be implemented in the quantile space and leads to fits that emphasize tail behavior. The quantile model averaging estimator (QMAE), which averages the quantile functions of candidate distributions rather than the cumulative distribution functions, has weight that minimizes the cross-validation error between the fitted and empirical quantile functions. The minimization of cross-validation error can be formulated as a quadratic program with provable unique optimal solution. We also describe an R package named `FMADist` for FMA that is easy to use and available for downloading from CRAN.

1.2 Sensitivity Analysis

Sensitivity analysis quantifies how a model output responds to variations in its inputs, which are critical to better understand system performance, to quantify risk, or to indicate where input change or management may be desirable. However, the following sensitivity question has never been rigorously answered: How sensitive is the mean or variance of a stochastic simulation output to the mean or variance of a stochastic input distribution? This question does not have a simple answer because there is often more than one way of changing the mean or variance of an input distribution as a function of the parameters, which leads to correspondingly different impacts on the simulation outputs.

In Chapter 3 we propose a new family of output-property-with-respect-to-input-property sensitivity measures for stochastic simulation. We focus on four useful cases of this general family: sensitivity of output mean or variance with respect to input-distribution mean or variance. Based

on problem-specific characteristics of the simulation we identify appropriate point and error estimators for these sensitivities that require no additional simulation effort beyond the nominal experiment. We also include two representative examples to illustrate the family, estimators and interpretation of results.

In Chapter 4 we create a local sensitivity analysis technique for the SAS Clinical Trial Enrollment Simulator (CTrES). Clinical trial enrollment is expensive and important, and subject to many uncertainties. Simulation overcomes these limits, so SAS Institute has created CTrES as a strategic decision-support tool specifically for clinical trial enrollment planning. The goal of sensitivity analysis fits into the framework proposed in Chapter 3, but the framework is not sufficient. The new challenges include distributions whose support depends on the its parameters, sensitivity to binary outcomes with respect to the probability of success, interacting inputs, and dependence among inputs due to the simulation stopping condition. We extend the framework to address these new challenges and demonstrate it on a realistic enrollment planning problem for the U.S.

Chapter 2

Better Input Modeling via Model Averaging

2.1 Introduction

Stochastic simulation is a method for analyzing the performance of a system that includes interactions among stochastic processes. At a high level, a stochastic simulation consists of two parts: inputs and logic. The inputs are the uncertain components of a system, while the logic is a collection of rules and algorithms that govern the behavior of the system as a function of the inputs Nelson (2013). Inputs are typically described by fully specified probability models, which includes the case of resampling a fixed set of data. *Input modeling*, as the name suggests, is the process of choosing simulation input models to approximate the uncertainty in the target system. In this chapter we are interested in situations when the input models are "fit" to a relevant sample of real-world data. For instance, we later model data on lot sizes of surface mount capacitors in a manufacturing simulation from Wagner and Wilson (1996).

Despite many advances in input modeling for complex situations—including non-stationary arrival processes, time-series inputs and heterogeneous random-vector inputs—basic univariate input modeling in practice has not advanced much in decades: fit a collection of plausible distributions via methods such as maximum likelihood estimation (MLE) and select one of the candidates based

on some ranking (e.g., goodness-of-fit test) or graphical assessment, perhaps giving extra consideration to a candidate suggested by the real process physics (e.g., sums of more basic random outcomes tend to be normally distributed). Yet under almost all circumstances the real-world data should not be expected to follow *any* of the candidate distributions exactly: mathematical distributions are idealizations that describe some precisely defined process physics (often in a limit), while the real-world processes generating the data have quirks and oddities that make them "real world."

The preceding paragraph might seem to suggest that one should avoid selecting a mathematical distribution altogether, and instead just resample the available real-world data to drive the simulation. This can be a good choice, but empirical distributions are inherently discrete (putting probability mass only on the sampled values), and therefore manifest gaps and lack a (possibly important) tail. Which begs the question, how much data are enough to forego the smoothing and inferred tails obtained by fitting a distribution?

Recently, Nelson et al. (2020) introduced frequentist model averaging (FMA) as an effective way to build input models that better represent the true, unknown input distributions, thereby reducing errors when making inference back to the real world. The premise of FMA is that there *may*, and often will, be one or more parametric probability distributions that fit the real-world data reasonably well, but not perfectly if employed alone. Therefore, FMA averages or mixes the candidate set of distributions to extend their reach, with the ultimate goal of getting the most fidelity from a given candidate set. Cross-validation (CV) is employed to tune the mixture and avoid overfitting. Nelson et al. (2020) provide strong theoretical and experimental evidence that the FMA distribution gets as close as possible to the real-world distribution when using only the component distributions in the given candidate set. By also including the empirical distribution (ED) in the candidate set, FMA provides protection when *none* of the mathematical distributions is adequate, and explicitly indicates how adequate the ED is by the weight assigned to it.

The FMA distribution explored in Nelson et al. (2020) is fit in probability space: it minimizes the cross-validation error between the fitted and empirical cumulative probability distributions.

Therefore, we refer to it as a probability model average estimator (PMAE). In this chapter we introduce FMA in the quantile space, and refer to it as a quantile model average estimator (QMAE). As the name suggests, QMAE minimizes the cross-validation error between the fitted and empirical quantile functions. The choice between PMAE and QMAE depends on the application, and we show it is easy to fit both.

Although an FMA distribution is simple to use in a stochastic simulation once the fitting is complete, the fitting process itself requires solving a numerical optimization problem. The second contribution of this chapter is to provide an R package for fitting and variate generation that can be downloaded from https://cran.r-project.org/web/packages/FMAdist/index.html.

The chapter is organized as follows. We describe our input model averaging method in Section 2.2, and the fitting of the QMAE in Section 2.3 (fitting the PMAE is similar and is described in Nelson et al. (2020)). Documentation of the R package we created follows in Section 2.4. An illustration using the package to model two datasets is found in Section 2.5, followed by conclusions.

2.2 Input Modeling and Input Model Averaging

The most common method for selecting a marginal distribution F to represent an independent and identically distributed (i.i.d.) input process—as described in textbooks (e.g., Law and Kelton (1991)) and employed by distribution fitting software (e.g., BestFit®)—is some variation of the following:

1. Given: $x_1, x_2, ..., x_N$ an i.i.d. sample from some unknown input distribution F^c.

2. Fit $q \geq 1$ candidate parametric families of distributions $\mathscr{F} = \{F_1, F_2, ..., F_q\}$ using methods such as MLE. This yields a set of fitted distributions $\widehat{\mathscr{F}} = \{\widehat{F}_1, \widehat{F}_2, ..., \widehat{F}_q\}$.

3. Rank the fitted distributions using one or more goodness-of-fit measures and evaluate the top

choices. Standard measures are hypothesis-test statistics such as chi-squared, Kolmogorov-Smirnov, Anderson-Darling and Cramér-von Mises, and likelihood-based statistics such as AIC and BIC.

4. Select $\widehat{F} = \text{Best Choice}\{\widehat{F}_1, \widehat{F}_2, \ldots, \widehat{F}_q\}$. Alternatively, use the (possibly smoothed) empirical distribution of $x_1, x_2, ..., x_N$ if nothing fits well.

In contrast to the method above that selects one element of $\widehat{\mathscr{F}}$, the FMA approach of Nelson et al. (2020) creates an "input model average" of the fitted distributions. This is different from finite-mixture models, such as McLachlan and Peel (2004), that assume the true distribution F^c is a mixture of instances of a common parametric family (e.g., normal). Instead, the premise is that there are one or more parametric families of distributions in $\mathscr{F}$ that are plausible choices, perhaps supported by real-world process physics. Therefore, the first two steps above are adopted, but rather than ranking the fitted candidate distributions and selecting the best, the mth fitted distribution is assigned a weight $w_m \geq 0$ with $\sum_{m=1}^{q} w_m = 1$. Thus, the model average distribution is

$$\widehat{F}(x, \mathbf{w}) = \sum_{m=1}^{q} w_m \widehat{F}_m(x), \tag{2.1}$$

where $\widehat{F}_m(x)$ is the mth fitted cumulative distribution function in the candidate set. Notice that some weights may be 0. Virtually any marginal distribution may be in the candidate set, including finite-mixture models and flexible families such as the generalized lambda distribution (Karian and Dudewicz, 2000), as well as the standard choices of normal, lognormal, exponential, gamma, Weibull, etc. *The key to FMA is selecting the weights* $\mathbf{w}$ *to obtain a better fit.*

It is clear from (2.1) that $\widehat{F}(x, \mathbf{w})$ includes each of the individual fitted candidate distributions as a special case of $\mathbf{w}$, while increasing their flexibility by allowing mixtures. Thus, FMA maintains the benefits of the tried-and-true families which, for sound theoretical reasons, are often good approximations, but provides additional degrees of freedom for adjusting to the complexities of real data.

For PMAE, the optimal weight is obtained through minimizing the difference between $\widehat{F}(x,\mathbf{w})$ and $F^c(x)$ in a comprehensive way that guards against overfitting. Specifically, Nelson et al. (2020) solved for $\mathbf{w}$ to minimize the cross-validation squared error with the ED, a consistent and unbiased estimator of $F^c(x)$. They proved that when the true distribution is *not* in the candidate set—which we never expect it to be with real data—then the optimal cross-validation weights converge to the optimal weights as $N \to \infty$. Additionally, Nelson et al. (2020) showed that when the candidate set includes the ED then the PMAE is consistent for F^c in the sense that the weight on the ED $\widehat{w}_{\mathrm{ED}} \xrightarrow{P} 1$ as $N \to \infty$.

Once we have the fitted weights $\widehat{w}_m,\ m = 1,2,\ldots,q$, random-variate generation is easy:

1. Select $M = m$ with probability $\widehat{w}_m,\ m = 1,2,\ldots,q$.
2. Generate $X \sim \widehat{F}_M$.
3. Repeat.

Cross-validation in the probability space is just one possible choice for fitting an FMA. Because probabilities are between 0 and 1, differences between the fitted and ED are also bounded, so large differences (say) in the tails contribute small absolute differences. This suggests we might form FMA distributions with different tail behavior if we do cross-validation in the quantile space. We introduce this new idea here.

Recall that the distributions in the candidate set are $F_m(x)$, $m = 1,2,\ldots,q$, and $\widehat{F}_m(x)$ is the fitted estimator of $F^c(x)$ under the mth candidate distribution. Let $\widehat{G}_m(u)$ be the quantile function corresponding to $\widehat{F}_m(x)$, $G^c(u)$ the quantile function of the true distribution $F^c(x)$, and $\mathbf{v} = (v_1, v_2, \ldots, v_q)^T$ a weight vector belonging to the set $\mathcal{V} = \{\mathbf{v} \in [0,1]^q \colon \sum_{m=1}^q v_m = 1\}$. The quantile model average estimator (QMAE) of $G^c(x)$ is

$$\widehat{G}(u,\mathbf{v}) = \sum_{m=1}^{q} v_m \widehat{G}_m(u),$$

where $0 \leq u \leq 1$. Obviously, the QMAE is defined only for a candidate set of distributions with common support, which will be assumed for all of our tests and analysis.

A good choice of weights $\mathbf{v}$, as in PMAE, will make $\widehat{G}(u,\mathbf{v})$ close to $G^c(u)$ in a comprehensive way. Of course, $G^c(u)$ is unknown. However, based to the fact the empirical cumulative distribution function,

$$\bar{F}(x) = N^{-1}\sum_{i=1}^{N} I\{x_i \leq x\},$$

is consistent for $F_c(x)$, its inverse $\bar{G}(u)$ is also a consistent for $G^c(u)$. By definition, the quantile function of $\bar{F}(x)$ is

$$\bar{G}(u) = \bar{F}^{-1}(u) = \inf\{x\colon \bar{F}(x) \geq u\} = (1-\gamma)x_{(\lfloor uN \rfloor)} + \gamma x_{(\lfloor uN \rfloor + 1)},$$

where $x_{(i)}$ is the i^{th} smallest x of the i.i.d. samples from F^c, and $\gamma = 1$ if $uN - \lfloor uN \rfloor > 0$, and $\gamma = 0$ otherwise. Therefore, $\bar{G}(u)$ is simply an order statistic of the observed values. Similar to the PMAE, cross-validation with $\bar{G}(u)$ is used as we describe explicitly in Section 2.3 below.

Although QMAE is a weighted average of quantile functions, the distribution parameters of these quantile functions are identical to those of the cdfs for PMAE, which are simply MLEs for distributions in the given candidate set $\mathscr{F}$. The only difference between PMAE and QMAE is caused by the CV-estimated weights. Therefore, given a good mixture weight $\widehat{\mathbf{v}}$ that minimizes the CV criterion, variate generation is identical to that of PMAE: each time a value of X is needed, generate integer $M \sim \widehat{v}_m$ to select the distribution, then generate $X \sim \widehat{F}_M$.

Remark: Although less familiar than mixture distributions, there has been previous work on quantile mixture models. Carole and Vanduffel (2015) derive an explicit expression for the quantiles of a mixture of two random variables as a function of the quantiles of the component quantile functions. The validity of the expression is shown through examining all possible cases of discrete and continuous variables with possibly unbounded support. Karvanen (2006) suggests the method of

L-moments or sample Trimmed L-moments (TL-moments) to estimate the parameters of quantile mixtures, where L-moments are linear combinations of order statistics and TL-moments are generalizations of L-moments with increased conceptual sample size. Although this paper proposes certain parametric families of distributions whose parameters can be estimated by the method of L-moments (or TL-moments) with higher reliability than those estimated using conventional moment matching, this method is difficult to apply in many cases because it is impossible to derive closed-form L-estimators for many commonly used distributions. All of this work differs from ours in that we do not assume that the true distribution F^c is a quantile mixture of known families of distributions.

2.3 Fitting

The FMA approach uses CV to provide a good fit without overfitting. The J-fold cross-validation we use is related to the Jackknife model averaging (JMA) of Hansen and Racine (2012), which was originally proposed for improving the quality of estimators in a heteroscedastic linear regression model. The JMA estimator was shown to outperform other estimators in terms of smaller asymptotic expected squared error.

To apply the JMA-like scheme for input modeling in stochastic simulations, we randomly divide the real-world data set $x_1, x_2, \ldots, x_N$ into J groups such that each group has $S = \lfloor N/J \rfloor$ observations. For the jth group, the observations are labeled $x_{(j-1)S+1}, \ldots, x_{jS}$, where $j = 1, 2, \ldots, J$. Let $\tilde{G}_m^{(-j)}(u)$ be the maximum likelihood estimator of $G^c(u)$ for the mth candidate distribution with observations from the jth group removed from the data set. Notice that this is just the inverse function of the MLE for candidate cdf F_m using the same data. Therefore, the QMAE with the jth group removed is

$$\tilde{G}^{(-j)}(u, \mathbf{v}) = \sum_{m=1}^{q} v_m \tilde{G}_m^{(-j)}(u).$$

Denote the ith smallest observation in the jth group as $x_{(i)}^{(j)}$. The ED estimator of $G^c(u)$ using

observations from the jth group only is denoted by $\bar{G}^{(j)}(u)$. Our J-fold CV criterion is

$$\begin{aligned} CV_J(\mathbf{v}) &= \sum_{j=1}^{J}\sum_{i=1}^{S}\left\{\tilde{G}^{(-j)}\left(\frac{i}{S+1},\mathbf{v}\right)-\bar{G}^{(j)}\left(\frac{i}{S+1}\right)\right\}^2 \\ &= \sum_{j=1}^{J}\sum_{i=1}^{S}\left\{\tilde{G}^{(-j)}\left(\frac{i}{S+1},\mathbf{v}\right)-x_{(i)}^{(j)}\right\}^2. \end{aligned}$$

In other words, we consider the sum of squared differences between the QMAE constructed without the jth group of real-world data, and the empirical quantile function constructed from only the jth group, across all groups. The optimal weight vector resulting from this criterion is

$$\widehat{\mathbf{v}} = \text{argmin}_{\mathbf{v}\in\mathscr{V}} CV_J(\mathbf{v}),$$

resulting in the quantile model averaging estimator $\widehat{G}(u,\widehat{\mathbf{v}})$ of $G^c(u)$. This contrasts with PMAE where the weight $\widehat{\mathbf{w}}$ minimizes

$$CV_J(\mathbf{w}) = \sum_{j=1}^{J}\sum_{s=1}^{S}\left\{\widetilde{F}^{(-j)}(x_{(j-1)S+s},\mathbf{w})-\bar{F}^{(j)}(x_{(j-1)S+s})\right\}^2.$$

The optimization problem we need to solve to find $\widehat{\mathbf{v}}$ can be formulated as a quadratic program

(QP). Specifically,

$$\begin{aligned}
\text{Minimize: } CV_J(\mathbf{v}) &= \sum_{j=1}^{J}\sum_{i=1}^{S}\left\{\tilde{G}^{(-j)}\left(\frac{i}{S+1},\mathbf{v}\right)-x_{(i)}^{(j)}\right\}^2 \\
&= \sum_{j=1}^{J}\sum_{i=1}^{S}\left\{\sum_{m=1}^{q} v_m\left[\tilde{G}_m^{(-j)}\left(\frac{i}{S+1}\right)-x_{(i)}^{(j)}\right]\right\}^2 \\
&= \sum_{j=1}^{J}\sum_{i=1}^{S}\left\{\sum_{m=1}^{q} v_m b_{mjs}\right\}^2 \qquad (2.2)\\
&= \sum_{j=1}^{J}\sum_{i=1}^{S}\mathbf{v}^T\mathbf{B}_{js}\mathbf{v} \\
&= \mathbf{v}^T\mathbf{B}\mathbf{v} \\
\text{subject to: } \sum_{m=1}^{q} v_m &= 1 \\
v_m &\geq 0,\ m=1,2,\ldots,q.
\end{aligned}$$

where the matrices $\mathbf{B}_{js}$ and $\mathbf{B}$ are defined in the obvious way. If the $q \times q$ matrix $\mathbf{B}$ is positive-definite, then the objective function is strictly convex and the QP has a unique optimal solution (refer to Chapter 16 in Nocedal and Wright (2006)). Since it is obtained from the quadratic term in (2.2), it is clear that the matrix $\mathbf{B}_{js}$ is positive semi-definite, and therefore so is its sum $\mathbf{B}$; that it is positive-definite with probability 1 can be shown in a similar way to Nelson et al. (2020) for PMAE. PMAE also leads to a QP.

Notice that the dimension of the QP is only the number of candidate distributions, q, and it only needs to be solved once. It is hard to imagine the number of candidates ever being larger than $q = 40$, which is a modest QP. In practice we expect that reasoned choices for the candidate set will lead to $2 \leq q \leq 5$, making it a very small QP. The computational burden is in computing the MLEs for each candidate distribution from all of the data, and from the data with each of the J folds removed (thus, $q(J+1)$ MLEs in total), and construction of the matrix $\mathbf{B}$. Again, these are one-time calculations. The number of observations N and folds J only affect the set up, not the size of the QP.

2.4 R Package

In this section we describe the R package we created for FMA fitting—either PMAE or QMAE—and variate generation from the fitted distribution. The software may be downloaded from https://cran.r-project.org/web/packages/FMAdist/index.html. This section is written in the form of standard R documentation. We illustrate use of the software in the following section.

Description

Creation of an input model via the frequentist model averaging (FMA) approach and random-variate generation for the fitted input model.

Usage

```
fmafit(X,Fset,J,type)
rfma(n,myfit)
```

Arguments

`X`	a numeric vector of nonzero length containing data values for fitting
`Fset`	a list of character strings that specifies the set of candidate distributions; supported distributions are `'normal'`, `'lognormal'`, `'exponential'`, `'gamma'`, `'weibull'`, `'inverse gaussian'`,`'student t'`, `'uniform'`, `'cauchy'`, `'loglogistic'`, `'ED'`, `'beta'`, `'logistic'`, `'pareto'`, `'rayleigh'`
`J`	number of groups to divide the data into for cross-validation; if not specified, `J = 10`
`type`	a character string specifying the type of model averaging estimator, `'P'` for probability, `'Q'` for quantile; if not specified, `type = 'P'`
`n`	number of random variates to generate
`myfit`	a list object returned by `fmafit` containing the four components needed for random-variate generation: `w`, `MLE_list`, `Fset` and `data`

Details

`fmafit` first fits each candidate parametric distribution in `Fset` to the data `X` using maximum likelihood estimation, which yields a set of fitted distributions $\widehat{\mathscr{F}} = \{\widehat{F}_1, \widehat{F}_2, \ldots, \widehat{F}_q\}$. The MLEs for each distribution are returned as `MLE_list`. Next a weight vector $\mathtt{w} = \{w_1, w_2, ..., w_q\}$ is obtained through cross-validation and also returned. The resulting model-average estimator of the true cumulative distribution of the data is

$$\widehat{F}(x, \mathbf{w}) = \sum_{m=1}^{q} w_m \widehat{F}_m(x).$$

The model average fitting can be either in probability space or quantile space. The difference between the two types of model averaging is only in the weight vector associated with the candidate distributions, which is obtained through cross-validation in either probability or quantile space.

`rfma` generates random variates that have the distribution of the model-average estimator. Each time a random variate is needed, a distribution is selected with probability equal to the corresponding weight and then a random variate from the fitted distribution is generated.

Values

`fmafit` returns an object called `myfit` which is a list with four components:

`w`	weight vector associated with distributions in `Fset`
`MLE_list`	list of MLEs for each candidate distribution with `'NA'` for `ED` (empirical distribution)
`Fset`	same as the input argument
`data`	same as input argument `X` (needed for ED)

`rfma` generates random variates from the distribution specified by `myfit`

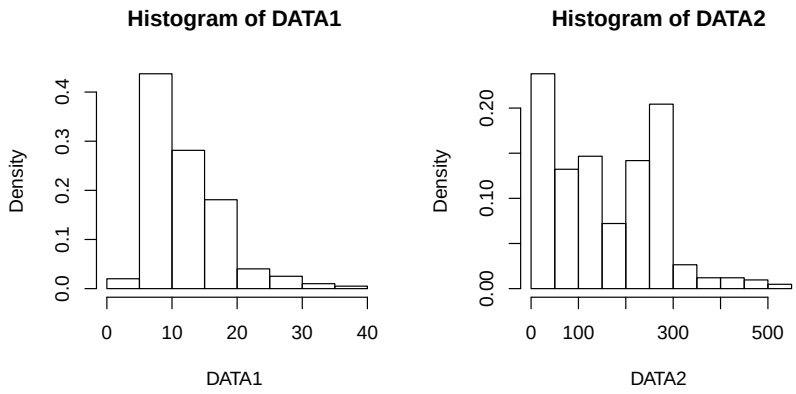

Figure 2.1: Histogram of financial returns (DATA1) and lot sizes (DATA2).

2.5 Illustrations

Nelson et al. (2020) provide a comprehensive empirical evaluation of PMAE by creating cases in which the true distribution F^c is known. Both the fidelity of the fit, and more importantly the fidelity of the simulation *output* with respect to the real-world system, tended to be greatly improved, especially when the tails of the input distributions matter. A queueing example, a highly reliable system and a stochastic activity network were considered, as well as a wide range of input distributions and candidate sets. Similar conclusions hold for QMAE.

In this section we illustrate PMAE, the new QMAE and our software on two real data sets, examining the fits that they provide and the protection afforded by including the ED in the candidate set. One data set is 200 financial returns, which we call DATA1. The second is 417 lot sizes of surface mount capacitors in a manufacturing simulation described in Wagner and Wilson (1996); we call this DATA2. The latter data set is bimodal and therefore is not well represented by the usual unimodal distribution choices. See Figure 2.1 for histograms of the two data sets. We used `J = 10` in both cases.

We first model DATA1 using `fmafit`. Of the distribution choices in `fmafit`, the best-fit distribution as measured by minimum AIC is `'gamma'`. Notice that `fmafit` can be employed to fit a single distribution, if desired, by only having a single choice in `Fset`. In Figures 2.2 and 2.3 we

compare this single choice to a model average of `Fset=('gamma','weibull','lognormal')` for PMAE and QMAE, respectively. We also include the empirical cumulative distribution function (ECDF) in both plots for comparison. The R command for QMAE fitting is

```
myfit <- fmafit(DATA1, c('gamma','weibull','lognormal'), 10, 'Q')
```

The weight vectors obtained for PMAE and QMAE are $\mathbf{w}_P = (0, 0.0804, 0.9196)$ and $\mathbf{w}_Q = (0, 0.3627, 0.6373)$, respectively. From both figures it is clear that PMAE and QMAE are closer to the ECDF than the single best-fit gamma distribution, which indicates that the two FMA estimators better represent the distribution of DATA1. As illustrated in the figures and by the different weights, PMAE and QMAE lead to distinct fits. The Kolmogorov-Smirnov distance between each fit and the ED is 0.08 for the gamma, and 0.06 for both PMAE and QMAE, showing better conformance to the data for model averaging.

Interesting results are observed when fitting the bimodal lot size data, DATA2. The single best-fit distribution based on AIC is `'weibull'`. We compare it with the PMAE and QMAE when `Fset= ('exponential', 'weibull','gamma','lognormal')`. The weight vectors associated with PMAE and QMAE are $\mathbf{w}_P = (0.4343, 0.5657, 0, 0)$ and $\mathbf{w}_Q = (0, 1, 0, 0)$, respectively. Figures 2.4 and 2.5 are plots of the CDFs of the resulting estimators. PMAE is better than the single best-fit distribution in the left tail of Figure 2.4 but worse in the right tail. QMAE, on the other hand, places all weight on `'weibull'`, which is exactly the same as the single-best distribution and better models the right tail. Thus, QMAE emphasizes the (long) right-tail behavior of the unknown input distribution more than PMAE does.

That said, neither fit is very good for this bimodal data when `Fset` includes only unimodal distributions. This is the common context when no standard distribution represents the data, even approximately, and including the ED as a candidate has significant value. To illustrate, we fit DATA2 with two candidate distribution sets, one including the ED (`Fset+ED`) and one without (`Fset`). The weight vector for the two PMAE fits are $\mathbf{w}_P = (0.4343, 0.5657, 0, 0)$ and

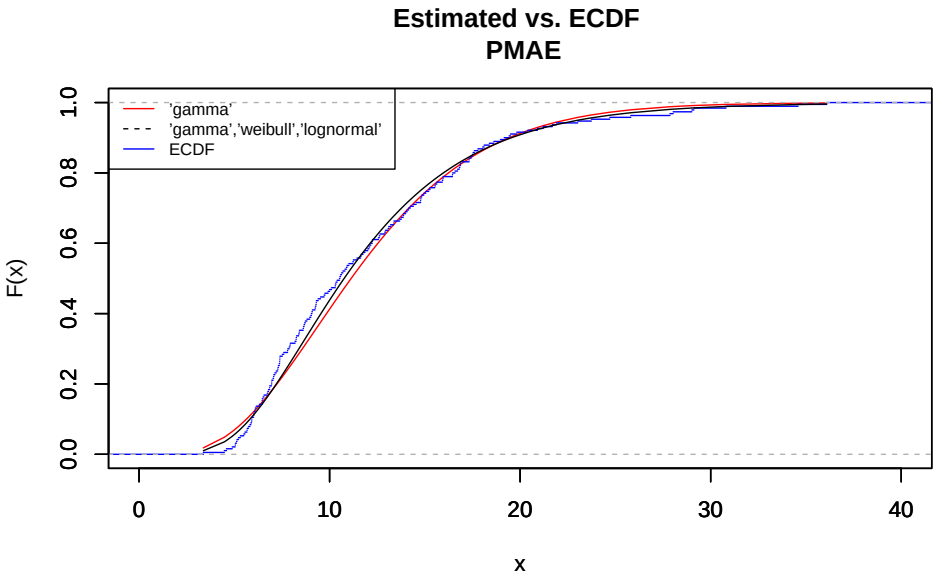

Figure 2.2: CDF of single best-fit distribution `'gamma'`, vs. PMAE with `Fset = ('gamma','weibull','lognormal')` for DATA1.

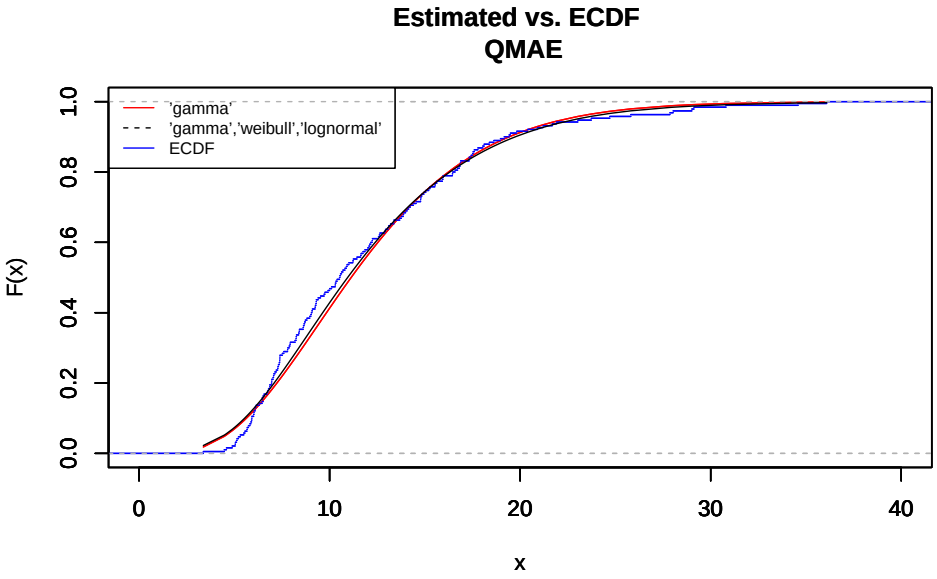

Figure 2.3: CDF of single best-fit distribution `'gamma'` vs. QMAE with `Fset = ('gamma','weibull','lognormal')` for DATA1.

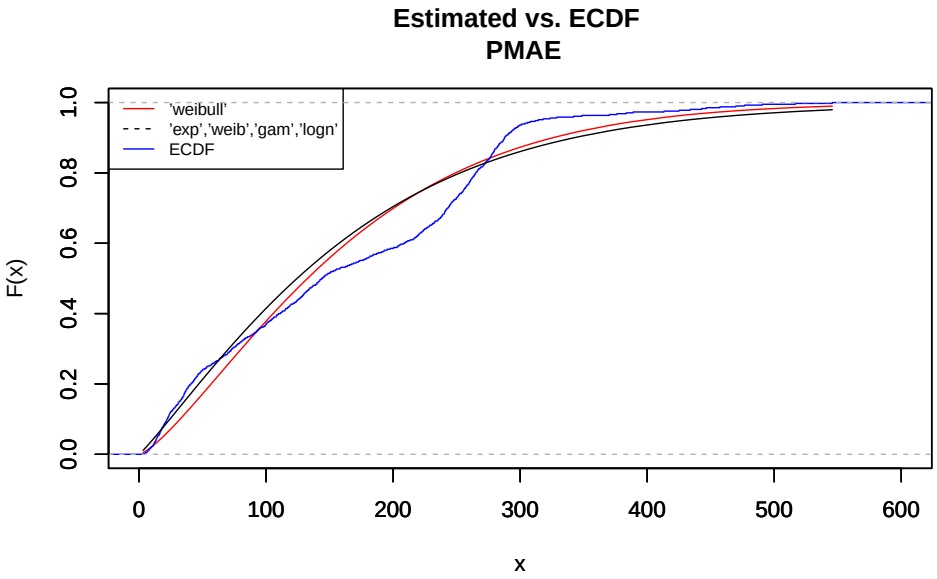

Figure 2.4: CDF of single best-fit distribution `'weibull'` vs. PMAE with `Fset = ('exponential','weibull','gamma','lognormal')` for DATA2.

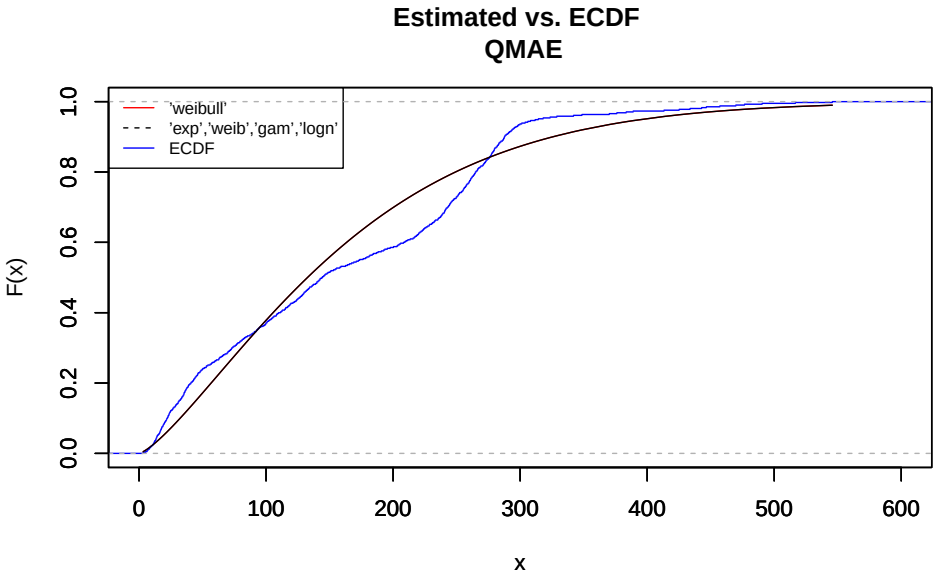

Figure 2.5: CDF of single best-fit distribution `'weibull'` vs. QMAE with `Fset = ('exponential','weibull','gamma','lognormal')` for DATA2.

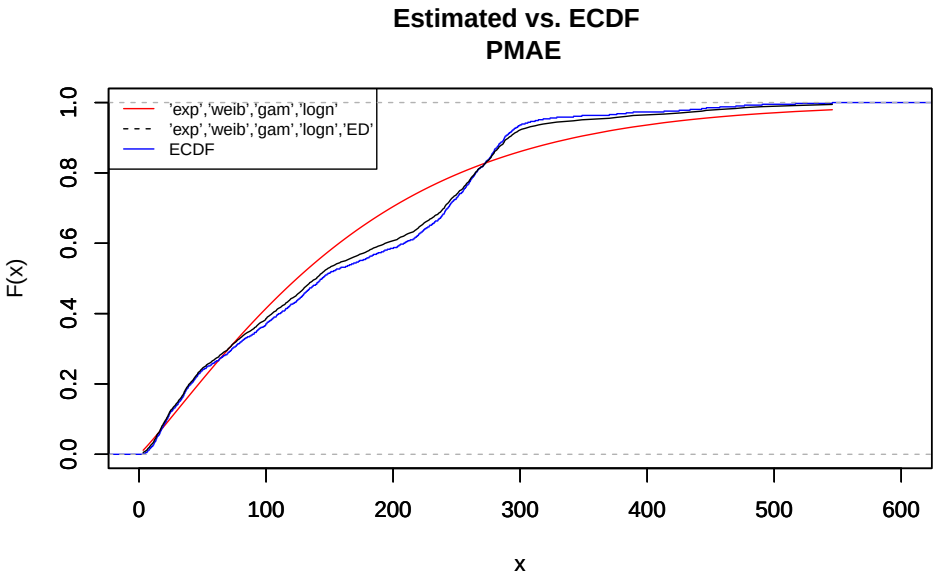

Figure 2.6: CDF of PMAE with `Fset = ('exponential','weibull','gamma','lognormal')` vs. PMAE with `Fset+ED` for DATA2.

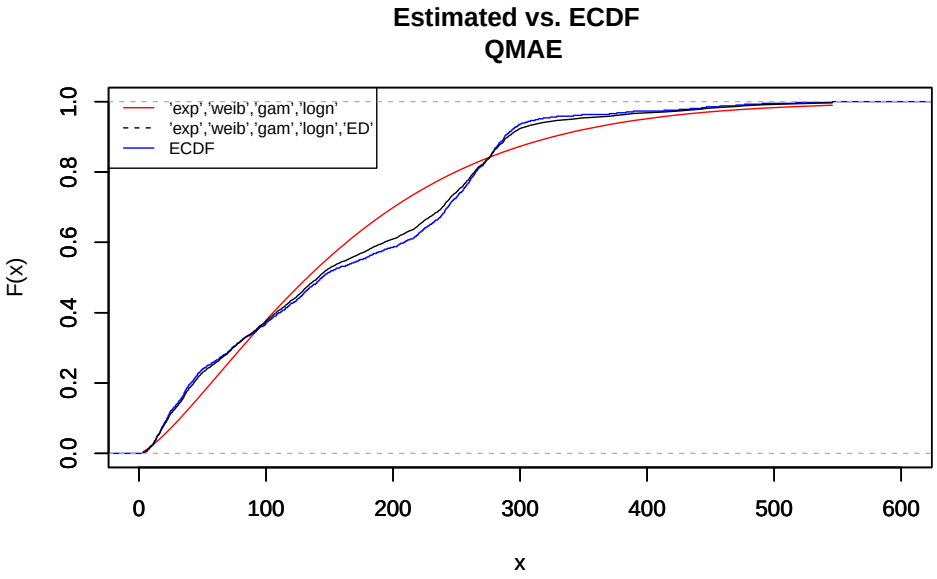

Figure 2.7: CDF of QMAE with `Fset = ('exponential','weibull','gamma','lognormal')` vs. QMAE with `Fset+ED` for DATA2.

$\mathbf{w}_P = (0.1596, 0, 0, 0, 0.8404)$, respectively. The corresponding weight vectors for the two QMAE fits are $\mathbf{w}_Q = (0, 1, 0, 0)$ and $\mathbf{w}_Q = (0.0180, 0.1583, 0, 0.0167, 0.8070)$. Both PMAE and QMAE place most, but not all, of the weight on the ED for `Fset+ED`. Thus, there is still some benefit of including the standard distributions. Figures 2.6 and 2.7 demonstrate the superior fit of `Fset+ED`, and thus the protection offered by including the ED in the candidate set of distributions. The Kolmogorov-Smirnov distance between each fit and the ED is 0.12 for the `'weibull'`, and both the PMAE and QMAE with `'exponential'`, `'weibull'`, `'gamma'` and `'lognormal'`; however, it is only 0.02 for both PMAE and QMAE when the `'ED'` is included, showing a substantially better fit.

2.6 Conclusions

In this chapter we described two methods for "frequentist model averaging" that allow a simulation modeler to exploit the proven value of the standard families of distributions included in every simulation language (normal, lognormal, exponential, gamma, Weibull, etc.), while acknowledging that real-world input data will never perfectly conform to such distributions. Through model averaging we greatly extend the reach of these distributions, and by tuning the model average via cross-validation with the empirical distribution we insure that the fit is representative of the given real-world data. Including the empirical distribution as one of the candidates provides protection against data sets for which none of the standard distributions fit well.

Our R software `fmafit` makes fitting a model average distribution easy and fast. We recommend doing both probability and quantile fitting and comparing the results. The user may then take the weights and parameter estimates returned by `fmafit` and implement them as a simple mixture distribution in any simulation software, or use `rfma` to generate observations outside of the simulation model to read in as needed.

We recommend keeping the candidate set, `Fset`, small, remembering that the weights are *esti-*

mates that will be noisier the more candidate distributions q there are. A set of $q \leq 5$ distributions including any with the right physical basis for the situation (e.g., Weibull for failures), that have good fit measures (e.g., AIC), plus the ED is our suggested approach. Our method also provides a way to judge when it is acceptable to use the ED alone: when the weight on the ED is close to 1. On the other hand, when this weight is far from 1, it indicates that the ED alone is insufficient. In any event, mixing smooths the ED in a way that is less arbitrary than, say, linear interpolation, and extends the ED's tails, which is often desirable.

Chapter 3

Meaningful Sensitivities:A New Family of Simulation Sensitivity Measures

3.1 Introduction

Computer models, including stochastic simulations, can be regarded as functions mapping inputs, denoted generically by $\mathbf{X} = (X^{(1)}, X^{(2)}, \ldots, X^{(K)})^\top$, into one or more outputs, denoted generically by Y, via a collection of rules and algorithms that mimic the features of the system under investigation: $Y = g(\mathbf{X})$. In this chapter we consider "inputs" that are more-or-less beyond the control of the modeler, rather than decision variables that can be chosen. These inputs may be characterized via subjective judgement or estimated from historical data, but in either case are subject to uncertainty that propagates (whether measured or not) through the model to the outputs. Sensitivity analysis (SA) investigates how the output of a computer model responds to variations in its inputs. SA is of critical importance for identifying the relative contributions of the inputs to output uncertainty, assessing model risk, designing robust systems, calibrating a model, and quantifying the interactions among inputs (Saltelli et al., 2000).

Based on the type of uncertainty inherent in the inputs, and the purpose of the model-based

analysis, there are two broad categories of SA: global and local. Global SA is applicable when the input is a random variable naturally varying within its range, $X^{(i)} \sim F_i$, such as the daily temperature or the wind speed at a site during a specific season, and also when the input is believed to be a constant but we have less-than-complete knowledge of its value, such as the failure rate of an electronic component or the stress tolerance of a material. The goal in global SA is to apportion the overall output uncertainty to each of the inputs as a measure their contribution. Management effort may then be applied to the inputs to which the output is the most sensitive, or those inputs that are not significant contributors may be excluded to create a more parsimonious model.

Measures of global SA attempt to discern the inputs that drive the output uncertainty across each input's overall range, while measures of local SA focus on the influence of the inputs near a nominal setting. Local SA makes sense when the input is some parameter or property of a random variable, such as its mean, and we have some confidence in its nominal value. The goal of local SA is to measure the impact on the output of small perturbations of an input around its nominal value. A local sensitivity measure is conceptually (and in our new family, precisely) a partial derivative of the output with respect to the input, which can be justified by a Taylor Series approximation which implies that the first-order partial derivatives are sufficient for predicting output change for small perturbations around the nominal setting. *The "sensitivity" we consider in this chapter is local.*

In the context of stochastic simulation when the simulation is driven by input probability distributions—$X^{(i)} \sim F_i(\cdot|\theta^{(i)})$—then the parameters of each distribution are one type of input, denoted here by $\theta^{(i)}$. Output variability depends on both the input probability distributions themselves (i.e., the inherent randomness of the system), and possibly uncertainty about the parameter values (e.g., if $\theta^{(i)}$ is estimated by $\widehat{\theta}^{(i)}$). When the distributions' parameters are estimated from historical data, then this variability is referred to as "input uncertainty" in the simulation literature; see for instance, Barton et al. (2002), Barton et al. (2014), Lam (2016) and Song et al. (2014). Thus, there is both sensitivity of the performance measures to the nominal values of these input-distribution parameters, and also statistical uncertainty as to their nominal values. *In this chapter*

we focus on the former: the local sensitivity of simulation output properties to input-distribution properties, and not input uncertainty. Thus, our measures are useful even if distribution parameters are obtained from experience, subjective judgement, or guesses, as well as from data.

The reason we emphasize "input-distribution properties" is that sensitivity of the simulation output to the natural input-distribution parameters themselves is often difficult to interpret; this can be true even when the mean or variance of the distribution is one of the parameters. For example, a common sensitivity measure implemented in commercial software (e.g., Simio®) is simply the slope coefficient of a linear regression relating simulation output Y to the sample mean of the input variates. This measure quantifies how much Y would change per unit change in the sample mean of the input random variable, say $X^{(i)}$, but cannot necessarily be interpreted as the partial derivative of $\mathrm{E}(Y)$ with respect to $\mathrm{E}(X^{(i)})$. Of course, the mean and variance are not the natural parameters of many distributions, such as the Weibull which is usually parameterized by shape and scale. Local sensitivities to such parameters are rarely meaningful to the simulation user.

Therefore, in this chapter we reach beyond the partial derivative of the output performance measure with respect to input-distribution parameters, and to the partial derivative of an output *property* with respect to an input *property*. This can be represented conceptually as $\partial H_O(Y)/\partial H_I(X^{(i)})$, where $H(\cdot)$ is an operator yielding a property of a random variable, and the subscript O and I are for the "output" and "input," respectively. Here we consider input distributions that are parametric, having parameters such as mean, variance, shape, scale, rate, etc. Thus, their properties can be represented as functions of their distribution parameters: $H_I(X^{(i)}) = r(\theta^{(i)})$. We focus the properties $\mathrm{E}(\cdot)$ or $\mathrm{Var}(\cdot)$, because of their practical usefulness, but our family is more general. Stated differently, we estimate the sensitivity of the *mean* or *variance* of the simulation output to the *mean* or *variance* of each input distribution around a nominal value of its parameters. To achieve our goal, we propose a new family of local sensitivity measures that enable us to quantify $\partial H_O(Y)/\partial r(\theta_0^{(i)})$ along a *meaningful direction* in the input-parameter space.

A practical example of the sort of insight we seek from local sensitivity analysis is in the clinical

trial enrollment simulation of Jiang et al. (2020). Two output properties of interest are the mean time to enroll 800 patients and the mean cost to enroll them. Users of this model are interested in how sensitive these outputs are to inputs such as the assumed rate of patient arrivals and the mean time to open each clinic. Our sensitivity measurs provide answers to such questions.

Our proposed sensitivity measures require the estimation of a *stochastic gradient* of the output property with respect to the natural input-distribution parameters, denoted $\nabla_{\boldsymbol{\theta}_0^{(i)}} H_O(Y)$. This is a well-studied problem. Existing simulation-based estimators can be categorized into two groups: indirect and direct methods. Indirect methods estimate an approximation of the true gradient by running additional simulations beyond the nominal setting, but they require no knowledge of the underlying mechanics of the simulation model (Fu, 2015). The direct methods, which do require additional knowledge, lead to estimators that are typically unbiased. We also employ the less-well-known method of Wieland and Schmeiser (2006) which is particularly well-suited to estimate output gradients with respect to input-distribution parameters. However, an appropriate stochastic gradient estimator depends on characteristics of the specific problem. Therefore, we describe three methods that apply to distinct situations that we expect to encounter in practice and demonstrate how to use them to obtain point and error estimators of our sensitivity measures.

The chapter is organized as follows. We define our new family of sensitivity measures in Section 3.2, and two representative examples of stochastic simulations to which they apply in Section 3.3. Appropriate gradient-estimation methods for the two examples and the resulting sensitivity estimators are established in Sections 3.4–3.5. Section 3.6 summarizes results from an empirical study of the two examples, followed by conclusions in Section 3.7.

3.2 A New Family of Sensitivity Measures

In this chapter we address the problem of local sensitivity of the mean or variance of the simulation output with respect to the mean or variance of its stochastic inputs. Some of the background

material in this section is based on Section 2 of Jiang et al. (2019).

Consider a simulation model with K independent, scalar, parametric input distributions denoted $F^{(1)}(\cdot|\theta^{(1)}), F^{(2)}(\cdot|\theta^{(2)}), \ldots, F^{(K)}(\cdot|\theta^{(K)})$, having in total $q \geq K$ input parameters (for some distributions θ is a vector). Let $\Theta = (\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(K)})$ be the vector of all input parameters, where $\theta^{(i)} \in \Re^{p_i}$, with $p_i \geq 1$ the dimension of the parameter vector for input distribution i. The simulation output of interest can be represented as $Y(\Theta) = \eta(\Theta) + \varepsilon(\Theta)$ where $\eta(\Theta)$ is the expected value of the simulation output given the input parameters, and $\varepsilon(\Theta)$ is the corresponding stochastic noise with mean 0 and finite variance. In this chapter we consider the parameters Θ to be fixed at Θ_0, so where no confusion is possible we will simply write Y. We also let $X^{(i)}$ represent a random variable with distribution $F^{(i)}$, whose mean μ_i and variance σ_i^2 are differentiable with respect to $\theta^{(i)}$ at the nominal setting $\theta_0^{(i)}$. Our local sensitivity is with respect to each input distribution separately, so for ease of exposition we focus first on a single input $X \sim F(\cdot|\theta)$ with parameter $\theta \in \Re^p$, having mean $\mu = \mu(\theta)$, variance $\sigma^2 = \sigma^2(\theta)$ and nominal parameter value θ_0.

Suppose that we are interested in the effect of a unit change in the variance of an input random variable X on the variance of the output Y, which conceptually is $\partial \mathrm{Var}(Y)/\partial\sigma^2$. This partial derivative is not well defined when there are multiple ways to achieve a change in σ^2. That is, *different* changes in the distribution parameters that lead to the *same* change in the variance of the input might result in a *different* change in the variance of the output. Therefore, the meaning of this derivative is not clear unless the input distribution belongs to the location-scale family $X = \mu + \sigma W$, where $W \sim (0,1)$. Similar issues arise if we want to estimate the impact on the variance of Y of changing the mean of X, or the impact on the mean of Y of changing the mean or variance of X. The key insight is that the mean and variance of both the output and the input are completely determined by θ; therefore, by *fixing* the direction of change in the input-parameter space we obtain a unique value for the desired sensitivities.

Now we are ready to formally introduce our new family of sensitivity measures. Given an output property H_O, an input property H_I, and a normed direction $\vec{\mathbf{d}}$ from the nominal parameter

setting θ_0, we define the sensitivity of $H_O(Y)$ with respect to $H_I(X)$ as

$$\frac{\vec{\mathbf{d}}^T \nabla_{\theta_0} H_O(Y)}{\vec{\mathbf{d}}^T \nabla_{\theta_0} H_I(X)} \tag{3.1}$$

where ∇ is the gradient operator. This is simply an application of the chain rule for directional derivatives. The only requirements are that $\nabla_{\theta_0} H_O(Y)$ exists and can be estimated, and that $\nabla_{\theta_0} H_I(X)$ exists and can be computed. These are mild conditions.

Remark. There are many possible ways to express "sensitivity," therefore, some sensible choices must be made create a well-defined measure. A key choice that we have made is that the *family* of the input distribution does not change as it is perturbed. Given that restriction, our definition is very flexible, as we illustrate later.

For practical reasons we focus on the four sensitivity measures shown in the Table 3.1. For example, we call the sensitivity of the mean of the output, $\mathrm{E}(Y) = \eta(\theta)$, with respect to the mean of the input $\mathrm{E}(X) = \mu(\theta)$, the *mean sensitivity to the mean (MSM)*. In the table, the first letter in bold denotes the property of the output Y of interest and the final letter in italic indicates with respect to what property of the input X. More formally,

$$\mathrm{MSM}_{\vec{\mathbf{d}}} = \frac{\partial \mathrm{E}(Y)}{\partial \mu_{\vec{\mathbf{d}}}} = \frac{\vec{\mathbf{d}}^T \nabla_{\theta_0} \mathrm{E}(Y)}{\vec{\mathbf{d}}^T \nabla_{\theta_0} \mu} \tag{3.2}$$

$$\mathrm{MSV}_{\vec{\mathbf{d}}} = \frac{\partial \mathrm{E}(Y)}{\partial \sigma^2_{\vec{\mathbf{d}}}} = \frac{\vec{\mathbf{d}}^T \nabla_{\theta_0} \mathrm{E}(Y)}{\vec{\mathbf{d}}^T \nabla_{\theta_0} \sigma^2} \tag{3.3}$$

$$\mathrm{VSM}_{\vec{\mathbf{d}}} = \frac{\partial \mathrm{Var}(Y)}{\partial \mu_{\vec{\mathbf{d}}}} = \frac{\vec{\mathbf{d}}^T \nabla_{\theta_0} \mathrm{Var}(Y)}{\vec{\mathbf{d}}^T \nabla_{\theta_0} \mu} \tag{3.4}$$

$$\mathrm{VSV}_{\vec{\mathbf{d}}} = \frac{\partial \mathrm{Var}(Y)}{\partial \sigma^2_{\vec{\mathbf{d}}}} = \frac{\vec{\mathbf{d}}^T \nabla_{\theta^0} \mathrm{Var}(Y)}{\vec{\mathbf{d}}^T \nabla_{\theta^0} \sigma^2}. \tag{3.5}$$

For many input distributions the gradient of the mean or variance of X with respect its parameter θ at θ_0, $\nabla_{\theta_0} \mu$ or $\nabla_{\theta_0} \sigma^2$, is available in closed form or easily computed numerically. The unknowns in (3.2)–(3.5) are $\nabla_{\theta_0} \mathrm{E}(Y)$ and $\nabla_{\theta_0} \mathrm{Var}(Y)$.

Table 3.1: New Local Sensitivity Measures.

	Output **M**ean	Output **V**ariance
Input *M*ean	**M**ean Sensitivity to *M*ean (**MS***M*)	**V**ariance Sensitivity to *M*ean (**VS***M*)
Input *V*ariance	**M**ean Sensitivity to *V*ariance (**MS***V*)	**V**ariance Sensitivity to *V*ariance (**VS***V*)

Estimating $\nabla_{\theta_0}\mathrm{E}(Y)$ has been studied extensively (Fu, 2015). There exist many simulation-based techniques to estimate this gradient and we extend some of the them to estimate $\nabla_{\theta_0}\mathrm{Var}(Y)$. Although gradient estimation is not our contribution, we do present gradient estimators that fit our needs in Section 3.4, based on different practical situations described in Section 3.3. Then in Section 3.5, we provide point and error estimators of the proposed sensitivity measures. Although we focus on the sensitivity of the mean and variance, other properties such as quantiles also fit into this framework.

3.2.1 Meaningful Directions

The proposed sensitivity measures can be computed along any direction $\vec{\mathbf{d}}$, but our definition will only be valuable if there are practically useful directions. For instance, for sensitivity with respect to the variance of the input, Jiang et al. (2019) introduced two meaningful directions. The *steepest-ascent direction* is the direction along which σ^2 increases the fastest: $\vec{\mathbf{d}} = \nabla_{\theta^0}\sigma^2/\|\nabla_{\theta^0}\sigma^2\|$; it is a pessimistic choice. The *minimum-mean-change direction* minimimizes the rate of change in the mean of the input while increasing its variance:

$$\begin{aligned} \underset{\vec{\mathbf{d}}\in\Re^p}{\text{Minimize:}} & \left|\vec{\mathbf{d}}^\top\nabla_{\theta^0}\mu(\theta)\right| \\ \text{subject to: } \vec{\mathbf{d}}^\top\nabla_{\theta^0}\sigma^2(\theta) & > 0 \\ \|\vec{\mathbf{d}}\| & = 1. \end{aligned}$$

For many distributions the mean can be held constant.

The meaningful directions described above, and their obvious generalizations, will be relevant and sufficient for many applications. However, there will be situations in which a problem-specific direction arises; see Jiang et al. (2020) for examples in a clinical trial enrollment simulation. In the following subsections we consider issues associated with shifted distributions and alternative parameterizations.

3.2.2 Shifted Distribution

Notice that the minimum-mean change direction of the input variance may not be unique for input distributions with $p > 2$ parameters. Here we address the special case of a three-parameter distribution obtained by shifting the lower bound of a two-parameter distribution.

Consider the shifted gamma distribution as an example, $X' = X + \xi$ where $X \sim \text{gamma}(\alpha, \beta)$, α is the shape parameter, β is the rate parameter, and ξ is the shift parameter (i.e., $\theta = (\alpha, \beta, \xi)$). Notice that ξ does not affect the variance. Thus, the steepest-ascent direction for sensitivity with respect to the variance of X' is

$$\vec{\mathbf{d}} = \frac{\nabla_{\theta_0}\sigma^2}{\|\nabla_{\theta_0}\sigma^2\|} = \left(\frac{\beta}{\sqrt{4\alpha^2+\beta^2}}, -\frac{2\alpha}{\sqrt{4\alpha^2+\beta^2}}, 0 \right) , \tag{3.6}$$

where $\left(\beta/\sqrt{(4\alpha^2+\beta^2)}, -2\alpha/\sqrt{4\alpha^2+\beta^2}\right)$ is the direction that most rapidly increases the variance of the X.

Because ξ can compensate any change in the mean, there are multiple ways to do a min-mean-change direction unless we fix ξ. We argue that fixing ξ is typically the most relevant case in practice because it defines the support of the distribution; if sensitivity with respect to the support is the goal then it should be assessed directly, rather than indirectly through a change in the mean or variance. For a practical example in which changing the support is relevant, see Jiang et al. (2020). With the lower bound ξ fixed, the minimum-mean-change direction for the sensitivity with respect

to the variance of X' is given by

$$\vec{\mathbf{d}} = \left(-\frac{\alpha}{\sqrt{\alpha^2+\beta^2}},\ -\frac{\beta}{\sqrt{\alpha^2+\beta^2}},\ 0 \right),$$

where $\left(-\alpha/\sqrt{\alpha^2+\beta^2},\ -\beta/\sqrt{\alpha^2+\beta^2}\right)$ is the min-mean-change direction for X.

3.2.3 Alternative Parameterizations

Another issue of note is that even for sensitivity measures from the same family along conceptually the same direction, different parametrization of the input distribution might result in a different sensitivity value. Consider again the gamma distribution that has two parameterizations in common use: gamma(α,β) for which $\mu = \alpha/\beta$, $\sigma^2 = \alpha/\beta^2$, and gamma$(k,\theta)$ for which $\mu = k\theta$ and $\sigma^2 = k\theta^2$. Thus, $\alpha = k$ and $\beta = 1/\theta$. The corresponding unit-norm steepest-ascent directions of the variance of the gamma distribution under these two parameterizations are

$$\vec{\mathbf{d}}_1 = \left(\frac{\beta}{\sqrt{4\alpha^2+\beta^2}},\ -\frac{2\alpha}{\sqrt{4\alpha^2+\beta^2}} \right), \text{ and } \vec{\mathbf{d}}_2 = \left(\frac{\theta}{\sqrt{\theta^2+4k^2}},\ \frac{2k}{\sqrt{\theta^2+4k^2}} \right), \tag{3.7}$$

respectively, and the min-mean-change directions are

$$\vec{\mathbf{d}}_1 = \left(-\frac{\alpha}{\sqrt{\alpha^2+\beta^2}},\ -\frac{\beta}{\sqrt{\alpha^2+\beta^2}} \right), \text{ and } \vec{\mathbf{d}}_2 = \left(-\frac{k}{\sqrt{\theta^2+k^2}},\ \frac{\theta}{\sqrt{\theta^2+k^2}} \right), \tag{3.8}$$

respectively.

Does it matter? Suppose that the service-time distribution of an M/G/∞ queue is gamma, and the output performance of interest, Y, is the number of customers in the system in steady state. Let λ be the rate parameter for the interarrival-time distribution. Then $\mathrm{E}(Y) = \lambda\alpha/\beta = \lambda k\theta$. Thus,

along the steepest ascent directions in (3.7), the corresponding $\mathrm{MSV}_{\vec{\mathbf{d}}}$'s are given by

$$\mathrm{MSV}_{\vec{\mathbf{d}}_1} = \frac{\lambda\beta^3 + 2\lambda\alpha^2\beta}{\beta^2 + 4\alpha^2}$$
$$\mathrm{MSV}_{\vec{\mathbf{d}}_2} = \frac{\lambda\theta^2 + 2\lambda k^2}{\theta^3 + 4k^2\theta} = \frac{\lambda\beta + 2\lambda\alpha\beta^3}{1 + 4\alpha^2\beta^2}.$$

Apparently, $\mathrm{MSV}_{\vec{\mathbf{d}}_1} \neq \mathrm{MSV}_{\vec{\mathbf{d}}_2}$, which can be explained by the different rates of change of the output mean and the input variance while increasing β vs. θ. The $\mathrm{MSV}_{\vec{\mathbf{d}}}$ along the two min-mean-change directions in (3.8), on the other hand, are both equal to 0, which makes sense because $\mathrm{E}(Y)$ does not depend on the variance of the service-time distribution, only the mean.

Remark. What should be done in practice? Currently we suggest either following the parameterization that was originally chosen for the input distribution, or taking the worst case among the alternative parameterizations. Within our family the user can pick *any*, or *multiple*, directions $\vec{\mathbf{d}}$ that they find meaningful without affecting our definition, or the point and error estimators presented below.

3.3 Two Examples

In Section 3.2 we defined four families of sensitivity measures and noted that the greatest difficulty to apply them is estimation of $\nabla_{\theta_0}\mathrm{E}(Y)$ and $\nabla_{\theta_0}\mathrm{Var}(Y)$. An appropriate method depends on characteristics of the input and the output because all gradient-estimation methods use observed outputs Y, and possibly observed inputs X, but in different ways. We employ the following two examples to illustrate three distinct contexts. An M/G/1 queue with gamma-distributed service time illustrates the situation when there are within-replication estimators of both the input-distribution parameter and the output property. A stochastic activity network illustrates two further cases: (i) when neither the input parameter nor the output property can be estimated within each replication (so multiple replications are essential), and (ii) when only an estimator of the output property,

but not of the input-distribution parameter, is observed within each replication. Many practical situations are covered by these cases.

3.3.1 M/G/1 Queue

An M/Gamma/1 queue has $K = 2$ input distributions and $q = 3$ parameters: the interarrival time following an exponential distribution with $\theta^{(1)} = \lambda$, and the service time following a gamma distribution with $\theta^{(2)} = (\alpha, \beta)$. To execute the simulation we set the value of these parameters to $\theta_0^{(1)}$ and $\theta_0^{(2)}$, respectively. Among a total of n replications, the jth replication generates m independent and identically distributed (i.i.d.) interarrival times, $X_{ij}^{(1)}, i = 1, 2, \ldots, m$, and m i.i.d. service times, $X_{ij}^{(2)}, i = 1, 2, \ldots, m$, where $m > 1$.

Since multiple input variates are observed within each replication, the input parameter Θ_0 can be estimated, for instance via maximum likelihood. Denote the estimators of the input parameters from within the jth replication as $\widehat{\Theta}_j = \left(\widehat{\theta}_j^{(1)}, \widehat{\theta}_j^{(2)}\right)$. We do this even though Θ_0 is known because one of the gradient estimators exploits it.

Replication j also generates m outputs, $W_{\ell j}, \ell = 1, 2, \ldots, m$. Suppose $W_{\ell j}$ is the waiting time of the ℓth of a total of m customers arriving to the system after a sufficient warm-up period and before the stopping time within the jth replication. Then one key output from the jth replication is $Y_j = \sum_{\ell=1}^{m} W_{\ell j}/m$, an estimator of the steady-state mean waiting time of customers in the system. If the performance measure of interest is the steady-state variance of the waiting time of customers in the system, then the key output is $Y_j = \sum_{\ell=1}^{m} (W_{\ell j} - \bar{W}_j)^2/(m-1)$ where $\bar{W}_j = \sum_{\ell=1}^{m} W_{\ell j}/m$. Thus, in this setting we observe i.i.d. pairs $(Y_j, \widehat{\Theta}_j)$, $j = 1, 2, \ldots, n$.

3.3.2 A Stochastic Activity Network

This example is based on a problem created by Burt and Garman (1971). A small instance of a project planning problem is modeled as a stochastic activity network (SAN). The network is

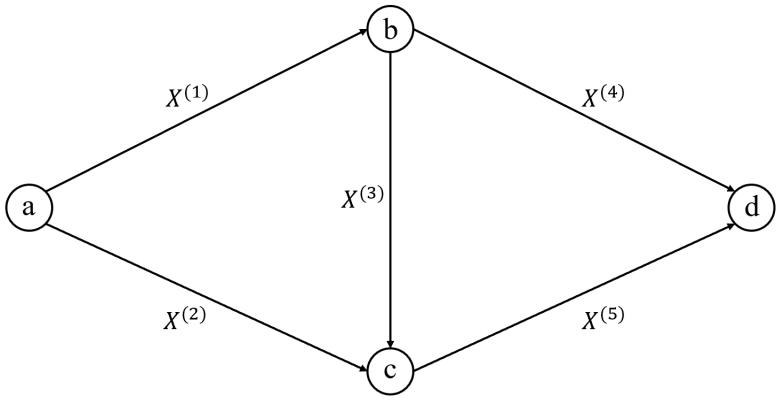

Figure 3.1: A Small Stochastic Activity Network

shown in Figure 3.1 where the nodes (circles) represent project milestones and the arcs (arrows) are activities to be completed. The project starts from the source node a and is completed when the sink node d is reached, with the rule that all outgoing activities from a node begin when all of the incoming activities to that node are completed. The duration of the ith activity is a random variable $X^{(i)}$. Thus, the time to complete the project, Y, will be the longest path through the network: $Y = \max\{X^{(1)} + X^{(4)}, X^{(1)} + X^{(3)} + X^{(5)}, X^{(2)} + X^{(5)}\}$.

In this example there are $K = 5$ inputs whose distributions and parameters are specified in Table 3.6. To execute the simulation we set the values of these parameters to nominal values and run a total of n replications. Notice that for this simulation each replication generates exactly one sample from each input variate and one output value. Let $X_j^{(i)}$ be the sample generated from the distribution of ith activity and Y_j be the output, both from the jth replication. Because of the single input variate from each input distribution within each replication, there is no natural within-replication estimator of $\theta^{(3)}, \theta^{(4)}$ and $\theta^{(5)}$.

If the output property of interest is the mean time to complete the project, then Y_j returned from replication j is the corresponding estimator. However, if the property of interest is the variance of the time to complete the project, then no estimator of this output is observed within each replication. In this case we need a method to obtain the gradient of the variance of Y with respect to input

parameters; we provide such a method in Section 3.4.

3.4 Stochastic Gradient Estimation

In this section we describe three gradient estimation methods that are appropriate for the contexts introduced in Section 3.3. The proper gradient estimator depends very much on specifics of the simulation and there is no one that is superior for all situations. We provide some guidance here to the vast literature on this subject as it relates to our problem; see for instance, Fu (2015) and L'Ecuyer (1990). Although most simulations have multiple input distributions, local sensitivity analysis is with respect to each input distribution separately so we consider only a single input distribution X with a scalar parameter θ having nominal value θ_0 here (e.g., the interarrival time in the M/G/1 queue, or the duration of the first activity in the SAN.)

3.4.1 Finite-Difference Method

A straightforward method to estimate the gradient is the finite-difference (FD) method, which perturbs each component of the input separately while holding the others at their nominal values. To implement FD, we need to make additional replications beyond the nominal experiment for each gradient direction. The simplest FD estimator is the one-sided forward difference estimator given by $\mathrm{FD}(\theta) = (Y(\theta+\Delta\theta) - Y(\theta))/\Delta\theta$ where $Y(\theta)$ is the output of the nominal experiment and $\Delta\theta$ is the perturbation size. In our context when we obtain n replications at the nominal setting, then n additional replications are required in each coordinate direction; therefore, a total of $n(p+1)$ simulation replications are required to estimate the gradient for a p-dimensional input parameter. Averaging $\mathrm{FD}(\theta_0)$ across n replications, the FD gradient estimator is

$$\frac{\widehat{\partial}_{\mathrm{FD}}\mathrm{E}(Y)}{\partial\theta_0} = \frac{1}{n}\sum_{j=1}^{n}\frac{Y_{n+j}(\theta_0+\Delta\theta) - Y_j(\theta_0)}{\Delta\theta}. \tag{3.9}$$

Notice that the FD estimator is biased because the derivative is the limit as $\Delta\theta \to 0$. However, making $\Delta\theta$ too small will result in a noisy estimator; variance reduction technique of common random numbers can be helpful. Thus, there is a trade-off between bias and variance in selecting the perturbation size for each component of the input separately, which can be burdensome for a high-dimensional problem. A more accurate estimator is often obtained by using central differences (Fu, 2015), but this requires about twice the simulation replications required for the one-sided forward-difference estimator.

For the gradient of $\mathrm{Var}(Y)$ with respect to the parameter θ (e.g., in the SAN), we need to obtain an estimator of the $\mathrm{Var}(Y)$, which can be achieved in two ways: batching the replications to estimate the variance, or decomposing $\mathrm{Var}(Y)$ into a function of $\mathrm{E}(Y)$ and $\mathrm{E}(Y^2)$.

Let b be the batch size so there are $k = n/b$ batches of b replications each. For the purpose of presentation we assume that n/b is integer. For the forward difference gradient estimator we do this twice: the first n replications use θ_0, and the second n replications use $\theta_0 + \Delta\theta$. The gradient estimator is

$$\frac{\widehat{\partial}_{\mathrm{FD}}\mathrm{Var}(Y)}{\partial\theta_0} = \frac{1}{k}\sum_{\ell=1}^{k}\frac{S^2_{k+\ell}(\theta_0+\Delta\theta) - S^2_{\ell}(\theta_0)}{\Delta\theta} \tag{3.10}$$

where $S^2_\ell = \sum_{j=(\ell-1)b+1}^{\ell b}\left(Y_j - \bar{Y}_\ell\right)^2/(b-1)$ and $\bar{Y}_\ell = \sum_{j=(\ell-1)b+1}^{\ell b} Y_j/b$.

3.4.2 Likelihood Ratio Method

The likelihood ratio (LR) method, which is also called the score function method, is a direct gradient estimator. It can be computationally efficient because the entire gradient is computed using only simulations at the nominal setting Θ_0 regardless of the dimension. For a p-dimensional input parameter and a simulation budget of n replications, only $2n/(p+1)$ replications are available to estimate each dimension of the gradient using the one-sided FD method, while all n replications are used to compute the gradient in all coordinate directions using the LR method. In addition, LR gradient estimators are easy to derive and exist for most common distributions for all parameters.

LR is particularly well-suited for cases such as the SAN when each replication generates only one input variate from each input distribution.

Suppose that X has a density $f(x|\theta)$. The LR gradient estimator of $\mathrm{E}(Y)$ with respect to θ in the one-dimensional case is

$$\mathrm{LR}(\theta) = Y(\theta)\frac{\partial \ln f(X|\theta)}{\partial \theta}.$$

Since the required partial derivatives for standard distributions are often known in closed-form or easily computed numerically, we do not need additional simulation runs beyond the nominal setting. Thus, to estimate $\partial \mathrm{E}(Y)/\partial\theta$ we average $\mathrm{LR}(\theta_0)$ over n replications and the gradient estimator is

$$\frac{\widehat{\partial}_{\mathrm{LR}}\mathrm{E}(Y)}{\partial \theta_0} = \frac{1}{n}\sum_{j=1}^{n} Y_j(\theta_0)\frac{\partial \ln f(X_j|\theta_0)}{\partial \theta}. \tag{3.11}$$

Notice that the term $\partial \ln f(X|\theta)/\partial\theta$ is the well-known score function in statistics. For instance, when X follows an exponential distribution, the expression in Equation (3.11) can be simplified to

$$\frac{1}{n}\sum_{j=1}^{n} Y_j(\theta_0)\left(\frac{X_j-\theta_0}{\theta_0^2}\right).$$

Although we have not seen it in the literature, rewritting $\mathrm{Var}(Y) = \mathrm{E}(Y^2) - \mathrm{E}^2(Y)$ leads to the asymptotically consistent LR gradient estimator

$$\frac{\widehat{\partial}_{\mathrm{LR}}\mathrm{Var}(Y)}{\partial \theta_0} = \frac{1}{n}\sum_{j=1}^{n}\left\{\left(Y_j(\theta_0)^2 - 2\bar{Y}Y_j(\theta_0)\right)\frac{\partial \ln f\left(X_j|\theta_0\right)}{\partial \theta}\right\} \tag{3.12}$$

where $\bar{Y} = \sum_{j=1}^{n} Y_j(\theta_0)/n$. In this case the LR gradient estimator will be biased for finite n due to the use of $\bar{Y}$ for $\mathrm{E}(Y)$. To achieve an unbiased estimator we could apply the LR concept to the estimator of $\mathrm{Var}(Y)$ based on the sum of squares of all-pairwise differences; although it is unbiased this estimator is computationally burdensome. In Appendix A we derive the LR gradient estimators of $\nabla_{\theta_0}\mathrm{Var}(Y)$ using these two methods.

3.4.3 Wieland-Schmeiser Method

A method due to Wieland and Schmeiser (2006) (WS) is also well-suited to estimate output gradients with respect to input parameters without additional simulation effort beyond the nominal experiment. WS is most appropriate when we observe multiple input variates within each replication so that the input parameter under the nominal setting θ_0 can be estimated, as in the M/G/1 queue example.

Let X be the interarrival-time in the M/G/1 queue and $\widehat{\theta}_j = 1/(\sum_{i=1}^{m} X_{ij}/m)$ be the maximum likelihood estimator (MLE) of θ_0 from replication j. Thus, from n replications we observe i.i.d. pairs $(Y_j, \widehat{\theta}_j),\ j = 1, 2, \ldots, n$. If their joint distribution is bivariate normal then

$$\mathrm{E}(Y|\widehat{\theta}) = \beta_0 + \beta_1 \widehat{\theta} \tag{3.13}$$

where $\beta_1 = \mathrm{Cov}(Y, \widehat{\theta})/\mathrm{Var}(\widehat{\theta})$. Therefore, the $\partial \mathrm{E}(Y)/\partial\theta$ under the nominal setting θ_0 is β_1 and the WS gradient estimator is simply the ordinary least squares (OLS) estimator of β_1:

$$\frac{\widehat{\partial}_{\mathrm{WS}} \mathrm{E}(Y)}{\partial \theta_0} \equiv \widehat{\beta} = \frac{\sum_{j=1}^{n} \left(Y_j - \bar{Y}\right)\left(\widehat{\theta}_j - \bar{\theta}\right)}{\sum_{j=1}^{n} \left(Y_j - \bar{Y}\right)^2}$$

where $\bar{Y} = \sum_{j=1}^{n} Y_j/n$ and $\bar{\theta} = \sum_{j=1}^{n} \widehat{\theta}_j/n$. The WS method regards $\widehat{\theta}_j$ as the realized value of the θ that is fixed at θ_0 and estimates the sensitivity of the response Y_j to this realized parameter $\widehat{\theta}_j$ as it varies across n replications. Because this relationship is linear when they are bivariate normal, the derivative at $\theta = \theta_0$ can be obtained via OLS. When θ is a vector, relationship (3.13) still holds if the joint distribution of $(Y_j, \widehat{\theta}_j)$ is multivariate normal. Lin et al. (2015) show that the corresponding WS gradient estimator can be obtained via multivariate regression.

Notice that joint normality of $(Y_j, \widehat{\theta}_j)$ is only a sufficient condition to apply the method of Wieland and Schmeiser (2006). And it is plausible to approximate the joint distribution as normal when both Y_j and $\widehat{\theta}_j$ are the average of a large number of observations within replication j, or

MLEs of their respective parameters. When this relationship does not hold, batching the replications can be used to induce normality, as suggested in Wieland and Schmeiser (2006).

When estimating the gradient of the variance of the output Y with respect to the input parameter θ, similar to the FD method, we can rewrite $\mathrm{Var}(Y)$ as $\mathrm{E}(Y^2) - \mathrm{E}^2(Y)$ and do multi-response regression. Regressing Y and Y^2 on $\widehat{\theta}$ leads to the gradient estimator

$$\frac{\widehat{\partial}_{\mathrm{WS}}\mathrm{Var}(Y)}{\partial\theta_0} \equiv \widehat{\alpha} - 2\widehat{\mathrm{E}}(Y)\widehat{\beta} = \frac{\sum_{j=1}^{n}\left(Y_j^2 - \overline{Y^2}\right)\left(\widehat{\theta}_j - \bar{\theta}\right)}{\sum_{j=1}^{n}\left(Y_j^2 - \overline{Y^2}\right)^2} - 2\overline{Y}\frac{\sum_{j=1}^{n}\left(Y_j - \bar{Y}\right)\left(\widehat{\theta}_j - \bar{\theta}\right)}{\sum_{j=1}^{n}\left(Y_j - \bar{Y}\right)^2}$$

where $\overline{Y^2} = \sum_{j=1}^{n} Y_j^2/n$.

We could also use batching to set up a single linear regression of the output sample variance on $\widehat{\theta}$ to estimate the gradient. In other words, we compute the sample variance within each batch, and to be consistent, batch the realized parameter $\widehat{\theta}$ with the same batch size to estimate its mean. Then the gradient can be estimated by regressing the sample variance on the batched mean of the realized parameter.

3.5 Sensitivity Measures and Their Variances

In this section we derive the point and variance estimators of the four families of sensitivity measures. From here on θ and $\widehat{\theta}$ are $p \times 1$, denoting the parameter and its estimator of a single input distribution with nominal value θ_0; and Θ and $\widehat{\Theta}$ are $q \times 1$, containing the parameters across all K input distributions with nominal value Θ_0.

For the four families of sensitivity measures introduced in Section 3.2, the corresponding point estimator is obtained by plugging the appropriate gradient estimator into Definitions (3.2)–(3.5),

i.e.,

$$\begin{aligned}
\widehat{\mathrm{MSM}}_{\vec{\mathbf{d}}} &= \vec{\mathbf{d}}^\top \widehat{\nabla}_{\theta_0} \mathrm{E}(Y) \left(\vec{\mathbf{d}}^\top \nabla_{\theta_0} \mu\right)^{-1} \\
\widehat{\mathrm{MSV}}_{\vec{\mathbf{d}}} &= \vec{\mathbf{d}}^\top \widehat{\nabla}_{\theta_0} \mathrm{E}(Y) \left(\vec{\mathbf{d}}^\top \nabla_{\theta_0} \sigma^2\right)^{-1} \\
\widehat{\mathrm{VSM}}_{\vec{\mathbf{d}}} &= \vec{\mathbf{d}}^\top \widehat{\nabla}_{\theta_0} \mathrm{Var}(Y) \left(\vec{\mathbf{d}}^\top \nabla_{\theta_0} \mu\right)^{-1} \\
\widehat{\mathrm{VSV}}_{\vec{\mathbf{d}}} &= \vec{\mathbf{d}}^\top \widehat{\nabla}_{\theta_0} \mathrm{Var}(Y) \left(\vec{\mathbf{d}}^\top \nabla_{\theta_0} \sigma^2\right)^{-1}.
\end{aligned} \tag{3.14}$$

All of these are linear functions of a gradient estimator $\widehat{\nabla}_{\theta_0}$. Thus, if $\widehat{\nabla}_{\theta_0}$ is unbiased, then so is the corresponding sensitivity estimator.

Notice that the only uncertain quantities in these expressions are the gradient estimators; therefore, their variances are

$$\begin{aligned}
\mathrm{Var}\left(\widehat{\mathrm{MSM}}_{\vec{\mathbf{d}}}\right) &= \vec{\mathbf{d}}^\top \mathrm{Var}\left(\widehat{\nabla}_{\theta_0} \mathrm{E}(Y)\right) \vec{\mathbf{d}} \left(\vec{\mathbf{d}}^\top \nabla_{\theta_0} \mu\right)^{-2} \\
\mathrm{Var}\left(\widehat{\mathrm{MSV}}_{\vec{\mathbf{d}}}\right) &= \vec{\mathbf{d}}^\top \mathrm{Var}\left(\widehat{\nabla}_{\theta_0} \mathrm{E}(Y)\right) \vec{\mathbf{d}} \left(\vec{\mathbf{d}}^\top \nabla_{\theta_0} \sigma^2\right)^{-2} \\
\mathrm{Var}\left(\widehat{\mathrm{VSM}}_{\vec{\mathbf{d}}}\right) &= \vec{\mathbf{d}}^\top \mathrm{Var}\left(\widehat{\nabla}_{\theta_0} \mathrm{Var}(Y)\right) \vec{\mathbf{d}} \left(\vec{\mathbf{d}}^\top \nabla_{\theta_0} \mu\right)^{-2} \\
\mathrm{Var}\left(\widehat{\mathrm{VSV}}_{\vec{\mathbf{d}}}\right) &= \vec{\mathbf{d}}^\top \mathrm{Var}\left(\widehat{\nabla}_{\theta_0} \mathrm{Var}(Y)\right) \vec{\mathbf{d}} \left(\vec{\mathbf{d}}^\top \nabla_{\theta_0} \sigma^2\right)^{-2}.
\end{aligned} \tag{3.15}$$

The key to estimating the variance of a sensitivity measure is estimating the variance of the corresponding gradient estimator $\widehat{\nabla}_{\theta_0}$, where the situations we consider can be categorized into the following three settings:

- *Setting 1*: The gradient estimator with respect to the parameters of a single input distribution, $\widehat{\nabla}_{\theta_0}$, is the average of i.i.d. observations of the basic gradient estimator, $\widehat{\nabla}_1, \widehat{\nabla}_2, \ldots, \widehat{\nabla}_n$. Thus, the variance-covariance matrix of the gradient estimator can be estimated by $\widehat{\mathbf{V}} = \widehat{\Sigma}/n$, where

$$\widehat{\Sigma} = \frac{1}{n-1} \sum_{j=1}^{n} (\widehat{\nabla}_j - \overline{\nabla})(\widehat{\nabla}_j - \overline{\nabla})^\top$$

and $\overline{\nabla} = \sum_{j=1}^{n} \widehat{\nabla}_j / n = \widehat{\nabla}_{\theta_0}$.

- *Setting 2*: The gradient estimator across all K distributions, $\widehat{\nabla}_{\Theta_0} = \left(\widehat{\nabla}^{\top}_{\theta_0^{(1)}}, \widehat{\nabla}^{\top}_{\theta_0^{(2)}}, \ldots, \widehat{\nabla}^{\top}_{\theta_0^{(K)}} \right)^{\top}$ is the OLS estimator of the slope coefficient $\widehat{\nabla}_{\Theta_0} = \widehat{\beta}_{1,\text{OLS}}$, where

$$\widehat{\beta}_{\text{OLS}} = (\mathbf{X}^{\top}\mathbf{X})^{-1}\mathbf{X}^{\top}\mathbf{Y} = \begin{bmatrix} \widehat{\beta}_{0,\text{OLS}} \\ \widehat{\beta}_{1,\text{OLS}} \end{bmatrix},$$

with $\mathbf{Y} = [Y_1, Y_2, \ldots, Y_n]^{\top}$ and

$$\mathbf{X} = \begin{bmatrix} 1 & \mathbf{x}_1^{\top} \\ 1 & \mathbf{x}_2^{\top} \\ \vdots & \vdots \\ 1 & \mathbf{x}_n^{\top} \end{bmatrix},$$

with $\mathbf{x}_i$ the predictor variables from the ith replication. Assuming the joint distribution of $(Y, \mathbf{x})$ is multivariate normal, this regression model is correct and the variance-covariance matrix of the slope coefficients is

$$\mathbf{V} = \frac{\sigma_{\varepsilon}^2}{n-q-2} \Sigma_{\mathbf{x},\mathbf{x}}^{-1}, \tag{3.16}$$

where σ_{ε}^2 is the conditional variance of Y given $\mathbf{x}$. Therefore, we can estimate it by

$$\widehat{\mathbf{V}} = \frac{s_{\varepsilon}^2}{n-q-2} \left(\widehat{\Sigma}_{\mathbf{x},\mathbf{x}} \right)^{-1}, \tag{3.17}$$

where $s_{\varepsilon}^2 = \text{SSE}/(n-q-1)$, SSE is the sum of squared errors of the multiple linear regression of Y on $\mathbf{x}$, and $\widehat{\Sigma}_{\mathbf{x},\mathbf{x}}$ is the sample variance-covariance matrix of $\mathbf{x}$. The estimator of the variance-covariance matrix of $\widehat{\nabla}_{\theta_0^{(i)}}$ is the ith $p_i \times p_i$ submatrix on the diagonal of $\widehat{\mathbf{V}}$. The complete derivation of this variance-covariance matrix and its estimator are found in Jiang et al. (2019).

- *Setting 3*: The gradient estimator with respect to the parameters of a single input distribution, $\widehat{\nabla}_{\theta_0}$, can be expressed as the average of $W_j + \widehat{\mu} U_j$, $j = 1, 2, \ldots, n$, where (W_j, U_j) are i.i.d., and $\widehat{\mu} \xrightarrow[n\to\infty]{\text{a.s.}} \mu$. If n is large enough so that we can treat $\widehat{\mu}$ as constant, then *Setting 3* becomes identical to *Setting 1*, with $\widehat{\nabla}_j = W_j + \widehat{\mu} U_j$.

In Appendix C we provide variance estimators for the FD, LR, and WS methods separately by categorizing each situation into one of the three settings above.

3.6 Empirical Illustrations

In this section we illustrate the estimation and interpretation of the proposed sensitivity measures using the two examples introduced in Section 3.3. This is *not* an evaluation or a comprehensive study of the gradient estimators that are inputs to our sensitivity measures. Rather, we demonstrate how these gradient estimators can be combined with our new family of sensitivities to yield useful and interpretable results. If and when better gradient estimators are invented, our sensitivity measures will benefit from them.

Since the true gradients for both examples are not known, but the systems are computationally inexpensive to simulate, we employ intensive simulation to precisely estimate the true gradients for each output property with respect to each input parameter using the FD method; this in turn yields a "true" value of the corresponding sensitivity measures. We compare these to simulations at the nominal setting that employ the LR or WS gradient estimators.

Recall that the proposed sensitivity measures reveal the change in the output mean or variance per unit change in the mean or variance of an input distribution along a meaningful direction. When we refer to "per unit change" for the mean it is in the natural units, while for the variance it in the natural units squared. Stating sensitivities as standard deviation rather than variance is possible, and probably more useful in practice, as illustrated in Jiang et al. (2020).

Table 3.2: Experiment Setup of M/G/1 Queue Example.

Input	Distribution	Parameter	Nominal Value
interarrival time (ARR)	exponential	mean	$\theta_0^{(1)} = 1$
service time (SER)	gamma	(shape, scale)	$\theta_0^{(2)} = (4, 5)$

3.6.1 M/G/1 Queue

The output property of interest is the steady-state mean waiting time of customers in an M/G/1 queue. We illustrate estimating its sensitivity with respect to the mean of each input distribution when the mean changes along the steepest ascent direction, and with respect to the variance of each input distribution when the variance changes along the steepest-ascent and minimum-mean-change directions.

The waiting time is simulated via Lindley's equation and, to speed up the convergence to steady state, the system is preloaded with one waiting time at the steady-state expected value obtained from the Pollaczek-Khinchine formula. The warm-up period is the first 200 customers; after that, the waiting times of 4,000 customers arriving to the system are averaged to estimate the steady-state mean. The nominal experiment ran 900 replications with the input distributions specified in Table 3.2. For simplicity of notation, we use "ARR" for the interarrival-time input and "SER" for the service-time input. The intensive simulation to estimate the true sensitivities ran 64,000 replications ensuring the relative error of the gradient estimator to be less than 0.001.

Since multiple variates are observed from both the interarrival-time and the service-time distributions within each replication of the nominal experiment, the WS gradient estimator is particularly appropriate. Furthermore, because the distributions of the MLEs of the distribution parameters, $\widehat{\Theta}$, are asymptotically normal and the mean waiting time is the average of the waiting times of a large number of customers arriving to system within each replication, it is plausible to approximate the distribution of $(Y, \widehat{\Theta})$ as multivariate normal and thus the relationship between $\mathrm{E}(Y|\widehat{\Theta})$ and $\widehat{\Theta}$ as approximately linear. Accordingly, we have the sufficient conditions to apply the WS method—

Table 3.3: Regression Results for M/G/1 Queue Example (* $p < 0.05$; ** $p < 0.01$; *** $p < 0.001$; **** $p < 2e-16$).

Parameter	Coefficient	Significance	StdErr (SE)
ARR_{mean}	-10.632	$***$	(0.398)
$\text{SER}_{\text{shape}}$	3.070	$***$	(0.215)
$\text{SER}_{\text{scale}}$	-2.532	$***$	(0.160)
Intercept	13.026	$***$	(0.510)
Observations	900		
R^2	0.514		
Adjusted R^2	0.512		
Residual Std. Error	0.197 (df = 896)		
F Statistic	316^{****} (df = 3; 896)		

linear regression of Y on $\widehat{\Theta}$—to obtain the gradient estimator, $\widehat{\nabla}_{\Theta_0}\text{E}(Y)$, and its variance-covariance matrix.

A summary of the fitted model is shown in Table 3.3, where we see that although the adjusted R^2 of 0.51 is low, all predictors are significant. We also applied model diagnostics to validate assumptions including normality, homoscedasticity, and linearity. In summary, we conclude that the linear model fits the data well. Thus, we can draw important conclusions about how changes in the input distribution parameters affect the mean waiting time from the fitted model. For example, the coefficient associated with the mean of the interarrival time is negative, which makes sense because longer interarrival times will help mitigate the congestion and reduce the expected waiting time. A similar explanation applies to the negative (positive) sign of the shape (scale) parameter of the service-time input distribution because increasing (decreasing) shape (scale) increases the mean of the service time which is the main driver of congestion in the queue.

After plugging the gradient estimates into (3.14) and their variances into (3.15), we report the MSM and MSV estimates and their standard errors along with their "true" values in Tables 3.4 and 3.5. The two subscripts specifying the direction of sensitivity measures are "SA," denoting the steepest-ascent direction, and "MM," denoting the minimum-mean-change direction. Notice for all MSM and MSV estimates, the "true" value is included in the $\pm 2 \times$ SE interval, indicating that the

Table 3.4: MSM Estimates for M/G/1 Queue Example.

$MSM_{Input,Dir}$	Estimate (WS)	SE	"True" Value (FD)
$MSM_{ARR,SA}$	-10.632	0.398	-9.959
$MSM_{SER,SA}$	15.514	1.024	14.658

MSM and MSV are pretty well estimated using the WS method with 900 simulation replications.

In Table 3.4 the $MSM_{SER,SA}$ estimate suggests that the steady-state mean waiting time is expected to increase by about 16 time units per unit increase in the mean of the service time at the fastest rate. The $MSM_{ARR,SA}$ estimate, on the other hand, implies that the steady-state mean waiting time is expected to decrease by around 10.6 time units per unit increase in the mean of the interarrival time. Thus, this table suggests that the steady-state mean waiting time is more sensitive to the mean service time at this nominal setting.

In Table 3.5 the $MSV_{SER,SA}$ estimate implies that the steady-state mean waiting time would increase by around 50 time units when the variance of the service time increases by one unit at the fastest rate, which is about three times the $MSM_{SER,SA}$ estimate. This can be explained by the fact in the SA direction for the variance both the the mean and the variance of the service time increase. The $\Delta\mu$ column in Table 3.5, where $\Delta\mu(\theta^0) = \vec{\mathbf{d}}^T \nabla_{\theta^0}\mu / \vec{\mathbf{d}}^T \nabla_{\theta^0}\sigma^2$, tells us approximately how much the mean of each input distribution, $\mu(\theta)$, would change if the variance of the distribution $\sigma^2(\theta)$ changes one unit. Notice that the $MSV_{SER,MM}$ estimate indicates that per unit increase in the variance of the service time in the minimum-mean-change direction would lead to only 2 time units increase in the mean waiting time. Moreover, the $\pm 2 \times$ SE interval includes 0, implying this sensitivity might not be statistically significant. This substantial difference in $MSV_{SER,SA}$ and $MSV_{SER,MM}$ emphasizes the critical importance of specifying a direction of change to be able to interpret results.

Table 3.5: MSV Estimates for M/G/1 Queue Example.

$\text{MSV}_{\text{Input,Dir}}$	Estimate (WS)	SE	"True" Value (FD)	$\Delta\mu$
$\text{MSV}_{\text{ARR,SA}}$	-5.316	0.199	-4.980	0.5
$\text{MSV}_{\text{SER,SA}}$	49.941	3.237	47.231	3.2
$\text{MSV}_{\text{SER,MM}}$	2.382	1.942	2.676	0

Table 3.6: Experiment Setup of SAN Example.

Input	Distribution	Parameter	Nominal Value
$X^{(1)}$	exponential	mean	$\theta_0^{(1)} = 5$
$X^{(2)}$	exponential	mean	$\theta_0^{(2)} = 15$
$X^{(3)}$	weibull	(shape, scale)	$\theta_0^{(3)} \equiv (\vartheta_1^{(3)}, \vartheta_2^{(3)}) = (5, 11)$
$X^{(4)}$	gamma	(shape, rate)	$\theta_0^{(4)} \equiv (\vartheta_1^{(4)}, \vartheta_2^{(4)}) = (30, 2)$
$X^{(5)}$	gamma	(shape, rate)	$\theta_0^{(5)} \equiv (\vartheta_1^{(5)}, \vartheta_2^{(5)}) = (20, 4)$

3.6.2 Stochastic Activity Network

In this example we measure the sensitivity of two output performance measures of the SAN—the mean and the variance of the time to complete the project—to the mean and variance of each of the five input distributions along meaningful directions. Specifically, for sensitivities with respect to the input mean (i.e., MSM and VSM measures), we consider the steepest ascent direction of the mean of the input, and for sensitivities with respect to the input variance (i.e., MSV and VSV measures), the directions are the steepest-ascent and the minimum-mean-change directions.

The nominal setup of the experiment is specified in Table 3.6. The three paths connecting the source node and the sink node, $X^{(1)} + X^{(4)}$, $X^{(1)} + X^{(3)} + X^{(5)}$, and $X^{(2)} + X^{(5)}$, have balanced means so that each path is approximately equally likely to be the longest path. The two output properties of interest represent two different situations: whether there is, or is not, a within-replication estimator of the property of interest. We illustrate the estimation and interpretation of the corresponding sensitivity measures separately in the next two subsections.

Output: Mean of the Project Completion Time

Despite the simplicity of this problem, gradients with respect to the activity time parameters are notoriously hard to estimate with generic methods; this fact has nothing to do with our sensitivity measures, it is simply a property of this noisy problem.

To guarantee the relative error of the gradient estimator is less than 0.001, we ran 200,000 replications in the intensive simulation. The nominal experiment was also run with 200,000 replications, which is larger than we would expect to in practice but we wanted to have a precise comparison of sensitivity measures obtained using different gradient estimation methods. We also ran nominal experiments with a more reasonable number of replications (10,000) and report those results at the end of this subsection. A gradient estimator tailored specifically for this problem, perhaps employing variance-reduction techniques, would also help our sensitivity measures.

The LR method is a good fit for the case when only one input variate is generated within each replication. The gradient estimator with respect to each input parameter is then an average of 200,000 corresponding LR gradient estimates. The LR gradient estimator of $\mathrm{E}(Y)$ with respect to the mean of an exponential distribution (e.g., $X^{(1)}$, $X^{(2)}$), the shape and scale of a Weibull distribution (e.g, $X^{(3)}$), and the shape and rate of a gamma distribution (e.g., $X^{(4)}$, $X^{(5)}$) are given in Appendix B. The estimated values of the gradients are shown in Table 3.7, along with the "true value" estimated using FD.

We also applied the WS method because there are sufficient replications to batch with a large enough batch size to obtain precise MLEs of each input parameter, and at the same time with enough batches for the subsequent regression. Specifically, we batched the observed input variate from each input distribution with batch size $b = 100$ to estimate the MLEs of each input distribution parameter and, to be consistent, the observed output with the same batch size to estimate its mean. Therefore, for the same reason as stated for the M/G/1 queue example, it is reasonable to approximate the joint distribution of the batch means of Y and the MLEs of all input parameters, $\widehat{\Theta}$, as multivariate normal and we can use the WS method to estimate the gradient, $\widehat{\nabla}_{\Theta_0}\mathrm{E}(Y)$, and its

Table 3.7: LR Gradient Estimates of SAN Example with Output E(Y).

Parameter	LR Gradient Estimate	SE	"True" Value (FD)
$X^{(1)}_{\text{mean}}$	0.728	0.017	0.739
$X^{(2)}_{\text{mean}}$	0.689	0.009	0.702
$X^{(3)}_{\text{shape}}$	-0.047	0.018	-0.033
$X^{(3)}_{\text{scale}}$	0.352	0.031	0.347
$X^{(4)}_{\text{shape}}$	0.138	0.012	0.167
$X^{(4)}_{\text{rate}}$	-2.240	0.184	-2.633
$X^{(5)}_{\text{shape}}$	0.174	0.015	0.175
$X^{(5)}_{\text{rate}}$	-0.909	0.075	-0.882

variance-covariance matrix through regressing $\bar{Y}$ on $\widehat{\Theta}$. The summary of the fitted model is shown in Table 3.8 where the coefficient column is the WS gradient estimates. As can be seen from Table 3.7 and 3.8, both the LR and WS gradient estimates are consistent with the "true" values and their SEs are small.

In Table 3.8 all predictors are significant except the shape parameter of the distribution of $X^{(3)}$, which might be because the rate of change in the mean of $X^{(3)}$ with respect to its shape is the smallest compared with that of the other parameters at the nominal setting. The positive signs associated with the means of $X^{(1)}$ and $X^{(2)}$ are not surprising because increasing the mean should increase the length of the corresponding path and accordingly the probability of being the longest. A similar explanation applies to the signs associated with other predictors. The adjusted R^2 is 0.88. We also did regression diagnostics to test the standard multiple linear regression assumptions and checked multicollinearity and outliers. In summary, we conclude that the linear model fits the data well.

After plugging the gradient estimates into (3.14) and their variances into (3.15), the MSM and MSV estimates using the LR and WS methods, their standard errors, and their true values are reported in Table 3.9 and 3.10. For both MSM and MSV measures estimated using either the LR or

Table 3.8: Regression Results for SAN Example with Output $\mathrm{E}(Y)$ (' ' $p < 1$; '$*$' $p < 0.05$; '$**$' $p < 0.01$; '$***$' $p < 0.001$).

Regression Result of $\bar{Y}$ on $\widehat{\Theta}$

Parameter	Coefficient	Significance	SE
$X^{(1)}_{\text{mean}}$	0.762	$***$	(0.018)
$X^{(2)}_{\text{mean}}$	0.704	$***$	(0.006)
$X^{(3)}_{\text{shape}}$	-0.023		(0.024)
$X^{(3)}_{\text{scale}}$	0.313	$***$	(0.042)
$X^{(4)}_{\text{shape}}$	0.138	$***$	(0.016)
$X^{(4)}_{\text{rate}}$	-2.203	$***$	(0.243)
$X^{(5)}_{\text{shape}}$	0.1780	$***$	(0.020)
$X^{(5)}_{\text{rate}}$	-0.894	$***$	(0.099)
Intercept	9.414	$***$	(0.469)
Observations	2000		
R^2	0.877		
Adjusted R^2	0.876		
Residual Std. Error	0.414 (df = 1991)		
F Statistic	1767^{***} (df = 8; 1991)		

Table 3.9: MSM Estimates of SAN Example.

$\text{MSM}_{\text{Input,Dir}}$	Estimate (LR)	SE	Estimate (WS)	SE	"True" Value (FD)
$\text{MSM}_{X^{(1)},\text{SA}}$	0.728	0.017	0.762	0.018	0.739
$\text{MSM}_{X^{(2)},\text{SA}}$	0.689	0.009	0.704	0.007	0.702
$\text{MSM}_{X^{(3)},\text{SA}}$	0.369	0.032	0.331	0.044	0.366
$\text{MSM}_{X^{(4)},\text{SA}}$	0.299	0.025	0.294	0.032	0.351
$\text{MSM}_{X^{(5)},\text{SA}}$	0.726	0.060	0.715	0.079	0.706

Table 3.10: MSV Estimates of SAN Example.

$\text{MSV}_{\text{Input,Dir}}$	Estimate (LR)	SE	Estimate (WS)	SE	"True" Value (FD)	$\Delta\mu$
$\text{MSV}_{X^{(1)},\text{SA}}$	0.073	0.002	0.076	0.002	0.074	0.1
$\text{MSV}_{X^{(2)},\text{SA}}$	0.023	0.0003	0.023	0.0002	0.023	0.033
$\text{MSV}_{X^{(3)},\text{SA}}$	0.090	0.011	0.072	0.015	0.083	0.133
$\text{MSV}_{X^{(3)},\text{MM}}$	0.045	0.010	0.032	0.013	0.039	0
$\text{MSV}_{X^{(4)},\text{SA}}$	0.299	0.025	0.294	0.032	0.352	1.001
$\text{MSV}_{X^{(4)},\text{MM}}$	0.046	0.006	0.035	0.008	0.036	0
$\text{MSV}_{X^{(5)},\text{SA}}$	1.467	0.121	1.445	0.159	1.425	2.019
$\text{MSV}_{X^{(5)},\text{MM}}$	0.116	0.038	0.013	0.051	0.018	0

WS method, the value of almost every estimate is close to the true value obtained using FD method and the SE is always smaller than the estimate itself by at least one order of magnitude. The only estimate that appears to have relatively large error is the LR $\text{MSV}_{X^{(5)},\text{MM}}$ estimate. This might be because the estimation error of the LR gradient estimate with respect to $X^{(5)}_{\text{rate}}$ is magnified in the minimum-mean-change direction. Since the WS method slightly outperforms the LR method in this setting, we use the corresponding estimates for illustrating the interpretation of MSM and MSV sensitivities.

In Table 3.9 the $\text{MSM}_{X^{(1)},\text{SA}}$ estimate is the largest, indicating that a unit increase in the mean of $X^{(1)}$ would lead to an increase in the mean project completion time by about 0.76 units, which is larger than the case when the mean duration of any other activity increases at the fastest rate.

However, since the differences between $\text{MSM}_{X^{(1)},\text{SA}}$, $\text{MSM}_{X^{(2)},\text{SA}}$, and $\text{MSM}_{X^{(3)},\text{SA}}$ are not significant, the mean duration of all three activities should receive attention when managing the mean projection completion time.

In Table 3.10 the $\text{MSV}_{X^{(1)},\text{SA}}$ implies that the mean project completion time is likely to increase around 0.076 time units, i.e., one-tenth of the $\text{MSM}_{X^{(1)},\text{SA}}$ estimate, per unit increase in the variance of $X^{(1)}$. This can be explained by $\Delta\mu$, which suggests that every unit increase in the variance along the steepest-ascent direction comes with 0.1 unit increase in the mean, and that the mean is more influential on the length of the longest path of the SAN. A similar explanation applies to the difference between the $\text{MSM}_{X^{(i)},\text{SA}}$ and the $\text{MSV}_{X^{(i)},\text{SA}}$ estimates for all of the other activities. Additionally, throughout we see the sensitivity to the variance in the steepest-ascent direction is consistently larger than in the minimum-mean-change direction, and in some cases when the mean is held constant the sensitivity may not be statistically significant, e.g., $\text{MSV}_{X^{(5)},\text{MM}}$. This is because the mean duration of activities is the primary determinant of the longest path, and the steepest-ascent direction of the variance also changes the mean, but the minimum-mean-change direction does not, as shown in the $\Delta\mu$ column. Comparing all the MSM and MSV estimates, the MSV with respect to $X^{(5)}$ along the steepest-ascent direction is the largest, suggesting that the variance of $X^{(5)}$ should receive attention under current setup if we want to control the length of the longest path.

Applying the same estimation process to a the nominal experiment with 10,000 replications, we report the LR gradient estimates and the WS gradient estimates obtained with batch size $b = 20$ in Table 3.11. The resulting MSM and MSV estimates are reported in Tables 3.12 and 3.13. To assist with comparison, we also include the true values of the gradients and the corresponding sensitivity measures in these tables.

Comparing Table 3.7 and 3.8 with Table 3.11, the WS gradient estimate obviously has the advantage because its SE does not suffer as seriously as the LR gradient estimate when the number of replications is smaller, even though most of the LR estimates themselves are still relatively

Table 3.11: Gradient Estimates of SAN Example with Output E(Y) and 10,000 Observations.

Parameter	LR Gradient Estimate	SE	WS Gradient Estimate	SE	"True" Value
$X^{(1)}_{\text{mean}}$	0.737	0.074	0.756	0.037	0.739
$X^{(2)}_{\text{mean}}$	0.669	0.038	0.701	0.012	0.702
$X^{(3)}_{\text{shape}}$	0.027	0.078	-0.042	0.038	-0.033
$X^{(3)}_{\text{scale}}$	0.209	0.136	0.611	0.084	0.347
$X^{(4)}_{\text{shape}}$	0.133	0.055	0.139	0.027	0.167
$X^{(4)}_{\text{rate}}$	-2.055	0.820	-2.101	0.409	-2.634
$X^{(5)}_{\text{shape}}$	0.158	0.066	0.109	0.035	0.175
$X^{(5)}_{\text{rate}}$	-0.775	0.330	-0.494	0.169	-0.882

Table 3.12: MSM Estimates of SAN Example with 10,000 Observations.

$\text{MSM}_{\text{Input,Dir}}$	Estimate (LR)	SE	Estimate (WS)	SE	"True" Value (FD)
$\text{MSM}_{X^{(1)},\text{SA}}$	0.737	0.074	0.756	0.037	0.739
$\text{MSM}_{X^{(2)},\text{SA}}$	0.669	0.038	0.701	0.012	0.702
$\text{MSM}_{X^{(3)},\text{SA}}$	0.227	0.142	0.646	0.089	0.366
$\text{MSM}_{X^{(4)},\text{SA}}$	0.274	0.109	0.280	0.055	0.351
$\text{MSM}_{X^{(5)},\text{SA}}$	0.621	0.264	0.397	0.135	0.706

close to true values. The big increase in the SE also explains the discrepancy in the sign of the LR gradient estimate with respect to $X^{(3)}_{\text{shape}}$.

In Table 3.12, the $\pm 2 \times$ SE interval for each LR MSM estimate includes the true value, but in some cases wrongly includes 0, e.g., $\text{MSM}_{X^{(3)},\text{SA}}$. On the other hand, for the WS MSM estimates, their $\pm 2 \times$ SE interval might fail to include the true value because of larger bias of the estimate itself, e.g., $\text{MSM}_{X^{(3)},\text{SA}}$ and $\text{MSM}_{X^{(5)},\text{SA}}$. Thus, when the number of observation is reasonable but still large, it is hard to tell which method has absolute advantage over the other based on this experiment. Similar observations can be drawn from those MSV estimates in Table 3.13. Notice the wrong signs of the LR and WS $\text{MSV}_{X^{(5)},\text{MM}}$ estimate, which might be because the minimum-mean-change direction of $X^{(5)}$ magnifies the moderate estimation error of the gradient estimate.

Table 3.13: MSV Estimates of SAN Example with 10,000 Observations.

$MSV_{Input,Dir}$	Estimate (LR)	SE	Estimate (WS)	SE	"True" Value (FD)	$\Delta\mu$
$MSV_{X^{(1)},SA}$	0.074	0.007	0.076	0.004	0.074	0.1
$MSV_{X^{(2)},SA}$	0.022	0.001	0.023	0.0004	0.023	0.033
$MSV_{X^{(3)},SA}$	0.031	0.049	0.140	0.025	0.083	0.133
$MSV_{X^{(3)},MM}$	0.001	0.041	0.060	0.020	0.039	0
$MSV_{X^{(4)},SA}$	0.274	0.110	0.281	0.055	0.352	1.001
$MSV_{X^{(4)},MM}$	0.017	0.029	0.005	0.012	0.036	0
$MSV_{X^{(5)},SA}$	1.253	0.533	0.800	0.273	1.425	2.020
$MSV_{X^{(5)},MM}$	-0.039	0.164	-0.169	0.067	0.018	0

Output: Variance of the Project Completion Time

When the output property of interest is the variance of the project completion then we need to batch the replications to estimate the variance when using the FD method. Thus, to ensure the relative error of the gradient estimator is less than 0.001, we ran 600,000 replications of intensive simulation with a batch size of $b = 8{,}000$. Assuming the same simulation budget for precise comparison, the nominal experiment was also run for 600,000 replications. Moreover, to observe performance with a more reasonable number of replications we also ran the nominal experiment with 10,000 replications; the results are displayed at the end of this section.

Both the LR method and the WS method are applied for estimating the gradient of $\mathrm{Var}(Y)$ with respect to each input distribution parameter. The LR gradient estimator is the average of 600,000 observations of the basic LR gradient, which is similar to the one in Appendix B except that Y_j is replaced by $Y_j^2 - 2\bar{Y}Y_j$. For the WS method, we did two regressions to estimate the gradient, $\widehat{\nabla}_{\Theta_0}\mathrm{Var}(Y)$. Specifically, the method of batching is first used with a batch size of 100 to set up two linear regressions and then the batched means of both Y^2 and Y were regressed on $\widehat{\Theta}$ (i.e., *Setting 2*).

The resulting LR gradient estimates, WS gradient estimates, and their true values estimated using FD are shown in Table 3.14. Based on this experiment, the LR gradient estimates have an

Table 3.14: Gradient Estimates of SAN Example with Output Var(Y).

Parameter	LR Gradient Estimate	SE	WS Gradient Estimate	SE	"True" Value
$X_{\text{mean}}^{(1)}$	1.182	0.193	1.204	1.025	1.036
$X_{\text{mean}}^{(2)}$	23.131	0.247	23.331	0.340	23.101
$X_{\text{shape}}^{(3)}$	-0.075	0.273	-0.324	1.322	-0.082
$X_{\text{scale}}^{(3)}$	-2.791	0.460	2.606	2.293	-2.997
$X_{\text{shape}}^{(4)}$	-1.502	0.189	-1.052	0.896	-1.388
$X_{\text{rate}}^{(4)}$	22.885	2.816	17.502	13.318	21.052
$X_{\text{shape}}^{(4)}$	0.895	0.228	0.238	1.105	0.697
$X_{\text{rate}}^{(4)}$	-4.579	1.125	-1.748	5.454	-3.543

obvious advantage over the WS gradient estimates because of the smaller SE; in most cases the SE is smaller by one order of magnitude. In some cases, the WS gradient estimate even has the wrong sign, e.g., with respect to $X_{\text{scale}}^{(3)}$. This is not surprising because its SE is nearly as large as the estimate itself. Our conjecture is that this is because the joint regression model of the WS method is noisier and needs more data to be accurate and precise, but the LR method does not need that.

With the gradient estimates and their variances ready, the resulting VSM and VSV estimates are posted in Table 3.15 and 3.16. Here we continue to see the benefit of using the LR gradient estimator when the output property is the variance because the corresponding sensitivity measure has smaller SE. On the other hand, the larger SE of the WS gradient estimate might lead to the wrong sign of the corresponding sensitivity measure estimate, e.g., the wrong signs of the WS $\text{VSM}_{X^{(3)},\text{SA}}$ and $\text{VSV}_{X^{(3)},\text{SA}}$ estimates due to the wrong sign of the WS gradient estimate with respect to $X_{\text{scale}}^{(3)}$. Also notice that, based on this experiment, VSV is harder to estimate than VSM, especially if using the WS method, in the sense that the SE of the estimate is often larger than the estimate itself. Since neither of the sensitivity estimates using LR or WS is uniformly accurate and precise, we focus on the true value obtained via FD for interpretation.

First to recap, the $\text{VSM}_{X^{(i)},\text{SA}}$ tells us the expected change in the variance of the output per unit

Table 3.15: VSM Estimates of SAN Example.

$\text{MSM}_{\text{Input,Dir}}$	Estimate (LR)	SE	Estimate (WS)	SE	"True" Value (FD)
$\text{VSM}_{X^{(1)},\text{SA}}$	1.182	0.193	1.204	1.025	1.036
$\text{VSM}_{X^{(2)},\text{SA}}$	23.131	0.247	23.331	0.339	23.101
$\text{VSM}_{X^{(3)},\text{SA}}$	-3.063	0.480	2.737	2.399	-4.288
$\text{VSM}_{X^{(4)},\text{SA}}$	-3.051	0.375	-2.333	1.776	-2.807
$\text{VSM}_{X^{(5)},\text{SA}}$	3.660	0.900	1.381	4.363	2.833

change in the mean of $X^{(i)}$ at the fastest rate, while $\text{VSV}_{X^{(i)},\text{SA}}$ tells us the effect per unit change in the variance of $X^{(i)}$ at the fastest rate. Comparing the $\text{VSM}_{X^{(i)},\text{SA}}$ estimate and the $\text{VSV}_{X^{(i)},\text{SA}}$ estimate for $i = 1,2,\ldots,5$, we see throughout that the ratio of two effects is almost equal to the corresponding $\Delta\mu$ value. In addition, the VSV estimates in the minimum-mean-change direction are consistently much smaller than the ones in the steepest-ascent direction. Both observations can be explained by the fact that in the steepest-ascent direction of the variance the mean will change by around $\Delta\mu$ unit per unit change in the variance but in the minimum-mean-change direction the mean is held constant. Thus, our observation suggests that the mean is also more important for determining the variance of the longest path of the SAN, especially the mean of $X^{(2)}$.

Furthermore, although it is expected that an increase in the variance of any input would increase the variance of Y, we see a statistically significant negative value associated with $X^{(4)}$ along the steepest-ascent direction. This might be because the three paths in this SAN example are designed such that the probability of each path being the longest is about the same. However, an increase in the variance of an activity breaks the balance, and thus the path where $X^{(4)}$ is more/less likely to be the longest, which might reduce the variance of Y. Comparing all the VSM and VSV estimates, the mean of $X^{(2)}$ and the variance of $X^{(5)}$ are significantly more important for controlling the variance of the project completion time than the others in the nominal setting and these two are where we would recommend putting most of the management effort.

With 10,000 replications for the nominal setting, the gradient estimates and the corresponding

Table 3.16: VSV Estimates of SAN Example.

$VSV_{Input,Dir}$	Estimate (LR)	SE	Estimate (WS)	SE	"True" Value (FD)	$\Delta\mu$
$VSV_{X^{(1)},SA}$	0.118	0.019	0.120	0.103	0.103	0.1
$VSV_{X^{(2)},SA}$	0.771	0.008	0.778	0.011	0.770	0.033
$VSV_{X^{(3)},SA}$	-0.340	0.167	0.655	0.808	-0.568	0.133
$VSV_{X^{(3)},MM}$	0.073	0.142	0.325	0.687	-0.158	0
$VSV_{X^{(4)},SA}$	-3.055	0.376	-2.336	1.778	-2.810	1.001
$VSV_{X^{(4)},MM}$	-0.095	0.100	-0.458	0.453	-0.064	0
$VSV_{X^{(5)},SA}$	7.396	1.818	2.806	8.813	5.722	2.020
$VSV_{X^{(5)},MM}$	0.342	0.569	1.789	2.769	0.193	0

sensitivity measure estimates are reported in Tables 3.17–3.19. Although both gradient estimates have larger SE as the number of observations decreases, the LR gradient estimator still outperforms the WS gradient estimator. However, the gradients are poorly estimated in any event as the SE is much larger than the estimate itself. Similar statements apply to the VSM and VSV estimates in Tables 3.18 and 3.19.

In summary, if we obtain a large enough number of replications, then both WS and LR can work for this example; at smaller (but still large) sample sizes there are issues, especially when the output is $\mathrm{Var}(Y)$. The WS method is better for estimating the sensitivity of the $\mathrm{E}(Y)$, even with a moderate number of observations, while the LR method has obvious benefits for estimating the sensitivity of the $\mathrm{Var}(Y)$. As noted earlier, gradient estimation is difficult for the SAN, even with FD.

3.7 Conclusions

In this chapter we defined a new family of sensitivity measures for a simulation output property with respect to some input property based on directional derivatives. Unlike gradients with respect to the input-distribution parameters, our sensitivity measures are easy to interpret and allow for the

Table 3.17: Gradient Estimates of SAN Example with Output Var(Y) and 10,000 Observations.

Parameter	LR Gradient Estimate	SE	WS Gradient Estimate	SE	"True" Value
$X_{\text{mean}}^{(1)}$	1.218	1.451	3.835	3.301	1.036
$X_{\text{mean}}^{(2)}$	21.222	1.605	22.985	1.094	23.101
$X_{\text{shape}}^{(3)}$	-1.560	1.843	-1.295	3.390	-0.082
$X_{\text{scale}}^{(3)}$	0.517	3.396	18.326	7.597	-2.997
$X_{\text{shape}}^{(4)}$	-0.740	1.428	-3.230	2.472	-1.388
$X_{\text{rate}}^{(4)}$	8.019	21.296	48.439	36.887	21.052
$X_{\text{shape}}^{(5)}$	0.753	1.610	-1.287	3.108	0.697
$X_{\text{rate}}^{(5)}$	-3.944	7.994	10.805	15.214	-3.543

Table 3.18: VSM Estimates of SAN Example with 10,000 Observations.

$\text{MSM}_{\text{Input,Dir}}$	Estimate (LR)	SE	Estimate (WS)	SE	"True" Value (FD)
$\text{MSM}_{X^{(1)},\text{SA}}$	1.218	1.451	3.835	3.301	1.036
$\text{MSM}_{X^{(2)},\text{SA}}$	21.222	1.605	22.985	1.094	23.101
$\text{MSM}_{X^{(3)},\text{SA}}$	0.322	3.554	19.392	8.022	-4.288
$\text{MSM}_{X^{(4)},\text{SA}}$	-1.071	2.840	-6.459	4.918	2.807
$\text{MSM}_{X^{(5)},\text{SA}}$	3.149	6.395	-8.510	12.178	2.833

Table 3.19: VSV Estimates of SAN Example with 10,000 Observations.

$\text{MSV}_{\text{Input,Dir}}$	Estimate (LR)	SE	Estimate (WS)	SE	"True" Value (FD)	$\Delta\mu$
$\text{MSV}_{X^{(1)},\text{SA}}$	0.122	0.145	0.384	0.330	0.104	0.1
$\text{MSV}_{X^{(2)},\text{SA}}$	0.707	0.054	0.766	0.037	0.770	0.033
$\text{MSV}_{X^{(3)},\text{SA}}$	0.740	1.160	4.207	2.274	-0.568	0.133
$\text{MSV}_{X^{(3)},\text{MM}}$	0.774	0.966	1.817	1.785	-0.158	0
$\text{MSV}_{X^{(4)},\text{SA}}$	-1.071	2.843	-6.466	4.924	-2.810	1.001
$\text{MSV}_{X^{(4)},\text{MM}}$	0.810	0.718	0.003	1.077	-0.064	0
$\text{MSV}_{X^{(5)},\text{SA}}$	6.366	12.917	-17.320	24.590	5.722	2.020
$\text{MSV}_{X^{(5)},\text{MM}}$	0.579	3.824	-13.989	6.027	0.193	0

selection of a direction that is meaningful for the problem at hand.

We focused on output mean or variance with respect to input mean or variance, but the only actual restriction is that the input and output properties must be a differentiable with respect to the input-distribution parameters. Identifying the inputs whose mean or variance has the greatest impact on output performance is often of interest for system design and control (e.g., Schoemig (1999), Hopp and Spearman (2011)). Specific directions that seem useful for many application were identified, and with the use of existing gradient estimation methods, point and error estimators for any member of the family are obtainable with data from the nominal experiment only.

Our *definition* of the family of sensitivity measures does not depend on the gradient estimator used, but the statistical properties of our *estimators* do. We illustrated estimation of sensitivity in different contexts in Section 3.6. Although we considered generic gradient estimation methods, specifically FD, LR and WS, problem-specific approaches may also be employed.

An open issue is that our family of sensitivity measure requires specifying a direction, but alternative parameterizations of an input distribution might lead to different values of the sensitivity measure along conceptually the same direction. Although we suggested adopting whatever parameterization was used in the simulation model, it makes sense to search for a parameterization-free definitions of "direction."

In this chapter we focused only on univariate input distributions. Our framework extends naturally to multivariate input distributions. However, meaningful directions involving, say, correlations are harder to specify.

Chapter 4

Sensitivity Analysis in Clinical Trials Simulation at SAS Institute

4.1 Introduction

The design of any clinical trial includes the development of a plan to enroll a target number of patients while remaining within an available budget. Clinical trial enrollment planning can be a daunting task for clinical research organizations (CROs) and pharmaceutical companies, considering the level of uncertainty under which the planning is done. Given the tight deadlines for creating the enrollment plan and the difficulty in capturing the sources of uncertainty, these plans often ignore the variability in the process and create inaccurate predictions of the total cost and total time for study enrollment. SAS Institute has been partnering with the healthcare industry for 40 years and has developed an analytical tool known as SAS Clinical Trial Enrollment Simulator (CTrES) for CROs and pharmaceutical companies. The objective of this tool is to equip its users with the power to develop high-fidelity plans for enrolling patients in clinical trials. SAS is a founding member organization of the CEO RoundTable on Cancer, which is committed to the health and well-being of employees with the belief that cancer can be prevented, and lives can be

prolonged (Goodnight, 2007). In line with this commitment, SAS recognizes how critical it is for CROs and pharmaceutical companies to have access to strategic decision-support tools to design better patient enrollment plans and accurate cost estimates. SAS offers CTrES as a solution for the healthcare industry.

There are three main sequential events that affect the enrollment timeline of a clinical trial: (i) starting clinical research efforts in a country; (ii) activating the clinical research sites in that country; and (iii) enrolling and tracking patients who arrive at each site. The timing of these events and their successful execution determine the performance of the clinical trial enrollment plan. The typical key performance indicators (KPIs) are the duration of time it takes to enroll a target number of patients in the clinical trial and the total cost of starting up the countries, activating the sites, enrolling patients and tracking the enrolled patients. Of these, the time to enroll patients in a trial is the most important consideration of the enrollment plan. Obtaining accurate predictions of these KPIs is often challenging because the events of country start-up, site activation and patient enrollment and tracking are connected through a sequence of subprocesses, each of which is subject to high level of uncertainty.

Here are some representative subprocesses corresponding to the main events (i)–(iii) enumerated above, which are the reasons why a clinical trial enrollment plan may achieve low patient enrollment or high cost. Under main event (i), after preparing the core regulatory package and completing the regulatory timeline, the pharmaceutical company could be unsuccessful at obtaining regulatory approval in a country while still incurring the country activation costs. After collecting information about a site, waiting for the availability of personnel, and spending the time needed to start up the site in main event (ii), site activation may still fail. Even if a site is successfully activated, it may fail to enroll patients. Moreover, after the arrival of patients, only the successful completion of screening will result in the enrollment of patients in the clinical trial in main event (iii).

Thus, there is a high degree of uncertainty at every step of clinical trial enrollment planning,

from the probability that a single site will succeed to enroll patients to the random arrival of patients to a potential site. In their 2013 impact report, the Tufts Center for the Study of Drug Development noted that as many as 37% of sites missed their enrollment targets and 11% failed to enroll a single patient. This lack of certainty turns enrollment planning into a difficult task. In fact, 80% of clinical trials fail to meet enrollment timelines, and one-third of Phase III clinical trial terminations stem from poor patient enrollment planning (Cognizant, 2015). Often the problem is inaccuracy in gauging the time that it takes to reach target patient enrollments and in estimating the total cost of starting clinical research efforts in new countries, activating clinical research sites, and screening and enrolling patients. In 2018, the Tufts Center for the Study of Drug Development reported 30%–40% of sponsors and CROs indicated dissatisfaction with their site initiation processes and concluded that clinical site initiation remains lengthy and highly inefficient. Failure to reach the target patient enrollment in time could lead to delays in getting medicine to the market and result in significant cost overruns.

The industry practice in clinical trial enrollment design is to make many assumptions about enrollment rates and various components of cost, motivated by experience and learning from feasibility studies (Box, 2018). In a feasibility study, a team contacts potential sites and asks questions about the types of patients that they typically treat in the therapeutic area of interest. The team also gathers answers to the following questions: (a) How long would it take to get your site ready to enroll patients? (b) How many patients would you expect to enroll each month? (c) How much would it cost to get ready for enrollment and how much would it cost to treat the patients according to the protocol? The answers to these questions are used to obtain a rough estimate of how long it would take to enroll a target number of patients.

An example of this rough estimate is provided on the left-hand-side (LHS) of Figure 4.1. In Figure 4.1 the cumulative number of patients enrolled (y-axis) is plotted against time (x-axis) and the implied total cost to enroll, say 800, patients is presented. As implied by the construction of a single path on the LHS, the rough estimate based on the data from a feasibility study lacks

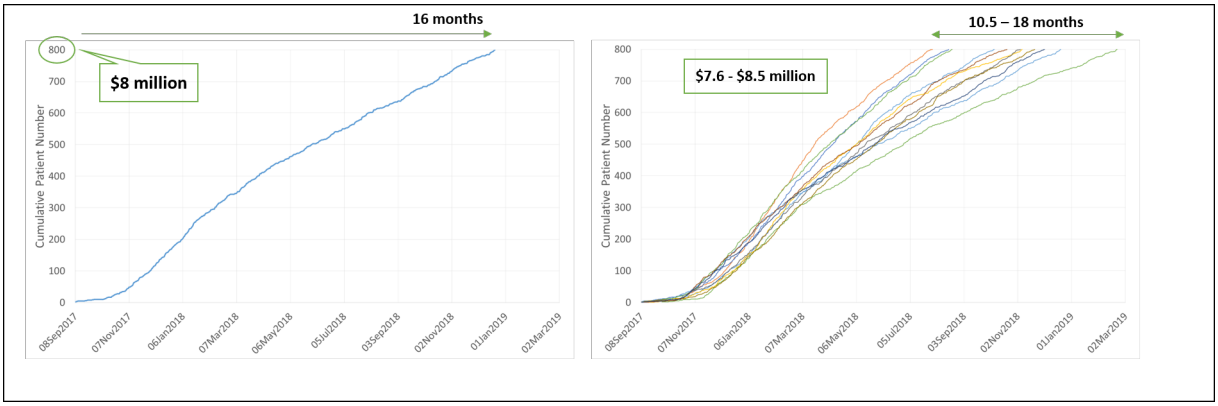

Figure 4.1: Illustrating time vs, patient enrollment in deterministic (LHS) and stochastic (RHS) solutions.

any formal quantification of risk. This is an example of a deterministic but incomplete solution to the problem of KPI prediction in clinical trial enrollment planning. However, accounting for the uncertainty in the inputs provides a range of between 10.5 months and 18 months for the time it takes to enroll 800 patients on the RHS of Figure 4.1, which is generated by a CTrES simulation. Similar statements can be made for the total cost. The two prediction intervals for the total cost and the time it takes to enroll 800 patients clearly demonstrate the significant impact of input risk on KPI variability. The capability to quantify this risk for CROs and pharmaceutical companies has two noteworthy benefits: First, it informs them about the level of risk in their cost and enrollment predictions; second, it guides them towards the identification of enrollment plans to reduce uncertainty. Therefore, it is important to plan patient enrollment and estimate cost by searching beyond traditional deterministic solutions.

Stochastic simulation is a natural choice to capture the risk arising in different stages of a clinical trial enrollment plan. The use of simulation to mimic the clinical trial enrollment process can help overcome the three primary challenges of clinical trial enrollment planning (Handelsman, 2012): 1) The patient enrollment process consists of a long sequence of dynamic random events; 2) the hierarchical relationship among country startups, site activations, and patient screening and enrollment complicates the process of design and analysis of patient enrollment; and 3) enrollment

planning must be driven by country, site, and patient data sets, and the solution must be robust to the data uncertainty and scalable to any number of countries and sites under consideration. SAS CTrES is the solution developed by SAS Institute to overcome these challenges.

In addition to the classical problem of KPI prediction, examples of the what-if questions that planners want to ask are the following: If mean site activation delay increased by 1 week, how would the mean KPI change? If mean screening failure probability increased by 1%, how would the mean KPI change? If the standard deviation of site activation delay increased by one week, how would the mean KPI change? Obtaining answers to these what-if questions helps CROs and pharmaceutical companies diagnose the current setup and decide where to put management effort towards the design of a better clinical trail enrollment plans.

Each of these questions can be answered by creating a new scenario in the SAS CTrES UI (User Interface). Specifically, the first question can be addressed by creating a second scenario where the mean site activation delay is increased by one week, and the simulation output data obtained from these two scenarios are compared. Unfortunately, a typical enrollment planning exercise may involve multiple countries and hundreds of sites. A study of the SAS CTrES simulation engine for a single-country, 10-site setting reveals 51 different stochastic inputs to support enrollment planning (Biller et al., 2019). Thus, at least 52 computationally intensive simulations would be needed just to evaluate the sensitivity to changes in the means for one possible scenario of countries and sites to activate. Thus, CTrES currently lacks the capability to quickly answer what-if questions in a way that scales with the number of countries and sites involved in a clinical trial design. *Our work reported here enables CTrES to overcome this limitation and equips CTrES with the power to answer what-if questions for any number of stochastic inputs using the output data obtained from simulation of the only base scenario.*

Answering the types of what-if questions posed above for the stochastic inputs of the simulation is a type of local sensitivity analysis, which focuses on the influence of the inputs on the output near a nominal setting. And while SAS already has global sensitivity analysis capabilities, it does

not support the type of local sensitivity analysis CTrES requires. *The focus of this chapter is creation of local sensitivity analysis technology for CTrES.* Although the methods presented here were created for CTrES, they are broadly applicable to many simulation contexts.

The chapter is organized as follows. Section 4.2 presents the literature review on existing studies about clinical trial enrollment planning. In Section 4.3 we illustrate the basic elements of CTrES including the process flow and the simulation inputs. The sensitivity measures of interest for the CTrES users and the solutions to new technical challenges are addressed in Section 4.4. Section 4.5 summarizes results from an illustrative one-country-ten-site case, followed by conclusions in Section 4.6.

4.2 Literature Review

Clinical trial enrollment planning has been studied from different perspectives for different purposes. However, most published research makes significant simplifying assumptions to formulate the problem as a mathematical model that is tractable.

From the perspective of production planning and supply chain design, the key is to position the right inventory of drugs at the right time at the right trial site considering both the cost of production, shipping, inventory carrying, enrollment, and duration of the clinical trial, e.g., Zhao et al. (2018, 2019). The problem is formulated as a multi-stage stochastic programming model and the only uncertainty considered is the number of patients, which is modeled as a countable number of scenarios where each scenario represents a possible realization based on previous trial data. Furthermore, the enrollment cost is either not considered or assumed to be independent of patient arrivals, which seems unrealistic in the scenarios modeled by CTrES.

Kouvelis et al. (2017) study the problem of maximizing the expected net present value of a drug considering the costs of clinical trial, the drug's likelihood of approval, and its subsequent expected revenue if approved given the maximum duration of the study. The problem is modeled

as a discrete-time, discounted dynamic program determining when and how many test sites should be opened and the rate at which patients should be recruited to achieve the optimum. To simplify the analysis, the paper assumes that the sites will be opened in a given order, which is restrictive unless all sites have identical capacity and zero startup cost. Moreover, under most cases, the recruitment rate is not controllable but rather a site-specific characteristic with uncertainty.

There are also many studies focusing on modeling of patient recruitment, e.g., Monte Carlo simulation models in Abbas et al. (2007), and the Pareto-Poisson statistical model in Mijoule et al. (2012). The most widely used is the Gamma-Poisson model in the empirical Bayesian framework proposed by Anisimov. This purely statistical model not only enables the prediction of recruitment with confidence bounds, but also evaluates various site performance measures and approximates the minimal number of sites needed with confidence (Anisimov, 2008, 2009, 2016). The model accounts for the natural variation in recruitment over time, in recruitment rates among different sites, and in site startup delays (Anisimov, 2008). However, the real-life clinical trail enrollment process is far more complex because of the uncertainty associated with site startup and enrollment success, and the patient screening success. Mijoule et al. (2012) further studies to what extent estimation error of the arrival rate generates an error in the prediction of the trial duration, which is known as "input uncertainty" in the simulation literature.

In summary, no model in the existing literature fully captures the risk arising in different stages of a clinical trial enrollment plan, let alone the capability of answering the what-if questions that are critical for indicating where input change or management effort may be desirable.

4.3 The Clinical Trial Enrollment Model

SAS considers any stochastic simulation to consist of system logic and simulation inputs. For a clinical trial enrollment simulation, the process flow in Figure 4.2 plays the role of the system logic, and Table 4.1 specifies the simulation inputs. Sampling realizations of these inputs and applying

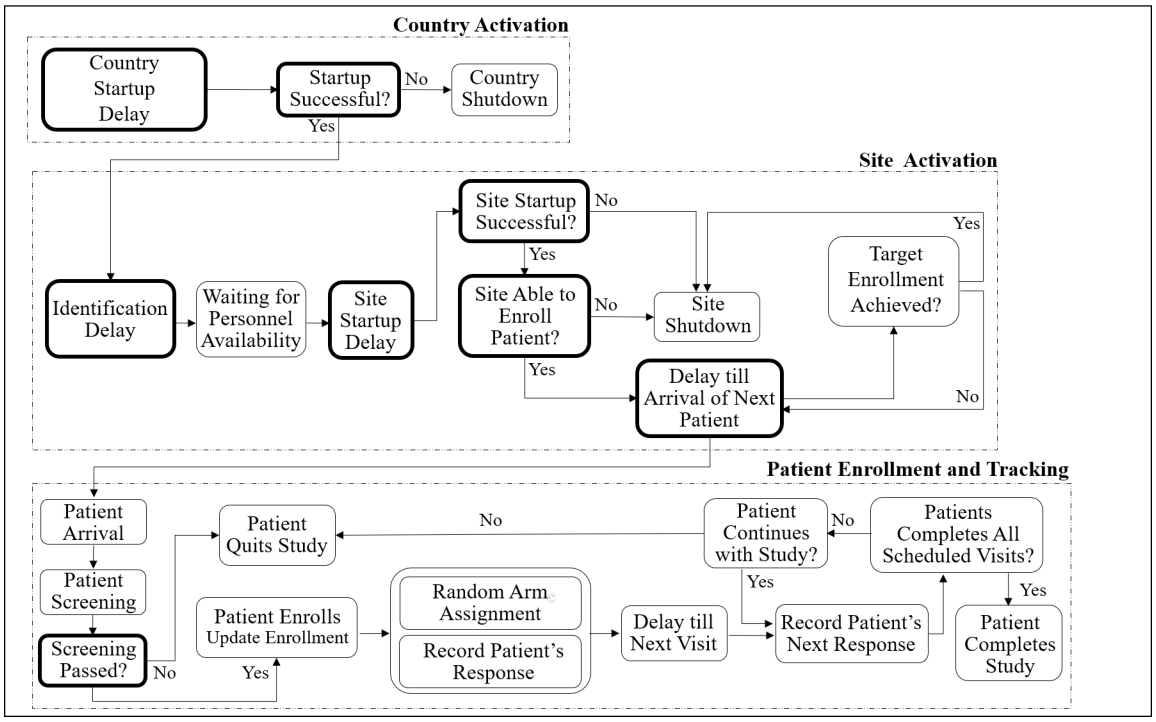

Figure 4.2: High-Level View of Clinical Trial Enrollment Process Flow.

the system logic enables the generation of predictions of KPIs.

Figure 4.2 presents a high-level illustration of the CTrES process flow, which is implemented in SAS Simulation Studio, a Java-based discrete-event simulation tool (Hughes et al., 2018). Thus, SAS Simulation Studio serves as the engine for SAS CTrES to address clinical trial enrollment planning questions for SAS customers; it is made available through a web interface as software as a service.

The simulation model is composed of three modules consistent with the three main events introduced in Section 1: (i) country activation, (ii) site activation, and (iii) patient enrollment and tracking. Each module introduces a specific entity flowing through the corresponding portion of the logic illustrated in Figure 4.2: (i) Country entities in Country Activation module, (ii) Site entities in Site Activation module, and (iii) Patient entities in Patient Enrollment and Tracking module. Each entity has attributes that are subject to uncertainty characterized by probability distributions based on expert opinions and historical data. Within each replication, the realized value of each uncertain

attribute of each entity is updated after the corresponding subprocesses and reported right before leaving the corresponding module.

Box (2018) outlines the following key points for the CTrES process flow: (a) A start time is established as Day 0 for the clinical trial study. (b) Countries are selected and may receive approval after a certain duration of delay. Countries may have different values for the startup delay during country activation. Once a country successfully starts up, site initiation begins. (c) Sites are initiated in the countries that start up successfully and can start enrolling patients. (d) Patients start arriving at sites that are activated successfully and able to enroll for screening. (e) Some of the patients fail the screening process while those passing the screening test are enrolled in the study. (f) Patients progress through the study. Some of the patients quit the study early while others reach the last scheduled visit. (g) As soon as total patient enrollment reaches the target enrollment, the patient arrival process is terminated. (h) The study remains operational until all the patients that are still flowing through the system either finish the study or drop out.

The two primary KPIs of interest for a CTrES user is the time it takes to enroll a given target number of patients, say 800, denoted as "TimeToEnrollTarget," and the implied total cost of the clinical trial, denoted as "TotalCost," which is the sum of country and site activation costs, and the costs of screening and enrolling 800 patients. There are two other timeline KPIs: "FirstTimeEnrolls," which is the time the first patient enrolls, and "EnrollmentDuration," which is the time between "FirstTimeEnrolls" and "TimeToEnrollTarget."

For these KPIs, only the stochastic inputs associated with countries and sites are relevant for the development of the local sensitivity analyzer for SAS CTrES. Table 4.1 lists those uncertain inputs and their corresponding probability distributions, which are the sources of uncertainty in the process flow illustrated in Figure 4.2. The use of the three-parameter triangular distribution to capture the uncertainty associated with the length of subprocesses is common practice so that expert users can provide the corresponding input parameters, i.e., minimum, most likely, and maximum values. The Bernoulli distributions are used to capture the uncertainty associated with a subprocess

Table 4.1: Distributions Relevant for Local Sensitivity Analysis.

Level	**Uncertainty**	**Input Distribution**	**Input Parameters**
Country	Startup_Delay	triangular distribution	(min, mode, max)
	Startup_Success	Bernoulli distribution	probability
	Screening_Failure	Bernoulli distribution	probability
Site	Startup_Delay	triangular distribution	(min, mode, max)
	Startup_Success	Bernoulli distribution	probability
	Enrollment_Success	Bernoulli distribution	probability
	Identification_Delay	triangular distribution	(min, mode, max)
	Site_Patient_Arrival	piecewise-constant non-stationary Poisson process (NSPP) with two pieces	(rate_high, rate_low)
	Duration_of_Rate_High	triangular distribution	(min, mode, max)

happening or not. Notice that although the enrollment of each patient is subject to the probability of passing screening, the screening failure distribution is designed at the country level.

The input that is quite different and worth more explanation is the patient arrival process, designed at the site level. The arrival process of patients is characterized by a piecewise-constant non-stationary Poisson process (NSPP) with two pieces because sites tend to have patients arriving at a higher rate at the beginning of the clinical trial. Moreover, there is uncertainty about the length of the time the arrival rate is high. A triangular distribution is used to capture this uncertainty.

4.4 Sensitivity Measures and New Challenges

The what-if questions described in Section 4.1 can be summarized as the quantification of the expected change in the mean KPI per unit change in the mean or standard deviation of each uncertain subprocess, as characterized by probability distributions specified in Table 4.1. Since the KPI is the simulation output and we are interested in its mean, where no confusion will arise, we will redefine KPI as the expectation of the simulation output from now on, i.e., KPI $\equiv$ E[output]. Therefore, the goal is to measure the sensitivity of each KPI to the mean or standard deviation of each input distribution, near a nominal setting. This goal fits in the framework of output-property-

with-respect-to-input-property sensitivity measures proposed in (Jiang et al., 2019, 2020). The sensitivity measures of interest in the context of CTrES are two special cases of the general family: *mean sensitivity to mean* (MSM) and *mean sensitivity to standard deviation* (MSSD). The MSSD measure is built upon the *mean sensitivity to variance* (MSV) measure described in Jiang et al. (2020) through replacing the variance by the standard deviation.

For ease of explanation we focus on a single output and a single input distribution. Let Y be the simulation output, $\mathrm{E}[Y]$ be one of the KPIs, and $X \sim F(\cdot|\theta)$ be one of the uncertain inputs that are listed in Table 4.1 with distribution parameter θ. Without loss of generality, let $\theta \in \Re^p$ where $p \geq 1$. Further, let $\mu = \mu(\theta)$ and $\sigma = \sigma(\theta)$ be the mean and standard deviation of input X, both of which are differentiable with respect to θ around the nominal setting $\theta = \theta^0$.

Recaping the definition introduced in Jiang et al. (2020), the MSM measure is defined as the directional derivative of $\mathrm{E}(Y)$ with respect to μ along a normed direction $\vec{\mathbf{d}}$ from the nominal parameter setting θ^0, i.e.,

$$\mathrm{MSM}_{\vec{\mathbf{d}}} = \frac{\partial \mathrm{E}(Y)}{\partial \mu_{\vec{\mathbf{d}}}} = \frac{\vec{\mathbf{d}}^T \nabla_{\theta^0} \mathrm{E}(Y)}{\vec{\mathbf{d}}^T \nabla_{\theta^0} \mu}.$$

A meaningful direction is the steepest-ascent direction of the mean, $\vec{\mathbf{d}} = \nabla_{\theta^0}\mu / \|\nabla_{\theta^0}\mu\|$, which is a defensive (aggressive) choice assuming the goal is to identify the maximal sensitivity. Similarly, MSSD is defined as

$$\mathrm{MSSD}_{\vec{\mathbf{d}}} = \frac{\partial \mathrm{E}(Y)}{\partial \sigma_{\vec{\mathbf{d}}}} = \frac{\vec{\mathbf{d}}^T \nabla_{\theta^0} \mathrm{E}(Y)}{\vec{\mathbf{d}}^T \nabla_{\theta^0} \sigma}.$$

For sensitivity with respect to the standard deviation, meaningful directions are the steepest ascent direction along which σ increases the fastest: $\vec{\mathbf{d}} = \nabla_{\theta^0}\sigma / \|\nabla_{\theta^0}\sigma\|$; and the minimum-mean-change direction, which minimizes the rate of change in the mean of the input while increasing its standard deviation. The minimum-mean-change direction can be determined through solving an optimization problem similar to the one in Section 2.1 of Jiang et al. (2020) after replacing σ^2 with σ.

In the context of CTrES, the gradient of the mean or standard deviation of the inputs with respect to the input parameter, $\nabla_{\theta^0}\mu$ or $\nabla_{\theta^0}\sigma$, are known and the key is estimating $\nabla_{\theta_0}\mathrm{E}(Y)$,

known as the stochastic gradient. For stochastic gradient estimation in CTrES we used the method of Wieland and Schmeiser (2006) as extended by Lin et al. (2015).

Specifically, let Y_j be the output and X_{ij}, $i = 1,2,\ldots,m_j$, be the input variates generated within replication j, $j = 1,2,\ldots,n$, where m_j could be random. The input parameter of X under the nominal setting, θ^0, can be estimated (e.g., maximum likelihood estimation, or moment matching) as a function of the input variates observed within each replication. The method of Wieland and Schmeiser regards the estimator of the parameter, $\widehat{\theta}_j$, as the realized value of θ that is fixed at θ^0 and estimates the sensitivity of the response Y_j to this realized parameter $\widehat{\theta}_j$ as it varies across n replications. Because this relationship is linear when the distribution of $(Y_j, \widehat{\theta}_j)$ is multivariate normal, the gradient, $\nabla_{\theta^0}\mathrm{E}(Y)$, can be estimated by the ordinary least square (OLS) estimator of the slope coefficient of linear regression of Y_j on $\widehat{\theta}_j$, i.e.,

$$\widehat{\nabla}_{\theta^0}\mathrm{E}(Y) = \widehat{\beta}_{1,\mathrm{OLS}} \text{ when } \widehat{\beta}_{\mathrm{OLS}} = \left(\widehat{\theta}^\top\widehat{\theta}\right)^{-1}\widehat{\theta}^\top\mathbf{Y} = \begin{bmatrix} \widehat{\beta}_{0,\mathrm{OLS}} \\ \widehat{\beta}_{1,\mathrm{OLS}} \end{bmatrix} \tag{4.1}$$

where $\mathbf{Y} = [Y_1, Y_2, \ldots, Y_n]^\top \in \Re^n$ is the vector of output, and

$$\widehat{\theta} = \begin{bmatrix} 1 & \widehat{\theta}_1^\top \\ 1 & \widehat{\theta}_2^\top \\ \vdots & \vdots \\ 1 & \widehat{\theta}_n^\top \end{bmatrix}.$$

In some cases when there is only one input variate observed (i.e., $m_j = 1$) such that $\widehat{\theta}_j$ can not be obtained, we break replications into batches and estimate the parameters within the batches. To be consistent, the observed output is batched with the same batch size to estimate its mean, which is regressed on the within-batch estimators of the parameters to estimate the gradient.

However, the framework in Jiang et al. (2020) is not sufficient for conducting local sensitivity

analysis for all CTrES inputs. In Table 4.1, only sensitivity to Screening_Failure fits perfectly within our previous work. The sensitivity to inputs following a triangular distribution needs a problem-specific direction. The sensitivities to the remaining inputs require new methods. We describe these new challenges and our solutions in four subsections below.

4.4.1 Direction $\vec{\mathbf{d}}$ for Triangular Distribution

The challenge presented by the triangular distribution is that its support depends on the distribution parameters and that makes the gradient of the mean or standard deviation with respect to the input parameters hard to interpret. In this case, the meaningful directions described above are not appropriate. This is an example of a problem-specific direction that we need to determine in the context of CTrES.

Denoting the parameters of a triangular distribution as (a,b,c), where a is the minimum, b is the mode, and c is the maximum, the mean and standard deviation of the distribution are given by

$$\mu = \frac{a+b+c}{3}$$
$$\sigma = \sqrt{\frac{a^2+b^2+c^2-ab-ac-bc}{18}}.$$

For sensitivity with respect to the mean (i.e., MSM), the unit-norm steepest ascent direction of the mean, where the probability density function (pdf) shifts to the right by $\sqrt{3}/3$ unit (i.e., $\vec{\mathbf{d}} = (\sqrt{3}/3, \sqrt{3}/3, \sqrt{3}/3)^\top$), still makes sense for CTrES. Along this direction the mean increases at the fastest rate while the standard deviation is kept constant, i.e., isolating the effect of input-distribution location with minimal change to its spread.

For sensitivity with respect to the standard deviation (i.e., MSSD), we chose a meaningful direction to be the direction where the end points of the pdf move in the opposite direction by the same amount, i.e, $\vec{\mathbf{d}} = (-\sqrt{2}/2, 0, \sqrt{2}/2)^\top$. The triangular distribution has no unique min-mean-change direction because of having more than two parameters. However, this particular min-

mean-change direction is practically meaningful for CTrES because the expert users who provide the parameters are often confident about the mode but not the support of the distribution. Thus, the sensitivity measure that tells users the impact of adjusting the minimum and the maximum of a triangular distribution without affecting the mean or mode is the most useful.

4.4.2 Sensitivity with Respect to Piecewise-constant NSPP

The piecewise-constant NSPP in CTrES consists of two distinct arrival rates, λ_{high} and λ_{low}, over two intervals $[0, L_{\text{high}})$ and $[L_{\text{high}}, T)$, where L_{high} is the duration of the time when the arrival rate is high. The duration L_{high} has a triangular distribution, and T is the time necessary to enroll the required number of patients. Because this piecewise-constant NSPP has two intervals with uncertain length, it is particularly challenging to directly measure the sensitivity with respect to its mean or standard deviation.

As suggested in Morgan et al. (2016), each interval in a piecewise-constant NSPP can be regarded as a single input distribution to the simulation with the observation interval matching the simulation interval. Therefore, the sensitivity with respect to this NSPP can be decomposed into sensitivities with respect to two independent Poisson processes. We describe the Poisson process as interarrival times following exponential distribution so that the corresponding stochastic gradient can be estimated using the method of Wieland and Schmeiser (2006). For exponential distribution $\mu = \sigma$, so we only do sensitivity to the mean of the interarrival time. With this formulation, the sensitivity falls within the framework of Jiang et al. (2020).

The stochastic input L_{high} is problematic because it affects the number of arrivals under the high and low rates. Therefore, we reformulated the sensitivity question to be "How sensitive are the KPIs to the *actual* duration of the time when the arrival rate is high?" To obtain this we simply do a regression of the simulation output on the observed value of L_{high} of all sites and the sensitivities are the corresponding coefficients.

4.4.3 Sensitivity with respect to Bernoulli Distribution

For the inputs following a Bernoulli distribution, only sensitivity with respect to the mean of the input, i.e., $\mu \equiv \mathrm{E}(X) = p$, makes sense, so MSM requires estimating the stochastic gradient $\partial \mathrm{E}(Y)/\partial p$. For Screening_Failure, the screening test results of at least 800 patients are recorded within each replication, so the stochastic gradient we need can be estimated using the method of Wieland and Schmeiser (2006). However, for Startup_Success and Enrollment_Success, only a single outcome (0 or 1) is observed in each replication. Thus, the method of Wieland and Schmeiser (2006) does not apply.

However, notice that when there is a single Bernoulli $\partial \mathrm{E}(Y)/\partial p$ can be derived directly by conditioning on $X \sim \text{Bernoulli}(p)$, i.e.,

$$\begin{aligned} \mathrm{E}[Y] &= \mathrm{E}[Y|X=1]p + \mathrm{E}[Y|X=0](1-p) \\ \Rightarrow \frac{\partial \mathrm{E}[Y]}{\partial p} &= \mathrm{E}[Y|X=1] - \mathrm{E}[Y|X=0]. \end{aligned} \tag{4.2}$$

Expression (4.2) can be estimated directly from the output data by

$$\hat{\eta} = \frac{\partial \widehat{\mathrm{E}}[Y]}{\partial p} = \frac{\sum_{i=1}^{n} Y_i \cdot \mathrm{I}\{X_i = 1\}}{\sum_{i=1}^{n} \mathrm{I}\{X_i = 1\}} - \frac{\sum_{i=1}^{n} Y_i \cdot \mathrm{I}\{X_i = 0\}}{\sum_{i=1}^{n} \mathrm{I}\{X_i = 0\}}.$$

An estimator of the variance of $\hat{\eta}$ is derived in Appendix E, which is needed because the denominators are random variables.

4.4.4 Interacting Inputs

In the context of CTrES, there are inputs that interact with each other. For example, as shown in Figure 4.2, the impact of a site's startup delay and interarrival time only matter if that site starts up successfully and is able to enroll patients. Similarly, the site-specific inputs of a country have impact on the KPIs only when that country starts up successfully. With such interacting inputs, it

is tricky to find an appropriate sensitivity measure.

Specifically, if the result of the country startup is failure, then no variates will be observed from uncertain inputs at the site level for all sites in the country. Similarly, if the startup or enrollment of a site fails, no variates will be observed from the inputs Site_Patient_Arrival or Duration_of_Rate_High for that site. One solution is measuring the sensitivities conditional on the successful startup of the country and all sites, and enrollment at all sites. However, this is a sensitivity conditional on a situation that rarely happens and it does not answer the what-if questions that help with plan management. What CTrES users want is an unconditional sensitivity measure.

Therefore, we propose a new input that considers the interaction among inputs. We demonstrate for the case when $X \sim F(\cdot|\theta)$, $B \sim \text{Bernoulli}(p)$, and $\theta = \text{E}[X]$ with nominal value θ^0. Define a new variable $X' = XB$ which has $\theta' = \text{E}[X'] = p\theta$ and B is the input that interacts with X. Because X' is observable on each replication, we can apply the method of Wieland and Schmeiser (2006) to estimate the stochastic gradient of $\text{E}[Y]$ with respect to θ' using OLS by regressing Y_j on the observed parameter $\widehat{\theta}'_j$, $j = 1, 2, \ldots, n$, i.e.,

$$Y = \beta_0 + \beta_1^\top \widehat{\theta}' + \varepsilon. \tag{4.3}$$

However, if we use the model in (4.3) we have $\nabla_{\theta^0}\text{E}[Y] = p\beta_1$ where β_1 can be estimated via OLS. Thus, for unconditional sensitivity we use $p\widehat{\beta}_{1,\text{OLS}}$ as the estimator of the gradient of $\text{E}[Y]$ with respect to θ.

4.4.5 Dependence Because Total Enrolled Patients is Fixed

A CTrES simulation stops when a fixed number of patients, say 800, is enrolled. This forces a constraint on the number of patients recruited at each open site because they have to total to 800. Therefore, there is functional dependence among the observed arrival processes of open sites. The goal here is to decide how to parameterize the interarrival time such that the dependence works in

our favor for local sensitivity analysis. For the purpose of explaining our solution, suppose each site has only as single arrival rate, instead of high and low.

Let $\widehat{\theta}^{(i)}$ be the observed parameter of the exponential distribution of the interarrival time of site i. Suppose there are S sites where each is affected by its Startup_Success $B^{(i)} \sim \text{Bernoulli}(p^{(i)})$. The regression model for estimating the gradient of $\mathrm{E}[Y]$ with respect to $\theta^{(i)}$ at the nominal setting is given by

$$Y = \beta_0 + \sum_{i=1}^{S} \beta_i^\top B^{(i)} \widehat{\theta}^{(i)} + \varepsilon.$$

Analysis of the model is straightforward if $B^{(i)}$ is independent of $B^{(j)}$ and $B^{(i)}$ is independent of $\widehat{\theta}^{(j)}$ for $i \neq j, \forall i, j$. However, the latter assumption does not hold because the simulation terminates when 800 patients are enrolled.

Specifically, when θ is the rate parameter λ, the arrival counting process of site i, $N^{(i)}(t)$, is Poisson$(\lambda^{(i)} t)$, and the time it takes to enroll 800 patients can be represented as:

$$\begin{aligned} T &= \inf\left\{ t \geq 0 : \sum_{i=1}^{S} B^{(i)} N^{(i)}(t) = 800 \right\} \\ &\Rightarrow \sum_{i=1}^{S} B^{(i)} N^{(i)}(T) = 800. \end{aligned} \tag{4.4}$$

Therefore, the observed arrival rate of site i is $\widehat{\lambda}^{(i)} = B^{(i)} N^{(i)}(T)/T$ and Equation (4.4) is equivalent to $\sum_{i=1}^{S} \widehat{\lambda}^{(i)} = 800/T$, which shows that $\widehat{\lambda}^{(i)}$'s are not independent. If $\widehat{\lambda}^{(i)}$ is larger than expected, the observed rates of other sites must be smaller to compensate. Such dependence among predictors of the regression makes sense from a local sensitivity point of view. Thus, we propose parameterizing the interarrival time by the rate parameter and using $p^{(i)}\widehat{\beta}_i$ as the change in $\mathrm{E}[Y]$ per unit increase in the observed rate of at site i.

If on the other hand, we let $\theta^{(i)}$ be the mean interarrival time $\mu^{(i)}$, then the observed mean interarrival time at site i is given by $\widehat{\mu}^{(i)} = T / \left(B^{(i)} N^{(i)}(T) \right)$ when $B^{(i)} = 1$, and is undefined otherwise. In this case we no longer have the sum of $B^{(i)}\widehat{\mu}^{(i)}$ to be some constant and the relationship

among the $\widehat{\mu}^{(i)}$'s depends on the observed $B^{(i)}$'s. Therefore, the resulting regression coefficients are hard to interpret.

4.4.6 Large Number of Inputs

A CTrES simulation typically involves multiple countries and hundreds of sites, which leads to a huge number of inputs. With so many sensitivities to look at, it is challenging to tease out the inputs that are critical for a better clinical trial enrollment plan. Therefore, backward stepwise regression is used to pre-screen all relevant inputs such that the reduced model only includes the ones that have statistically significant impact on the KPIs.

4.5 An Illustrative Case: One Country with Ten Sites

In this section we illustrate interesting results discovered via local sensitivity analysis on a CTrES simulation with 1 country, 10 sites. This is a realistic case for a clinical trial in the U.S., but specific parameter values were chosen only for demonstration purposes. The country and all sites are subject to the uncertainties specified in Table 4.1; There are 62 stochastic inputs with 135 parameters in total. The two primary KPIs are the mean time it takes to enroll 800 patients (denoted as "TimeToEnrollTarget") and the mean of the implied total cost (denoted as "TotalCost"). The simulation was run for 6000 replications and the estimated mean "TimeToEnrollTarget" and mean "TotalCost" are around 61 weeks and 8 million dollars. Using these 6000 replications we measured the sensitivity of each KPI to the mean and standard deviation of each stochastic input and screened out the unimportant ones.

When we interpret the sensitivity measure, the change in the mean or standard deviation of an input is in its actual units, i.e., in weeks for Startup_Delay, Identification_Delay, and interarrival time, and in percentages for Startup_Success, Enrollment_Success, and Screening_Failure. For ease of representation, we express the units of cost in thousands of dollars (K). Except for L_{high},

where we care about the effect of a change in its *actual* value, for other inputs we focus on the MSM measures because the MSSD measures only make sense for inputs that have a triangular distribution and the change in the standard deviation of those inputs has negligible impact on the KPIs in this particular illustration.

We first consider the sensitivities with respect to L_{high}, which tell us how the KPIs respond to a change in the duration of the time when the arrival rate is high at each site. Notice that there is risk pooling because the change in L_{high} of all sites that are activated successfully might contribute to the change in the KPIs, which by design makes the sensitivity lower. Therefore, we set a threshold for reporting sensitivities to be a negative value significantly different from 0. This is because an increase in the duration of the high rate period increases the average rate of arrivals and thus decreases the KPIs of interest. For the response mean "TotalCost," the sensitivity with the largest magnitude is around $-\$6K$ with respect to L_{high} of site 10, indicating that when the L_{high} of site 10 increases by 1 week, the mean "TotalCost" of enrolling 800 patients is expected to decrease by $-\$6K$. Among all sites, site 10 has the largest impact because it has relatively high arrival rate throughout but incurs the lowest costs of screening and enrolling patients. The sensitivities for mean "TimeToEnrollTarget" are not reported because the coefficients of most of the sites are statistically significant but close to zero.

For MSM results, the site-specific interarrival time is always the most important category among the remaining 52 inputs. Specifically, mean "TimeToEnrollTarget" is sensitive to the interarrival time at all sites, especially during the high rate period at sites 5 and 9. The values are 6 and 12, indicating that 1 week increase in the mean interarrival time at site 5 would increase the mean "TimeToEnrollTarget" by 6 weeks and 12 weeks for site 9. For inputs determining whether a subprocess happens or not, Screening_Failure has the largest impact: 1% increase in the screening failure probability would increase the mean enrollment duration by around 0.8 week.

In summary, to shorten the time it takes to enroll 800 patients, we recommend putting management effort on increasing the patient traffic at sites 5 and 9.

For mean "TotalCost," the sensitivities with respect to the mean interarrival time of sites 5, 7, and 10 during the period with high arrival rate are around $-\$120K$, $-\$90K$, and $\$120K$, respectively, which are the largest among all significant inputs. That is, the mean total cost is expected to decrease by $\$120K$ when the mean interarrival time at site 5 increase by 1 week, and decrease by $\$90K$ for site 7. On the other hand, if the mean interarrival time at site 10 increased by 1 week, the mean total cost would increase by $\$120K$. The opposite impact of an increase in the mean interarrival time at these sites is because of their high arrival rates but different costs of screening and enrolling patients. Among all site, sites 5 and 7 have the highest and site 10 has the lowest cost. Thus, the increase in mean interarrival time at sites 5 or 7 would decrease the proportion of patients get enrolled with high costs and thus reduces the total cost. Comparing sensitivities with respect to the inputs capturing the uncertainties associated with the success of subprocesses, Screening_Failure is the largest, which is around $\$23K$. That is, 1% increase in the screening failure probability comes with the increase in the mean total cost by $\$23K$. Among the 10 sites, sensitivities with respect to the Startup_Success and Enrollment_Success of site 10 have the largest magnitude, i.e., $-\$3K$ and $-\$4K$, respectively.

Therefore, the most efficient way to reduce the total cost is to put management effort on decreasing the patient traffic at site 5 or increasing the patient traffic at site 10. However, because the time to enroll patients is the most important concern of the enrollment plan, the management strategy that is both time and cost efficient is increasing the patient traffic at sites 5, 9, and 10.

4.6 Conclusions

SAS CTrES is a powerful tool for CROs and pharmaceutical companies for clinical trail enrollment planning because it is capable of capturing all the uncertainties throughout the process and quantifying the risk in the cost and enrollment prediction beyond the traditional deterministic solutions. However, CTrES lacks the capability to quickly answer the what-if questions that are important

for problem diagnosis and management of a clinical trial. We extend the framework in Jiang et al. (2020) and enable CTrES to conduct local sensitivity analysis to answers the what-if questions for any number of stochastic inputs without running addition simulations beyond the basic scenario. Instead of directly opening more sites to improve only the most important KPI, the time it takes to enroll a given target number of patients, the sensitivity measures suggest smart resource and management effort allocation strategies that are both time efficient and cost efficient.

Bibliography

Abbas, I., J. Rovira, and J. Casanovas (2007). Clinical rial optimization: Monte Carlo simulation Markov model for planning clinical trials recruitment. *Contemporary Clinical Trials 28*(3), 220–231.

Anisimov, V. V. (2008). Using mixed Poisson models in patient recruitment in multicentre clinical trials. In *Proceedings of the World Congress on Engineering*, Volume 2, London, U.K., pp. 1046–1049.

Anisimov, V. V. (2009). Predictive modelling of recruitment and drug supply in multicenter clinical trials. In *Proceedings of Joint Statistical Meeting*, Washington, D.C., pp. 1248–1259.

Anisimov, V. V. (2016). Predictive hierarchic modeling of operational characteristics in clinical trials. *Communications in Statistics-Simulation and Computation 45*(5), 1477–1488.

Barton, R. R., S. E. Chick, R. C. H. Cheng, S. G. Henderson, A. M. Law, B. W. Schmeiser, L. M. Leemis, L. W. Schruben, and J. R. Wilson (2002). Panel discussion on current issues in input modeling. In *Proceedings of the 2002 Winter Simulation Conference*, pp. 353–369. IEEE.

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

Biller, B., A. Mokashi, I. Oliveira, S. Pathan, J. Yi, and J. Box (2019). Time travel into the future

of clinical trial enrollment design. In *Proceedings of the SAS Global Forum 2019 Conference*, Cary, NC: SAS Institute Inc.

Box, J. (2018). Simulating enrollment plans. In *Proceedings of the 2018 PhUSE EU Connect Paper TT08*, Cary, NC: SAS Institute Inc.

Burt, J. M. and M. B. Garman (1971). Conditional monte carlo: A simulation technique for stochastic network analysis. *Management Science 18*(3), 207–217.

Carole, B. and S. Vanduffel (2015). Quantile of a mixture with application to model risk assessment. *Dependence Modeling 3*(1), 172–181.

Cochran, W. G. (2007). *Sampling Techniques*. John Wiley & Sons.

Cognizant (2015). Patient recruitment forecast in clinical trials. https://www.cognizant.com/whitepapers/patients-recruitment-forecast-in-clinical-trials-codex1382.pdf. Accessed: 2020-05-18.

Conway, R. W. (1963). Some tactical problems in digital simulation. *Management Science 10*(1), 47–61.

Fu, M. C. (2015). Stochastic gradient estimation. In M. C. Fu (Ed.), *Handbook of Simulation Optimization*, Volume 216 of *International Series in Operations Research & Management Science*, pp. 105–147. New York: Springer.

Goodnight, J. (2007). SAS. http:ceoroundtableoncancer.org/members/sas. Accessed: 2020-05-18.

Handelsman, D. (2012). Applying business analytics to optimize clinical research operations. In *Proceedings of the SAS Global Forum 2012 Conference*. SAS Institute Inc.

Hansen, B. E. and J. S. Racine (2012). Jackknife model averaging. *Journal of Econometrics 167*(1), 38–46.

Hopp, W. J. and M. L. Spearman (2011). *Factory Physics*. Long Grove, Illinois: Waveland Press.

Hughes, E., R. Pratt, and B. Biller (2018). Solving business problems with SAS analytics and OPTMODEL. Technology workshop presented at INFORMS Analytics Conference, Baltimore, MD.

Jiang, W. X., B. L. Nelson, and L. J. Hong (2019). Estimating sensitivity to input model variance. In *Proceedings of the 2019 Winter Simulation Conference*, National Harbor, MD, pp. 3705–3716. IEEE.

Jiang, X., B. Biller, and B. L. Nelson (2020). Simulation sensitivity analysis for clinical trial enrollment planning. *INFORMS Journal on Applied Analytics*, under review.

Karian, Z. A. and E. J. Dudewicz (2000). *Fitting Statistical Distributions: The Generalized Lambda Distribution and Generalized Bootstrap Methods*. New York: CRC Press.

Karvanen, J. (2006). Estimation of quantile mixtures via l-moments and trimmed l-moments. *Computational Statistics & Data Analysis 51*(2), 947–959.

Kouvelis, P., J. Milner, and Z. Tian (2017). Clinical trials for new drug development: Optimal investment and application. *Manufacturing & Service Operations Management 19*(3), 437–452.

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

Law, A. M. and W. D. Kelton (1991). *Simulation Modeling and Analysis* (2 ed.). New York: McGraw-Hill.

L'Ecuyer, P. (1990). A unified view of the IPA, SF, and LR gradient estimation techniques. *Management Science 36*(11), 1364–1383.

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

McLachlan, G. and D. Peel (2004). *Finite Mixture Models*. New York: John Wiley & Sons.

Mijoule, G., S. Savy, and N. Savy (2012). Models for patients' recruitment in clinical trials and sensitivity analysis. *Statistics in Medicine 31*(16), 1655–1674.

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

Nelson, B. (2013). *Foundations and Methods of Stochastic Simulation: A First Course*. New York: Springer Science & Business Media.

Nelson, B. L. (2016). some tactical problems in digital simulationfor the next 10 years. *Journal of Simulation 10*(1), 2–11.

Nelson, B. L., X. Jiang, A. Wan, and X. Zhang (2020). Reducing simulation input-model risk via input model averaging. *INFORMS Journal on Applied Analytics*. forthcoming.

Nocedal, J. and S. J. Wright (2006). *Numerical Optimization* (2 ed.). New York: Springer.

Saltelli, A., K. Chan, and E. M. Scott (2000). *Sensitivity Analysis*. New York: John Wiley & Sons.

Schoemig, A. K. (1999). On the corrupting influence of variability in semiconductor manufacturing. In *Proceedings of the 1999 Winter*, pp. 837–842. IEEE.

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

Wagner, M. A. F. and J. R. Wilson (1996). Using univariate bézier distributions to model simulation input processes. *IIE Transactions 28*(9), 699–711.

Wieland, J. R. and B. W. Schmeiser (2006). Stochastic gradient estimation using a single design point. In *Proceedings of the 2006 Winter Simulation Conference*, Piscataway, NJ, pp. 390–397. IEEE.

Zhao, H., E. Huang, R. Dou, and K. Wu (2019). A multi-objective production planning problem with the consideration of time and cost in clinical trials. *Expert Systems with Applications 124*, 25–38.

Zhao, H., K. Wu, and E. Huang (2018). Clinical trial supply chain design based on the Pareto-optimal trade-off between time and cost. *IISE Transactions 50*(6), 512–524.

Appendix A

Derivation of $\widehat{\partial}_{\mathrm{LR}}\mathrm{Var}(Y)/\partial\theta_0$

Using standard LR reasoning, the LR gradient estimator of $\mathrm{E}(Y^2)$ with respect to θ in the one-dimensional context is given by

$$Y^2(\theta)\frac{\partial \ln f(X|\theta)}{\partial \theta}.$$

Averaging across n replications, the gradient estimator of $\mathrm{E}(Y^2)$ with respect to θ is

$$\frac{\widehat{\partial}_{\mathrm{LR}}\mathrm{E}(Y^2)}{\partial\theta_0} = \frac{1}{n}\sum_{j=1}^{n} Y_j^2(\theta_0)\frac{\partial \ln f\left(X_j|\theta_0\right)}{\partial\theta}. \tag{A.1}$$

Using insight (**??**) and estimating $\mathrm{E}(Y)$ by $\bar{Y}$, we can obtain the a gradient estimator of $\mathrm{Var}(Y)$ with respect to θ at nominal setting in (3.12) after plugging in the estimator of $\nabla_{\theta_0}\mathrm{E}(Y)$ from (3.11) and the estimator of $\nabla_{\theta_0}\mathrm{E}(Y^2)$ from (A.1).

Notice that we could do all pairwise differences to estimate $\mathrm{Var}(Y)$, i.e.,

$$s_Y^2 = \frac{1}{2\binom{n}{2}}\sum_{i\neq j}\left(Y_i - Y_j\right)^2 = \frac{1}{n(n-1)}\sum_{i\neq j}\left(Y_i - Y_j\right)^2,$$

where the gradient of each summand with respect to θ can be estimated using LR reasoning as

$$(Y_i - Y_j)^2 \frac{\partial \ln \left[f(X_i|\theta_0) f(X_j|\theta_0) \right]}{\partial \theta}.$$

Thus, the resulting gradient estimator is

$$\frac{\widehat{\partial}_{\mathrm{LR}} \mathrm{Var}(Y)}{\partial \theta_0} = \frac{1}{n(n-1)} \sum_{i \neq j} \left(Y_i - Y_j \right)^2 \frac{\partial \ln \left[f(X_i|\theta_0) f(X_j|\theta_0) \right]}{\partial \theta} \ .$$

Although this gradient estimator is unbiased, it is expensive to compute.

Appendix B

LR Gradient Estimators for SAN with Output $\mathrm{E}(Y)$

Distribution of X	Parameter	LR Gradient from jth Replication
exponential	μ (mean)	$Y_j(X_j-\mu)/\mu^2$
weibull	k (shape)	$Y_j\left(\frac{1}{k}+\log X_j-\log\lambda-\log\frac{X_j}{\lambda}\left(\frac{X_j}{\lambda}\right)^k\right)$
weibull	λ (scale)	$Y_j\left(-\frac{k}{\lambda}+\frac{k}{\lambda}\left(\frac{X_j}{\lambda}\right)^k\right)$
gamma	α (shape)	$Y_j\left(\log\beta\digamma(\alpha)+\log X_j\right)$
gamma	β (scale)	$Y_j\left(\frac{\beta}{\alpha}-X_j\right)$

Appendix C

Variance Estimators

This appendix derives variance estimators associated with our sensitivity point estimators.

C.1 Estimating Variance of the $\widehat{\nabla}_{\theta_0}\mathrm{E}(Y)_{\mathrm{FD}}$ and $\widehat{\nabla}_{\theta_0}\mathrm{E}(Y)_{\mathrm{LR}}$

We consider estimators of the variances of $\widehat{\nabla}_{\theta_0}\mathrm{E}(Y)_{\mathrm{FD}}$ and $\widehat{\nabla}_{\theta_0}\mathrm{E}(Y)_{\mathrm{LR}}$ together because both belong to *Setting 1* where we have i.i.d. observations of the corresponding $\widehat{\nabla}_{\theta_0}$ so that we can compute its sample variance-covariance matrix. Specifically, extending the expression in (3.9) to $\theta = (\vartheta_1, \vartheta_2, \ldots, \vartheta_p)^\top$, where ϑ_i's are individual components of θ, the FD gradient estimator is an average of the i.i.d. observations of the basic FD gradient estimator, $\mathrm{FD}(\theta_0)_j = (\mathrm{FD}(\vartheta_1)_j, \mathrm{FD}(\vartheta_2)_j, \ldots, \mathrm{FD}(\vartheta_p)_j)^\top$, across n replications, where $\mathrm{FD}(\vartheta_k)_j = (Y_{n+j}(\theta_0 + \Delta\vartheta_k) - Y_j(\theta_0))/\Delta\vartheta_k$. Thus, the $p \times p$ sample variance-covariance matrix of $\mathrm{FD}(\theta_0)$ divided by n is an estimator of the variance-covariance matrix of the FD gradient estimator.

Similarly, based on the expression in (3.11), the LR gradient estimator for $\theta \in \Re^p$ is also an average of the i.i.d. observations of the basic LR gradient estimator, $\mathrm{LR}(\theta_0)_j = Y(\theta_0)_j \nabla_{\theta_0} \ln f(X_j)$, $j = 1, 2, \ldots, n$. Thus, the variance-covariance matrix of the LR gradient estimator can also be estimated by the sample variance-covariance matrix of $\mathrm{LR}(\theta_0)$ divided by n.

C.2 Estimating the Variance of the $\widehat{\nabla}_{\theta_0}\mathrm{E}(Y)_{\mathrm{WS}}$

When using the method of Wieland and Schmeiser (2006), we regress Y on $\widehat{\Theta}$ to estimate the gradient of $\mathrm{E}(Y)$ with respect to all input parameters,

$$\nabla_{\Theta_0}\mathrm{E}(Y) = \left(\nabla^{\top}_{\theta_0^{(1)}}\mathrm{E}(Y), \nabla^{\top}_{\theta_0^{(2)}}\mathrm{E}(Y), \ldots, \nabla^{\top}_{\theta_0^{(K)}}\mathrm{E}(Y)\right)^{\top},$$

where $\nabla^{\top}_{\theta_0^{(i)}}$ is the gradient with respect to the parameters of ith input distribution. Assuming the joint distribution of $(Y, \widehat{\Theta})$ is multivariate normal, we have the correct regression model and the gradient estimator is the OLS estimator of the slope coefficients. This case belongs to *Setting 2* and the variance-covariance matrix can be estimated by $\widehat{\mathbf{V}}$ in (3.16) where the predictor variable $\mathbf{x}$ is $\widehat{\Theta}$.

C.3 Estimating Variance of $\widehat{\nabla}_{\theta_0}\mathrm{Var}(Y)_{\mathrm{FD}}$ and $\widehat{\nabla}_{\theta_0}\mathrm{Var}(Y)_{\mathrm{LR}}$

Estimation of the variances of $\widehat{\nabla}_{\theta_0}\mathrm{Var}(Y)_{\mathrm{FD}}$ and $\widehat{\nabla}_{\theta_0}\mathrm{Var}(Y)_{\mathrm{LR}}$ are similar and straightforward because in both cases the $\widehat{\nabla}_{\theta_0}$ is, or at least can be approximated as, an average of i.i.d. observations of the basic gradient estimator.

Extended from (3.10), the FD estimator of $\widehat{\nabla}_{\theta_0}\mathrm{Var}(Y)_{\mathrm{FD}}$ is an average of the i.i.d. observations of $\mathrm{FD}(\theta_0)_\ell = (\mathrm{FD}(\vartheta_1)_\ell, \mathrm{FD}(\vartheta_2)_\ell, \ldots, \mathrm{FD}(\vartheta_p)_\ell)^{\top}$, for $\ell = 1, 2, \ldots, k$, where $\mathrm{FD}(\vartheta_k)_\ell = \left(S^2_{k+\ell}(\theta_0 + \Delta\vartheta_k) - S^2_\ell(\theta_0)\right)/\Delta\vartheta_k$ is obtained within the ℓth batch. Hence, this case belongs to *Setting 1* and the estimator of the variance-covariance matrix of $\widehat{\nabla}_{\theta_0}\mathrm{Var}(Y)_{\mathrm{FD}}$ is the sample variance-covariance matrix of $\mathrm{FD}(\theta_0)$ divided by k.

The LR gradient estimator in (3.12) belongs to *Setting 3* and is approximated as an average of the i.i.d. observations of the basic LR gradient estimator, $\mathrm{LR}(\theta_0)_j = \left(Y_j^2(\theta_0) - 2\bar{Y}Y_j(\theta_0)\right)\nabla_{\theta_0}\ln f(X_j)$, across n replications. Thus, the variance-covariance matrix of $\nabla_{\theta_0}\mathrm{Var}(Y)$ can be estimated in the same way as for the LR gradient estimator of $\nabla_{\theta_0}\mathrm{E}(Y)$.

However, when approximating *Setting 3* as *Setting 1*—that is, treating $\bar{Y}$ as constant—the resulting variance estimator is not generally consistent because of non-negligible covariances terms.

C.4 Estimating Variance of $\widehat{\nabla}_{\theta_0}\mathrm{Var}(Y)_{\mathrm{WS}}$

If we use the WS method together with batching to set up the linear regression, the variance estimated within batches, S^2, is regressed on the batch means of $\widehat{\Theta}$ with batch size b, $\widehat{\Theta}(b)$, to estimate the complete gradient

$$\nabla_{\Theta_0}\mathrm{Var}(Y) = \left(\nabla^{\top}_{\theta_0^{(1)}}\mathrm{Var}(Y), \nabla^{\top}_{\theta_0^{(2)}}\mathrm{Var}(Y), \ldots, \nabla^{\top}_{\theta_0^{(K)}}\mathrm{Var}(Y)\right)^{\top}.$$

Under the assumption the joint distribution of $(S^2, \widehat{\Theta}(b))$ is multivariate normal, we have *Setting 2* where the response vector $\mathbf{Y} = [S_1^2, S_2^2, \ldots, S_k^2]^{\top}$, $\mathbf{x}$ is $\widehat{\Theta}(b)$.

Alternatively we can do multi-response regression, i.e., regressing both Y and Y^2 on $\widehat{\Theta}$, to obtain the gradient estimator $\widehat{\nabla}_{\Theta_0}\mathrm{Var}(Y) = \widehat{\nabla}_{\Theta_0}\mathrm{E}(Y^2) - 2\bar{Y}\widehat{\nabla}_{\Theta_0}\mathrm{E}(Y)$, which is a linear combination of two OLS estimators. Again, if we treat $\bar{Y}$ as constant, then the variance-covariance matrix of the gradient estimator is

$$\begin{aligned}\mathrm{Var}\left(\widehat{\nabla}_{\Theta_0}\mathrm{Var}(Y)_{\mathrm{WS}}\right) = \mathrm{Var}\left(\widehat{\nabla}_{\Theta_0}\mathrm{E}(Y^2)\right) + 4\overline{Y}^2\mathrm{Var}\left(\widehat{\nabla}_{\Theta_0}\mathrm{E}(Y)\right) \\ - 4\overline{Y}\mathrm{Cov}\left(\widehat{\nabla}_{\Theta_0}\mathrm{E}(Y^2), \widehat{\nabla}_{\Theta_0}\mathrm{E}(Y)\right).\end{aligned} \tag{C.1}$$

Specifically, we have two regression models:

$$\begin{aligned}\mathbf{Y} &= \mathbf{X}\beta + \varepsilon_1 \\ \mathbf{Y}^2 &= \mathbf{X}\alpha + \varepsilon_2\end{aligned}$$

where

$$\mathbf{X} = \begin{bmatrix} 1 & \widehat{\Theta}_1^\top \\ 1 & \widehat{\Theta}_2^\top \\ \vdots & \vdots \\ 1 & \widehat{\Theta}_n^\top \end{bmatrix}$$

is common to both models, $\widehat{\Theta}_j$ is $\widehat{\Theta}$ estimated within the jth replication, $\mathbf{Y} = [Y_1, Y_2, \ldots, Y_n]^\top$, and $\mathbf{Y}^2 = [Y_1^2, Y_2^2, \ldots, Y_n^2]^\top$. We estimate $\nabla_{\Theta_0}\mathrm{E}(Y)$ and $\nabla_{\Theta_0}\mathrm{E}(Y^2)$ by the OLS estimator of the slope coefficients of the two models above, i.e., $\beta_{1,\mathrm{OLS}}$ and $\alpha_{1,\mathrm{OLS}}$. Since both cases belong to *Setting 2*, assuming the joint distributions of $(Y, Y^2, \widehat{\Theta})$ is multivariate normal, and $\mathbf{Y} \perp \varepsilon_1$ and $\mathbf{Y}^2 \perp \varepsilon_2$, then the variance-covariance matrices of these two OLS estimators, and the covariance matrix between them, can be estimated from regression, which gives us all the terms needed for estimating the variance-covariance matrix in (C.1). The complete derivation is in Appendix D. Of course, this estimator is biased because the covariance between $\widehat{\nabla}_{\Theta_0}\mathrm{E}(Y^2)$ and $2\bar{Y}\widehat{\nabla}_{\Theta_0}\mathrm{E}(Y)$ and between $\bar{Y}$ and $\widehat{\nabla}_{\Theta_0}\mathrm{E}(Y)$ are not taken into account.

Appendix D

Complete Derivation of Variance of Multi-response Regression

When writing $\mathrm{Var}(Y) = \mathrm{E}(Y^2) - \mathrm{E}^2(Y)$ and set up multivariate regression to estimate $\nabla_{\Theta_0}\mathrm{Var}(Y)$, the gradient estimator is given by

$$\widehat{\nabla}_{\Theta_0}\mathrm{Var}(Y)_{\mathrm{WS}} = \widehat{\nabla}_{\Theta_0}\mathrm{E}(Y^2)_{\mathrm{WS}} - 2\bar{Y}\widehat{\nabla}_{\Theta_0}\mathrm{E}(Y)_{\mathrm{WS}}$$

where $\widehat{\nabla}_{\Theta_0}\mathrm{E}(Y^2)_{\mathrm{WS}} = \widehat{\Sigma}_{Y^2,\widehat{\Theta}}\left(\widehat{\Sigma}_{\widehat{\Theta},\widehat{\Theta}}\right)^{-1}$ is equivalent to the OLS estimator of the slope coefficients of the multiple linear regression of Y^2 on $\widehat{\Theta}$, and $\widehat{\nabla}_{\Theta_0}\mathrm{E}(Y)_{\mathrm{WS}} = \widehat{\Sigma}_{Y,\widehat{\Theta}}\left(\widehat{\Sigma}_{\widehat{\Theta},\widehat{\Theta}}\right)^{-1}$ is equivalent to the OLS estimator of the slope coefficients of the multiple linear regression of Y on $\widehat{\Theta}$. Treating $\bar{Y}$ as constant, the variance-covariance matrix of the gradient estimator is simplified to

$$\begin{aligned}\mathrm{Var}\left(\widehat{\nabla}_{\Theta_0}\mathrm{Var}(Y)_{\mathrm{WS}}\right) = \mathrm{Var}\left(\widehat{\nabla}_{\Theta_0}\mathrm{E}(Y^2)\right) + 4\overline{Y}^2\mathrm{Var}\left(\widehat{\nabla}_{\Theta_0}\mathrm{E}(Y)\right) \\ - 4\overline{Y}\mathrm{Cov}\left(\widehat{\nabla}_{\Theta_0}\mathrm{E}(Y^2), \widehat{\nabla}_{\Theta_0}\mathrm{E}(Y)\right).\end{aligned}$$

Assuming the relationship between Y and $\widehat{\Theta}$ and between Y^2 and $\widehat{\Theta}$ are both linear, both cases belong to *Setting 2* where $\mathrm{Var}\left(\widehat{\nabla}_{\Theta_0}\mathrm{E}(Y)\right)$ can be estimated by $\widehat{\mathbf{V}}$ in (3.16) with $\mathbf{x} = \widehat{\Theta}$, and

$\text{Var}\left(\widehat{\nabla}_{\Theta_0}\text{E}(Y^2)\right)$ can be estimated by $\widehat{\mathbf{V}}$ in (3.16) with the response vector $\mathbf{Y}^2 = [Y_1^2, Y_2^2, \ldots, Y_n^2]^\top$ and the predictor variable $\mathbf{x} = \widehat{\Theta}$.

We can derive, in a similar manner to Jiang et al. (2019), that

$$\text{Cov}(\widehat{\nabla}_{\Theta_0}\text{E}(Y)_{\text{WS}}, \widehat{\nabla}_{\Theta_0}\text{E}(Y^2)_{\text{WS}}) = \frac{\sigma_{\varepsilon_{Y,Y^2}}}{n-q-2}\Sigma_{\widehat{\Theta},\widehat{\Theta}}$$

where $\sigma^2_{\varepsilon_Y}$ is the covariance between ε_1 (Y given $\widehat{\Theta}$) and ε_2 (Y^2 given $\widehat{\Theta}$).

Therefore,

$$\frac{\text{Cov}\left(\widehat{\nabla}_{\Theta_0}\text{E}(Y^2)_{\text{WS}}, \widehat{\nabla}_{\Theta_0}\text{E}(Y)_{\text{WS}}\right)}{\text{Var}\left(\widehat{\nabla}_{\Theta_0}\text{E}(Y)_{\text{WS}}\right)} = \frac{\sigma_{\varepsilon_{Y,Y^2}}}{\sigma^2_{\varepsilon_Y}}.$$

where $\sigma^2_{\varepsilon_Y}$ is the variance of ε_1 (Y given $\widehat{\Theta}$). Because $\sigma_{\varepsilon_{Y,Y^2}}$ can be estimated by $s_{\varepsilon_{Y,Y^2}}$, the sample covariance between the residuals of the multiple linear regression of Y on $\widehat{\Theta}$ and the residuals of the multiple linear regression of Y^2 on $\widehat{\Theta}$, and $\sigma^2_{\varepsilon_Y}$ can be estimated by $s^2_{\varepsilon_Y}$, the sample variance of the residuals of the multiple linear regression of Y on $\widehat{\Theta}$, we can estimate $\text{Cov}\left(\widehat{\nabla}_{\Theta_0}\text{E}(Y^2), \widehat{\nabla}_{\Theta_0}\text{E}(Y)_{\text{WS}}\right)$ by

$$\frac{\widehat{\text{Var}}\left(\widehat{\nabla}_{\Theta_0}\text{E}(Y)\right) s_{\varepsilon_{Y,Y^2}}}{s^2_{\varepsilon_Y}}.$$

Therefore, the estimator of the variance-covariance matrix of the complete WS gradient estimator is

$$\widehat{\text{Var}}\left(\widehat{\nabla}_{\Theta_0}\text{Var}(Y)_{\text{WS}}\right) = \left(\frac{s^2_{\varepsilon,\text{E}2}}{n-q-2} + 4\bar{Y}\left(\bar{Y} - \frac{s_{\varepsilon_{Y,Y^2}}}{s^2_{\varepsilon_Y}}\right)\frac{s^2_{\varepsilon,\text{E}}}{n-q-2}\right)\left(\widehat{\Sigma}_{\widehat{\Theta},\widehat{\Theta}}\right)^{-1}.$$

Appendix E

Variance of $\partial\widehat{\mathrm{E}}[Y]/\partial p$

Let $X \sim \text{Bernoulli}(p)$ be the input, X_i (0 or 1) be the observed X within the ith replication, and n be the total number of replications. Let K be the number of X_i that are equal to 1; then we know that $K \sim \text{Bin}(n, p)$. Based on Cochran (2007), conditional on $K \neq 0$ and $K \neq n$, we have for large n and to the order of n^{-2}:

$$\mathrm{E}\left(\frac{1}{K}\middle| K \neq 0, K \neq n\right) = \frac{1-(1-p)^n}{(1-p^n-(1-p)^n)np} + \frac{(1-p)(1-(1-p)^n)}{(1-p^n-(1-p)^n)n^2p^2}. \tag{E.1}$$

Similarly, we have

$$\mathrm{E}\left(\frac{1}{n-K}\middle| K \neq 0, K \neq n\right) = \frac{1-(1-p)^n}{(1-p^n-(1-p)^n)n(1-p)} + \frac{p(1-(1-p)^n)}{(1-p^n-(1-p)^n)n^2(1-p)^2}. \tag{E.2}$$

Now let Y be the simulation output as a function of the input X, i.e., $Y|X=1 \sim (\mu_1, \sigma_1^2)$ and

$Y|X=0 \sim (\mu_0, \sigma_0^2)$. Because K is random, the variance of $\hat{\eta}$ can be written as

$$\begin{aligned} \text{Var}(\hat{\eta}) &= \text{E}[\text{Var}(\hat{\eta}|K)] + \text{Var}[\text{E}(\hat{\eta}|K)] \\ &= \text{E}[\text{Var}(\hat{\eta}|K)] \\ &= \text{E}\left[\frac{\sigma_1^2}{K} + \frac{\sigma_0^2}{n-K}\right]. \end{aligned} \tag{E.3}$$

The second equation is because $\text{Var}[\text{E}(\hat{\eta}|K)] = \text{Var}[\eta] = 0$. Plugging (E.1) and (E.2) in Expression (E.3) we have

$$\text{Var}\left(\hat{\eta}\middle|K \neq 0, K \neq n\right) \equiv \frac{1-(1-p)^n}{1-p^n-(1-p)^n}\left(\frac{\sigma_1^2}{np} + \frac{\sigma_0^2}{n(1-p)} + \frac{\sigma_1^2(1-p)}{n^2p^2} + \frac{\sigma_0^2 p}{n^2(1-p)^2}\right).$$