USING METAMODELING TO APPROXIMATE
COST-EFFECTIVENESS ACCEPTABILITY CURVES OF
COST-EFFECTIVENESS ANALYSIS

A Thesis in
Industrial Engineering
by
Jingyi Xie

© 2020 Jingyi Xie

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

May 2020
The thesis of Jingyi Xie was reviewed and approved by the following:

Qiushi Chen  
Assistant Professor of Industrial and Manufacturing Engineering  
Thesis Advisor

Ethan Xingyuan Fang  
Assistant Professor of Statistics

John M. Carroll  
Distinguished Professor of Information Sciences and Technology

Ling Rothrock  
Professor of Industrial and Manufacturing Engineering  
Interim Head of the Department of Industrial and Manufacturing Engineering
Abstract

Cost-effectiveness Acceptability Curves (CEACs) based on probabilistic sensitivity analysis (PSA) are commonly used to represent the model uncertainty in cost-effectiveness analysis (CEA). Conducting PSA for complex simulation models could be computationally challenging. Metamodeling is a commonly used approach to simplify the model representation and to reduce the computation burden, which has been found useful in sensitivity/threshold analysis and value of information analysis. The objective of this study is to explore whether metamodels can be used to represent the model uncertainty and approximate CEACs in CEA.

We adapted the microsimulation model from a tutorial example in a literature. To develop metamodels for the original microsimulation model, we trained four classes of metamodels, including linear regression models (with up to second-order polynomials), LASSO with two-way interaction terms, Generalized Additive Models and neural networks, on independent training sets of 5000 or fewer PSA samples. Using an independent testing set of 10,000 PSA samples, we calculated the “true” CEAC as the gold standard, and metamodel-based CEACs based on predicted model outcomes by metamodels. To compare the approximated CEACs with the true CEAC, we defined the approximating error as the maximum absolute value of the differences between the two curves over a willingness-to-pay range $0-150,000 per quality-adjusted life-years. The analysis was repeated for 20 times to account for sampling randomness. To evaluate the robustness of the metamodeling approach, we explored different experiment designs. The analysis was further extended with two comparators to three comparators in CEA. We considered different sampling methods including PSA, Latin hypercube sampling (LHS) and full factorial design to generate training sets. In addition, we tested the metamodeling approach in a real-world case study using a CEA from a published study.

In our base case analysis with two comparators, we found that linear metamodel generated CEACs with an average approximation error 0.91% compared with the true CEAC when using 5000 training samples. Introducing nonlinearity in metamodel reduced the approximation error. The linear model with second-order polynomials resulted in error 0.26%. More complex models like Generalized Additive Models and neural network might not necessarily improve the result. With fewer training samples, the approximation error increased slightly. Metamodels showed the similar performance of approximating CEACs in the cases with three comparators. The approximation errors from PSA and LHS differed slightly and they were below 1.50% for most of the cases we tested.

In conclusion, we demonstrated that metamodels could closely approximate the CEAC results in PSA of CEA. Metamodeling is a promising approach to effectively reduce the computational burden of calculating CEAC from PSA for complex simulation models.
# Table of Contents

List of Figures vi

List of Tables viii

Acknowledgments ix

Chapter 1
  Introduction 1

Chapter 2
  Literature Review 3
    2.1 Cost-Effectiveness Analysis 3
    2.2 Probabilistic Sensitivity Analysis and Cost-effectiveness Acceptability Curve 3
    2.3 Metamodel 7
    2.4 Microsimulation 8

Chapter 3
  Study Design 10
    3.1 Metamodeling Framework 10
    3.2 Different Numbers of Comparators 12
    3.3 Types of Metamodels 15
      3.3.1 Linear Model (LM) 16
      3.3.2 Linear Model with Two-Factor Interaction Terms (LM2) 17
      3.3.3 LASSO Regression with Two-Way Interaction Terms 17
      3.3.4 Generalized Additive Model (GAM) 17
      3.3.5 Neural Networks 18
    3.4 Sampling Methods for Training Sets 19
      3.4.1 Probabilistic Sensitivity Analysis (PSA) 19
      3.4.2 Full Factorial Design 19
      3.4.3 Latin Hypercube Sampling (LHS) 20
    3.5 Microsimulation Models 20
      3.5.1 Method Overview 20
      3.5.2 The Sick-Sicker Microsimulation Model 21
      3.5.3 Model Input 22
Chapter 4
Computational Study

4.1 Base Case Results of Two Comparators
4.1.1 Linear Model (LM)
4.1.2 Linear Model with Two-Way Interaction Terms (LM2)
4.1.3 LASSO Regression with Two-Way Interaction Terms
4.1.4 Generalized Additive Model (GAM)
4.1.5 Neural Networks

4.2 Base Case Results of Three Comparators
4.2.1 Linear Model (LM)
4.2.2 Linear Model with Two-Way Interaction Terms (LM2)
4.2.3 LASSO Regression with Two-Way Interaction Terms
4.2.4 Generalized Additive Model (GAM)
4.2.5 Neural Networks

4.3 Effect of CEAC Shapes
4.4 Effect of Sampling Schemes
4.5 Real-World Cost-Effectiveness Analysis Study: Lymphoma Treatment Model

Chapter 5
Discussion and Conclusion

Appendix A
Input Parameters for Alternative Cases of Two Comparators

Appendix B
Input Parameters for Alternative Case of Three Comparators

Bibliography
List of Figures

2.1 Cost-effectiveness plane for new treatment compared with current practice. 5

3.1 Overall Framework .................................................. 11

3.2 Schematic diagram of difference between true CEAC and approximated CEACs ........................................ 12

3.3 Cost-effectiveness acceptability curves of Case A and Case C ........ 13

3.4 Cost-effectiveness acceptability curves of Case H and Case F ........ 13

3.5 Cost-effectiveness acceptability curves of Case E and Case G ........ 14

3.6 Cost-effectiveness acceptability curves of base case of three comparators .... 14

3.7 Cost-effectiveness acceptability curves of alternative case of three comparators ........................................ 15

3.8 Schematic representation of a simple neural network. .............. 18

3.9 Schematic representation of the sick-sicker microsimulation model .... 22

4.1 (a) True cost-effectiveness acceptability curve and approximated cost-effectiveness acceptability curves of Case A from LM metamodels when using 5000 training samples, (b) Difference between true cost-effectiveness acceptability curve and approximated cost-effectiveness acceptability curves of Case A from LM metamodels when using 5000 training samples. .... 26
4.2 (a) True cost-effectiveness acceptability curve and approximated cost-effectiveness acceptability curves of Case A from LM metamodels when using 200 training samples, (b) Difference between true cost-effectiveness acceptability curve and approximated cost-effectiveness acceptability curves of Case A from LM metamodels when using 200 training samples. ....... 27

4.3 (a) True cost-effectiveness acceptability curve and approximated cost-effectiveness acceptability curves of Case A from LM2 metamodels terms when using 5000 training samples, (b) Difference between true cost-effectiveness acceptability curve and approximated cost-effectiveness acceptability curves of Case A from LM2 metamodels when using 5000 training samples. ................. 28

4.4 (a) True cost-effectiveness acceptability curve and approximated cost-effectiveness acceptability curves of 3 treatments from LM2 metamodels when using 1000 training samples, (b) Difference between true cost-effectiveness acceptability curve and approximated cost-effectiveness acceptability curves of 3 treatments from LM2 metamodels when using 1000 training samples. ......................................................... 31

4.5 Cost-effectiveness acceptability curves of three lymphoma treatments from published paper [1]. ................................................................. 36

4.6 True cost-effectiveness acceptability curves and approximated cost-effectiveness acceptability curves of three lymphoma treatments from LM metamodels when using 500 training samples. ................................. 37

4.7 True cost-effectiveness acceptability curves and approximated cost-effectiveness acceptability curves of three lymphoma treatments from LASSO metamodels when using 500 training samples. ................................. 38
List of Tables

3.1 Input Parameters for Case A .................................................. 23
3.2 Base Case Input Parameters of Treatment I & II in Three Comparators . 24
3.3 Base Case Input Parameters of Treatment III in Three Comparators . . 24
4.1 Base Case Approximation Errors (%) of Two Comparators .............. 25
4.2 Base Case Approximation Errors (%) of Three Comparators ............ 30
4.3 Approximation Errors (%) of Different Shapes of CEAC ................. 35
4.4 Approximation Errors (%) of Different Sampling Schemes .............. 35
4.5 Approximation Errors (%) of Real-World Data ............................ 36
A.1 Input Parameters of Case H .................................................... 43
A.2 Input Parameters of Treatment I in Case E ............................... 43
A.3 Input Parameters of Treatment II in Case E ............................... 44
B.1 Alternative Case Input Parameters of Treatment I & II in Three Comparators 45
B.2 Alternative Case Input Parameters of Treatment III in Three Comparators 45
Acknowledgments

First and foremost, I would like to express my deepest gratitude to my advisor Dr. Qiushi Chen for his guidance and support throughout my master’s study. Dr. Chen provided a tremendous amount of insightful advice on both my research and personal growth. I have been lucky enough to have Dr. Chen as my master’s advisor.

I am extremely grateful to my thesis committee members: Dr. Ethan Xingyuan Fang and Dr. John M. Carroll for generously offering their time, support and guidance throughout the review of this thesis.
Chapter 1
Introduction

Modeling is a useful tool for healthcare decision making. However, model is subject to uncertainty. There are two types of uncertainty. One is the first-order uncertainty which indicates inherent uncertainty from stochastic simulation, and the other is the second-order uncertainty which means the uncertainty in model input parameters. Probabilistic sensitivity analysis (PSA) is commonly used to address the second-order uncertainty, which is a technique used in economic modelling to quantify the level of confidence in the output of the analysis. In PSA, input parameter values are drawn by random sampling from each distribution. Cost and effect outputs of PSA can be presented in the cost-effectiveness acceptability curve (CEAC). This curve plots the probability that the intervention will be cost-effective at a specific willingness-to-pay (WTP) value on the vertical axis against a range of WTP values on the horizontal axis. CEAC transforms the uncertainty in parameters into uncertainty in decisions, which provides economic evaluations for healthcare decision makers [2].

PSA of complex model, like microsimulation and agent-based model, can be extremely time consuming. The nature of the models is the same, complex model is constructed by combination of mathematical functions. Thus, metamodels can be implemented to represent the model uncertainty and simplify the relationship between the inputs and outputs of the complex model. The structure of metamodel is much simpler than the original complex model and this property can reduce the computation burden.

Researchers have used metamodeling approach to increase transparency of simulation models and to summarize the model results in sensitivity/threshold analysis and value of information analysis. Metamodeling approach is promising for approximating model results in cost-effectiveness analysis, but its use in generating CEAC results in PSA has not been thoroughly studied yet. Inspired by previous literature, our objective of this study is to explore the feasibility of using metamodeling approach to approximate CEAC results in PSA of cost-effectiveness analysis. Then we aim to assess the robustness of the metamodeling approach with different study designs.
The remainder of this thesis is organized as follows. Chapter 2 reviews the published literature of several key concepts behind this study. Chapter 3 focuses on overall framework of the analysis with a range of possible variations. Principles of different metamodels and sampling methods are briefly described. In addition, a sick-sicker microsimulation is introduced, which is utilized as a tested microsimulation in this study. Chapter 4 presents the main study findings. Chapter 5 summarizes the main findings, considers limitations and future work, and draws the conclusions of this research.
Chapter 2
Literature Review

This chapter provides an overview of the key concepts behind using metamodeling to approximate cost-effectiveness acceptability curve (CEAC) of cost-effectiveness analysis (CEA). In the following we will review four streams of research that are relevant to our study: CEA, probabilistic sensitivity analysis (PSA), metamodel and microsimulation.

2.1 Cost-Effectiveness Analysis

Cost-effectiveness analysis (CEA) evaluates actions/systems in terms of cost and effectiveness. It assists decision makers to choose one action/system with the preferred cost and effectiveness compared with alternatives.

During 1950’s, CEA has been implemented to make optimum plans of the complex national defense systems [3]. The task and interrelation of CEA is discussed with an example of designing a communication system in a nuclear environment [4]. Researchers explored the application [5,6] and limitations [7] of CEA in military planning.

CEA has been developed for examining problems and estimating yield for given investments in health field since 1960’s [8,9]. It has been applied to assess the treatment of specific diseases such as chronic renal disease [10], femoropopliteal disease [11], Lyme disease [12–14], autoimmune thyroid disease [15,16], gastroesophageal reflux disease [17,18], Parkinson’s disease [19–21], Celiac disease [22,23], sickle cell disease [24–26], metastatic colorectal cancer [27–29] and so on. CEA has a broad effect in health and medicine field [30], Allan and colleagues also introduced a clinician’s guide to CEA [31].

2.2 Probabilistic Sensitivity Analysis and Cost-effectiveness Acceptability Curve

Several factors may cause model uncertainties, including parameter estimation, data extrapolation and model structure. Sensitivity analysis is used to account for the model
uncertainties by evaluating the impact of changing model parameter values on the model outputs. Probabilistic sensitivity analysis (PSA) is a method to account for parameter uncertainty in CEA study [32]. Contrasted with traditional sensitivity analysis, PSA considers uncertainties in all input parameters simultaneously. Input parameter values of PSA are drawn by random sampling from recommended statistical distributions. Further, the impact of input parameter uncertainty on the model output is analyzed [33, 34]. In practice, PSA can estimate the probability of one intervention being cost-effective. PSA is used to handle uncertainty in CEA of choosing between treatment strategies for gastroesophageal reflux disease [18]. Lord and colleagues also extended PSA with the bootstrap procedure to estimate uncertainty ranges for costs [35].

PSA is considered to be impractical for microsimulation (also known as patient-level simulation models) because the model simulates the health care of thousands of individual patients in each single run. In microsimulations, an entire assessment of parameter and decision uncertainty using PSA requires two levels of simulation. One level based on fixed parameters to evaluate a single expected value, the other level is to sample from possible distribution of input parameter values [36]. O’Hagan and colleagues developed a method to reduce the computational burden of Monte Carlo PSA for microsimulation using analysis of variance (ANOVA) [37]. The formulae of determining optimal sample size was deduced, for both standard Monte Carlo method and for the new ANOVA method. Researchers also apply the metamodeling approach to reduce the computation of analyzing PSA of complex model, which is described in the next section.

Cost and effect outputs of PSA can be presented in the cost-effectiveness plane (also called incremental cost-effectiveness plane) as a cloud of points, which represents the differences in costs and health outcomes between interventions. The cost-effectiveness plane plots the incremental effects ($\Delta E$) on the x axis against the incremental costs ($\Delta C$) on the y axis. “Standard care” or called “current practice” is frequently plotted at the origin, so the incremental values represents the difference between new intervention versus (minus) current practice.

Incremental cost-effectiveness ratio (ICER) and willingness-to-pay (WTP) are used to determine whether an intervention is cost-effective or not. ICER is a ratio of incremental cost to incremental effect ($\Delta C/\Delta E$), which represents the slope of the line joining any point on the cost-effectiveness plane to the origin. WTP is a range of monetary values that the societal is willing to pay for one unit gain in effect outcome, a given WTP value is denoted as $\lambda$. The cost-effectiveness plane is divided into four quadrants by the origin (Figure 2.1). Northwest quadrant illustrates the position that new intervention is less
effective and more costly than the comparator. Thus, new intervention in Northwest quadrant is never cost-effective regardless of $\lambda$ (new treatment dominated). In Southeast quadrant, new intervention generates higher effect outcomes with lower cost compared with the comparator. Intervention in Southeast quadrant is always cost-effective regardless of $\lambda$ (new treatment dominates). In Northeast quadrant of cost-effectiveness plane, a new intervention is cost-effective if ICER is greater than $\lambda$. In contrast, a new intervention is cost-effective if ICER is less than $\lambda$ when falling in Southwest quadrant.

![Cost-effectiveness plane for new treatment compared with current practice.](image)

**Figure 2.1.** Cost-effectiveness plane for new treatment compared with current practice.

However, producing confidence intervals around ICERs can be statistically challenging [38,39]. Cost-effectiveness acceptability curve (CEAC) is introduced as an alternative and is considered as a more general solution to present uncertainty in cost-effectiveness analysis. CEAC is derived from the joint distribution of incremental costs and incremental effects. CEAC plots the probability of one intervention being cost-effective under a range of $\lambda$. This probability can be identifiable from the cost-effectiveness plane as the proportion of points falling to the south and east of a line through the origin with slope equal to $\lambda$ [40].

An alternative approach to construct CEAC is the net-benefit framework. Incremental net monetary benefit (INMB) [41] transforms health outcomes into monetary values, which can evaluate whether a new intervention is cost-effective or not. A new intervention is cost-effective under $\lambda$ with a positive INMB in Equation 2.1. Algorithm 1 presents the procedure to plot CEAC.
\[ \text{INMB} = \lambda \cdot \Delta E - \Delta C \] (2.1)

**Algorithm 1** Cost-Effectiveness Acceptability Curve

- **M**: the total number of observations
- **K**: the total number of parameters
- **i**: \( i \)th parameter, \( i \in 1, 2, \ldots K \)
- **j**: \( j \)th observation in parameter, \( j \in 1, 2, \ldots M \)
- **\( x_{ij} \)**: \( j \)th observation of \( i \)th parameter
- **(\( \Delta C, \Delta E \))**: pairs of incremental cost and incremental effect between standard care and new intervention from PSA outputs
- **\( \lambda \)**: a given willingness-to-pay (WTP) value, ranging from 0 to WTP
- **\( n \)**: number of new intervention being cost-effective given \( \lambda \)
- **p**: probability of new intervention being cost-effective given \( \lambda \)
- **N**: the total number of \( (\Delta C, \Delta E) \) pairs
- **l**: a specific \( (\Delta C, \Delta E) \) pair from group of \( (\Delta C, \Delta E) \) pairs

**procedure** PLOT CEAC(M, K, WTP)

1. set seed
2. choose distribution for each parameter
3. **for** \( j = 1 \) to \( M \) **do**
   - \( (x_{1j}, x_{2j}, \ldots x_{ij}, \ldots x_{Kj}) \leftarrow \) sample from corresponding distribution
   - run simulation with \( (x_{1j}, x_{2j}, \ldots x_{ij}, \ldots x_{Kj}) \)
4. **end for**
5. return \( (\Delta C, \Delta E) \)
6. **for** \( \lambda = 0 \) to WTP **do**
7.   - \( n \leftarrow 0 \)
8.   - \( p \leftarrow 0 \)
9.   **for** \( l = 1 \) to \( N \) **do**
10.      - if \( \text{INMB} > 0 \) **then**
11.         - \( n \leftarrow n + 1 \)
12.      **end if**
13.   **end for**
14.   \( p = \frac{n}{N} \)
15.   plot \( p \) on vertical axis against WTP on horizontal axis
16. **end for**
17. return CEAC
18. **end procedure**
2.3 Metamodel

Some complex models, like microsimulations and agent-based models can be time-consuming. Metamodel is a method to reduce the computation burden in this complicated models. Metamodel is a simplified model of an existing model, which represents the relations between model input and output.

In theory, metamodel simplifies the simulation optimization in two ways, including generating deterministic response and shortening the run times [42]. Combined with the leave-k-out cross validation strategy, best metamodel is more effective for creating inexpensive approximations of computationally expensive simulations among different types of metamodels [43].

Metamodel has been broadly used in research of healthcare services. In area of healthcare monitoring and decision making, researchers implement a metamodel for a care process monitoring application (CPMA), and evaluate it for addressing community care processes [44]. A Care Process Metamodel is presented for business intelligence healthcare monitoring solutions [45]. Burke et al. discussed the public health metamodeling pathway to support good decision making for preparedness [46]. Daniels and colleagues presented a metamodel for evaluating counseling programs and identifying the appropriate selection of evaluation procedures [47]. Mark and colleagues developed a metamodel for quantifying health damages of power grid expansion plans and enabling decision-makers to assess the plan with high certainty [48]. A metamodel-based methodology is proposed to address organization problems of healthcare systems with process view, resource view and organization view [49].

In terms of modeling Hospital Information Systems (HIS), researchers build a metamodel to overcome the HIS complexity and supervise the HIS [50]. A three-layer graph-based metamodel (3LGM) is introduced to support the systematic management of HIS and to assess the quality of information processing in hospitals [51]. The Clinical Adoption Meta-Model (CAMM) is developed to describe the clinical adoption of multiple health information systems from health records to hospital information systems [52]. In addition, metamodels are trained to estimate the hospital capacity and describe the response of emergency department [53]. For example, studying the patient waiting time [54–56], and evaluating the resource planning [57,58].

Different types of metamodels have been found useful in cost-effectiveness analysis models from previous research. Linear regression metamodeling can summarize and present results in threshold and sensitivity analysis [59], and can compute expected
value of partial sample information from PSA [60]. Generalized Additive Model (GAM) metamodelling and Gaussian process metamodelling can estimate multiparameter partial expected value of perfect information (EVPI) from PSA samples [61]. Multiple linear regression and Gaussian process metamodelling significantly reduces the time of computing overall EVPI and partial EVPI in Bayesian value of information analysis [62].

2.4 Microsimulation

In 1957, microsimulation models were proposed by Guy Orcutt for socio-economic system to formulate and test the behavior of the real-world decision-making units, such as individuals, families and firms [63]. The microsimulation model has been developed for consumer marketing to describe behavior in certain types of market and to suggest changes in marketing strategy [64]. In transportation research area, microsimulation models are discussed to organise car sharing schemes [65], to simulate traffic flow and model transportation networks [66,67].

Microsimulation models for epidemiology support the decision for controlling sexually transmitted diseases (STDs) by describing the mechanisms at individual level [68]. Vlas and colleagues used microsimulation models for the epidemiology and control of schistosomiasis by simulating surveys and interventions in Burundi [69]. Habbema and colleagues applied the microsimulation models to epidemiological modeling of helminthic infections [70]. Population Health Microsimulation Model (POHEM) is implemented to estimate and predict present and future epidemiological and economic impacts of multiple sclerosis (MS) in Canada [71]. Microsimulation models for the transmission of HIV assess the treatment [72–75] and investigate the disease spread [76–78]. Microsimulation models are also used to quantify the future health and economic burden of osteoarthritis [79], to evaluate the treatments in osteoporosis [80], to estimate the contribution of risk factor trends to Canada cardiovascular disease [81] and US intestinal-type noncardia gastric adenocarcinoma (NCGA) incidence [82].

Microsimulation models have significant effects in health policy and health economics area. During the 1970’s, the microsimulation models have been performed to study U.S. health manpower policies [83], the role of the hospital [84] and the role of the physician services sector in the health care system [85]. Microsimulation models are applied to examine the equilibrium of medical savings accounts combined with high-deductible catastrophic health plans (MSA/CHPs) on both health care and non-health care expenditures [86]. Researchers use microsimulation models to estimate the effects of tax subsidies for health insurance [87,88], to predict the employees’ health plan choices
and track the progress [89], and to assess the prospect of state health insurance action [90].

Microsimulation Screening Analysis (MISCAN) is introduced to evaluate the effect of cancer screening on the individual and population level [91]. MISCAN has been developed to assess the screening of colorectal cancer [92,93], breast cancer [94,95], lung cancer [96], cervical cancer [97] and prostate Cancer [98].

According to [99], the following microsimulation models contribute to investigating health reform and healthcare demand: State Health Policy Microsimulation [100], Transfer Income Model (TRIM3) [101], RAND’s Comprehensive Assessment of Reform Efforts (COMPARE) [102], Health Benefits Simulation Model (HBSM) [103], the Health Insurance Reform Simulation Model (HIRSM) [104], Medical Expenditure Microsimulation Model (MEDSIM) [105], Australian Population and Policy Simulation Model (APPSIM) [106], LifePaths [107], Congressional Budget Office Long-Term (CBOLT) [108], the Cornell Dynamic Population Microsimulation Model (CORSIM) [109], the Pension and Retirement Income Simulation Model (PRISM) [110] and Dynamic Simulation of Income Model (DYNASIM) [111].
Chapter 3
Study Design

The aim of this chapter is to demonstrate the overall framework of our study designs in assessing the robustness of the metamodeling approach. The chapter starts by describing the overall framework and the performance measure of the metamodeling approach. The next section illustrates the study designs of testing whether the performance of the metamodeling approach will be affected by different variations. The performance of the metamodeling approach is examined with different numbers and shapes of comparators, different choices of metamodels, different training sizes and different sampling methods for training set.

3.1 Metamodeling Framework

The schematic representation of overall framework is shown in Figure 3.1. The model parameters are sampled from recommended statistical distribution [36]. Beta distribution is used for generating probability and utility input, and gamma distribution is employed for sampling cost input. In this study, model outputs are cost and effect for each arm, where the effect is measured by quality-adjusted life-year. Metamodels are built for incremental cost and incremental effect separately.

Standardization, also called a z-score, is applied to scale the input parameters but not the responses when using probabilistic sensitivity analysis (PSA) and Latin hypercube sampling (LHS) samples to train metamodels. Let $X$ denotes the $k$ input factors and $Y$ denotes the quantitative response. The $i^{th}$ input factor can be standardized with Equation 3.1. Since there is no variance in samples from full factorial design, the input parameters are scaled with normalization (Equation 3.2) when training the metamodels.

$$Z_i = \frac{X_i - \bar{X}_i}{\sigma_i} \quad (3.1)$$

where $\bar{X}_i$ and $\sigma_i$ corresponds to the mean and standard deviation of the $i^{th}$ input factor.
\[ Z_i = \frac{X_i - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \]  

(3.2)

The true CEAC can be obtained by running the simulation models with 10,000 independently sampled parameter sets, which is considered as gold standard. Similarly, the approximated CEAC can be generated from testing sets. The difference between the true CEAC and the approximated CEACs is not the same over different WTP values, which could be positive or negative, and small or large. Thus, the approximation error is defined as Equation 3.3. It means the maximum absolute value of the difference between true CEAC and the approximated CEACs. In the analysis with three or more comparators, the difference between true CEAC and approximated CEACs over the entire range of WTP is consisted of the maximum absolute difference between true CEAC and approximated CEACs at each WTP value. The analysis is repeated for 20 times to account for sampling randomness, which results in twenty approximation error values. We compute the performance measure of a specific metamodeling approach as the mean value, with 95% confidence interval of the sample mean to account for the range of approximation errors. Figure 3.2 is the schematic diagram of difference between true CEAC and approximated CEACs.

\[ \text{approximation error} = \max \{ |\text{true CEAC} - \text{approximated CEAC}| \} \]  

(3.3)

Figure 3.1. Overall Framework
3.2 Different Numbers of Comparators

To assess the robustness of the metamodeling approach, two comparators with different CEAC shapes are tested. Referring to Case A, Case E and Case H from [112], three shapes of two comparators are considered.

CEAC of Case A is considered as the standard case because it is representative. CEAC of Case C, Case G and Case F are the alternatives of Case A, Case E and Case H respectively. The probability of being cost-effective of the former cases and that of latter cases equal to one correspondingly. Scenarios of Case A, Case E and Case H are described as follow:

- CEAC of Case A (Figure 3.3) illustrates the scenario that the probability of being cost-effective is 0 under small WTP value. With the increase of WTP, the probability of being cost-effective increases and gradually approaches 1.

- CEAC of Case E (Figure 3.5) demonstrates the scenario that the probability of being cost-effective is around 0.5 at the beginning and is approximate to 1 with the increase of WTP.

- CEAC of Case H (Figure 3.4) shows the scenario that the probability of being cost-effective is 0 when WTP is small and grows slowly with the increase of WTP.
**Figure 3.3.** Cost-effectiveness acceptability curves of Case A and Case C

**Figure 3.4.** Cost-effectiveness acceptability curves of Case H and Case F
In addition, three comparators with two different shapes of CEAC are evaluated. CEACs of base case of three comparators are shown in Figure 3.6, and CEACs of alternative case are plotted in Figure 3.7.

Figure 3.5. Cost-effectiveness acceptability curves of Case E and Case G

Figure 3.6. Cost-effectiveness acceptability curves of base case of three comparators
3.3 Types of Metamodels

Five types of metamodels are performed to study whether the metamodeling approach depends on choice of metamodels. In addition, different training sizes which vary from 5000 to 200, are conducted to test the performance of metamodels. Assume that there is some relationship between $Y$ and $X = (X_1, X_2, ..., X_k)$, which can be represented in a general form

$$Y = f(X)$$  \hspace{1cm} (3.4)

where $f$ is a fixed but unknown function, $Y$ is a function of the factors $X$. Given a known set of inputs $X$, the metamodeling approach estimates $f$ by predicting $Y$ using

$$\hat{Y} = \hat{f}(X) + \varepsilon$$  \hspace{1cm} (3.5)

where $\varepsilon$ is error term, which is a random variable assumed to be normally distributed with a mean of 0 and a variance of $\sigma^2$. $\hat{f}$ is an estimation of $f$ and $\hat{Y}$ is the prediction result for $Y$. Different choices of metamodels represent different $\hat{f}$ functions.
3.3.1 Linear Model (LM)

Linear model (hereinafter referred to as LM) defines a model output as a linear function of model inputs, which assumes \( \hat{f} \) is a linear function. In this study, linear model indicates the first order model (linear) without interaction, that is

\[
Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k + \varepsilon
\]

which can be written as

\[
Y = \beta_0 + \sum_{j=1}^k \beta_j X_j + \varepsilon
\]

where \( \beta_0, \beta_1, \beta_2, \ldots, \) and \( \beta_k \) are unknown regression coefficients and \( \beta_0 \) is the intercept. These regression coefficients can be calculated by using \texttt{lm} function from stats package in R. The following descriptions and equations briefly explain the principle. Let \( y_i \) denotes the \( i \)th observation of response \( Y \), \( x_{ij} \) denotes the \( i \)th observation of input factor \( X_j \). The \( i \)th observation of \( Y \) can be shown as

\[
y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_k x_{ik} + \varepsilon_i
\]

which is

\[
y_i = \beta_0 + \sum_{j=1}^k \beta_j x_{ij} + \varepsilon_i \quad i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, k
\]

where \( \varepsilon_i \) is the error term.

\[
\varepsilon_i = y_i - \beta_0 - \sum_{j=1}^k \beta_j x_{ij}
\]

Replace \( \varepsilon_i \) with its estimator \( e_i \), \( e_i = y_i - \hat{y}_i \) represents the \( i \)th residual. The residual sum of squares (RSS) is defined as

\[
RSS = e_1^2 + e_2^2 + \cdots + e_n^2 = \sum_{i=1}^n e_i^2
\]

or

\[
RSS = \sum_{i=1}^n \left( y_i - \beta_0 - \sum_{j=1}^k \beta_j x_{ij} \right)^2
\]

The least squares approach is commonly used to estimate the coefficients by minimizing the RSS. Referring to [59], raw regression coefficients are sensitive to scale. This characteristic could make the coefficients more difficult to interpret in relation to each
other. Thus, model input parameters are recommended to be scaled to break the limitation with standardization.

### 3.3.2 Linear Model with Two-Factor Interaction Terms (LM2)

Linear model with two-factor interaction terms (hereinafter referred to as LM2) introduces nonlinearity in metamodel. Main effects and two-way interaction effects are included to estimate \( f \), that is

\[
Y = \beta_0 + \sum_{j=1}^{k} \beta_j X_j + \sum_{m=1}^{k-1} \sum_{j=1}^{k} \beta_{mj} X_m X_j + \sum_{j=1}^{k} \beta_{jj} X_j^2 + \varepsilon, \quad m \neq j \tag{3.13}
\]

Referring to [59], the input parameters are standardized to remedy the scale-sensitive problem.

### 3.3.3 LASSO Regression with Two-Way Interaction Terms

LASSO is a shrinkage method that regularizes the coefficient estimates when fitting a model of all \( n \) input parameters. By introducing a \( \ell_1 \) penalty term, the LASSO coefficients minimize the quantity

\[
\sum_{i=1}^{n} \left( y_i - \beta_0 - \sum_{j=1}^{k} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{k} |\beta_j| = \text{RSS} + \lambda \sum_{j=1}^{k} |\beta_j| \tag{3.14}
\]

LASSO can perform variable selection by shrinking the coefficient estimates towards zero. We use LASSO regression with two-way interaction terms to capture the nonlinearity in later analysis.

### 3.3.4 Generalized Additive Model (GAM)

Generalized Additive Model (GAM) extends a standard linear model by allowing non-linear functions of each of the input variables, while maintaining additivity [113]. In GAM, each linear component \( \beta_j x_{ij} \) in Equation 3.9 is replaced by a smooth nonlinear function \( f_j(x_{ij}) \) to obtain the nonlinearity between each variable and the response. GAM for regression problems can be written as

\[
y_i = \beta_0 + f_1(x_{i1}) + f_2(x_{i2}) + \cdots + f_p(x_{ip}) + \varepsilon_i
\]

\[
= \beta_0 + \sum_{j=1}^{p} f_j(x_{ij}) + \varepsilon_i \tag{3.15}
\]
In this study, $f_j(x_{ij})$ is represented using smoothing splines. When fitting a curve to a set of data, RSS is one of calibrations that measure the fitness. However, it may cause overfitting if function $f$ is chosen by interpolating all of the response data $Y$. Such function $f$ lacks flexibility, which fails to fit additional input data $X$. Ideally, function $f$ can not only make RSS small, but it is also smooth. By introducing a penalty term to the loss function, a smoothing spline minimizes

$$
\sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \int g''(t)^2 dt
$$

(3.16)

where the term $\lambda \int g''(t)^2 dt$ penalizes the variability in function $f$ [113].

A cubic spline is a piecewise polynomial with three constraints. First, it is constrained to be continuous. Second, its first derivative is continuous. Third, its second derivative is continuous. Each constraint imposed will release one degree of freedom. In general, a cubic spline with $H$ knots uses a total of $H + 4$ degrees of freedom.

### 3.3.5 Neural Networks

A neural network is a network of neurons. A simple feedforward artificial neural network is composed of input layer, hidden layer, output layer and neurons in each layer. The structure of a simple neural network is shown in Figure 3.8. Each circle represents a neuron. The connections between neurons are modeled as weights. All inputs are modified by a weight and summed, which is referred as a linear combination.

![Figure 3.8. Schematic representation of a simple neural network.](image-url)
3.4 Sampling Methods for Training Sets

Three sampling methods are applied to generate training sets, including probabilistic sensitivity analysis (PSA), Latin hypercube sampling (LHS), and full factorial design. In PSA, input parameter values are drawn by random sampling from each distribution. LHS returns uniformly distributed input parameter values. Full factorial design generates the deterministic input value, with lower bound and upper bound of each parameter.

3.4.1 Probabilistic Sensitivity Analysis (PSA)

Input parameter values of PSA are drawn by random sampling from recommended statistical distributions [36]. When selecting distributions from parameters, normal distribution is considered as an effective candidate for any parameter based on the central limit theorem. Regardless of the distribution of data, the theorem states that the sample means will be approximately normally distributed given sufficient samples. Since the fundamental interest is about expectations (mean values), normal distribution is potential for reflecting uncertainty in any input parameters. Except for normal distribution, beta distribution is recommended for binomial data and beta distribution is suitable for multinomial data when estimating probabilities. In addition, beta distribution, log-normal distribution can be applied for utilities and relative risk parameters respectively. Cost value can be sampled from gamma distribution or log-normal distribution.

3.4.2 Full Factorial Design

In full factorial design, all input factors are set at two level individually, which is called “high” and “low” level or “+1” and “-1” level. Full factorial design contains all high and low level combinations of all the input factors. If the number of factor is $k$, a full factorial design has $2^k$ runs. With sampling lower bound and upper bound of each input factor, full factorial design generates the deterministic input value for microsimulation. Since the utility and cost input factors are sampled from recommended statistical distributions, the lower bound and upper bound are considered as 5th and 95th percentile of the distribution respectively. There is no information between extreme values (lower bound and upper bound). Thus, the input parameters are scaled with normalization (Equation 3.2) when training the metamodels.

The process of full factorial design is as follow: First, calculate lower bound and upper bound of $k$ input factors. Second, design $2^k$ full factorial design with all high/low
combinations. Third, perform the microsimulation $2^k$ runs with one combination each time.

### 3.4.3 Latin Hypercube Sampling (LHS)

LHS was described by Michael McKay of Los Alamos National Laboratory in 1979 [114]. It is a method of generating random samples of parameter values from a multidimensional distribution. LHS is based on the Latin square design, where only one sample is in each row and each column. In one-dimensional LHS, cumulative density function (CDF) is evenly divided into $N$ partitions and one sample is randomly selected in each partition. For two-dimensional LHS, the two variables must be independent. One-dimensional LHS samples are generated for these two variables separately and then are randomly combined into two-dimensional pairs. In two-dimensional LHS, only one sample is in each row and each column, and sampling is random in each grid [115]. It can be extended into higher-dimensional LHS with the same process. When performing LHS, the number of sample points and parameters must be specified in advance.

For each parameter, LHS returns a random sampling point in each evenly partitioned CDF region. Thus, the CDF values need to be transformed into actual values of parameters with inverse normalization (Equation 3.17). Assuming that 5th and 95th percentile of the distribution are lower bound and upper bound individually.

$$Z_i = X_i \times (X_{max} - X_{min}) + X_{min}$$ (3.17)

The process of LHS is as follow: First, calculate lower bound and upper bound of each parameter. Second, specify the number of sample points and parameters and construct LHS. Third, compute actual values with inverse normalization.

### 3.5 Microsimulation Models

#### 3.5.1 Method Overview

Microsimulation model is a modeling technique that operates at the level of individual units, such as persons. Each individual has unique identifier and associated attributes. We use transition probabilities to these individuals to represent simulated changes in state and behavior [116]. Microsimulation model is consisted of mutually exclusive and collectively exhaustive states. Each individual is simulated through the model one at a time and can only be in one state at any given cycle [117]. In a cost-effectiveness analysis,
each state and each possible transition will result in a specific cost and effectiveness outcome. The result estimates cost and effectiveness outcomes of applying the transition probabilities over certain cycles, including both means and distributions.

Microsimulation model is not constrained by the Markov assumption (memoryless). Both state and transition outcomes depend on the individual’s characteristics, past transitions and events of the individual. Thus, it can keep track of each individual’s history. According to [118], the steps to simulate one hypothetical individual over time are as follow:

1. The individual is assigned to an initial state with cost and effectiveness of staying in the initial state for one cycle.
2. In each cycle, the transition probability of the next cycle (transiting to a different state or staying in the current state) is assigned based on the individual characteristics and history of past states.
3. The health state in the next cycle, where the individual will transit to, is sampled from a categorical distribution based on the transition probability to each possible health state.
4. Each state is associated with particular cost and effectiveness outcomes of being in this state for one cycle length. State-specific cost and effectiveness outcomes are based on the individual characteristics and history of past states.
5. The result can estimate the total (discounted) cost and effectiveness outcomes for the individual in certain cycles by aggregating all the state and transition values over the cycles, applying discounted if needed.

3.5.2 The Sick-Sicker Microsimulation Model

A sick-sicker microsimulation model from a previous publication is used to test the metamodeling approach [119]. Its schematic representation is illustrated in Figure 3.9. Four states are included in the sick-sicker model: healthy (H), sick (S1), sicker (S2) and dead (D). All individuals are assumed to be in H state in the starting point. Healthy individuals are possible to stay healthy or develop a disease and progress to S1 in next transit. Individuals in S1 can recover (return to H), stay in S1, or progress to S2. Individuals in S2 are not able to recover, which indicates that they cannot transit to H or S1. D is an absorbing state. Individuals from H, S1 and S2 can reach D. Mortality rate
is constant for individuals in H and will increase for those in S1 and S2. All individuals
are tracked for 30 years, with cycle length of 1 year. This cycle length is constant over
the time horizon of this model.

Figure 3.9. Schematic representation of the sick-sicker microsimulation model

3.5.3 Model Input

Two comparators are evaluated, which are standard care (without treatment) and new
intervention (with treatment) strategies. The treatment can be performed for individuals
in S1 and S2. If individuals are accepting treatment, cost of treatment will be generated
in addition to the cost of being in S1 or S2. The treatment increases the effectiveness for
individuals in S1, however has no effect for those in S2.

The total number of simulated individuals is 10,000. The probability to die when
healthy is assumed to be constant as 0.005. The rate ratios of death in sick (rr.S1) and
in sicker (rr.S2) compared with healthy are 3 and 10 respectively. Multiplying rr.S1 by
the rate of death in healthy (r.HD), the rate of death in sick (r.S1D) is achieved. Further,
converting rate into probability to die while in sick (p.S1D). Similarly, the probability of
die while in sicker (p.S2D) can be calculated.

According to microsimulation and outcome functions in [118], the cost and effect
functions can determine whether an individual receives treatment or not by using binary
scalar. If a treatment is used, cost and effectiveness will increase in relevant states.

Input parameters are modified to generate different CEACs of two comparators.
Modified input parameters of Case A are listed in Table 3.1. The modified input
parameter values of Case H and Case E can be found in Appendix A. Treatment I is
standard care (without treatment) by default and treatment II is new intervention (with
treatment). For Case A and Case H, input parameters are sampled from one set of distributions respectively by using “0” representing “without treatment” case and “1” representing “with treatment” case in microsimulation and outcome functions. However, due to the special CEAC shape of Case H, input parameters are sampled from two sets of distributions. That is, treatment I and treatment II are both “with treatment” strategies with corresponding set of distributions in Case H.

Table 3.1. Input Parameters for Case A

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Distribution</th>
<th>Mean</th>
<th>SD</th>
<th>LB</th>
<th>UB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Mean</td>
<td>SD</td>
<td>LB</td>
<td>UB</td>
<td></td>
</tr>
<tr>
<td>Annual Transition Probabilities</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( p_{HS1} )</td>
<td>Disease onset (H to S1)</td>
<td>beta((21.1,119.56))</td>
<td>0.150</td>
<td>0.030</td>
<td>0.104</td>
<td>0.202</td>
</tr>
<tr>
<td>( p_{S1H} )</td>
<td>Recovery (S1 to H)</td>
<td>beta((12,12))</td>
<td>0.500</td>
<td>0.100</td>
<td>0.335</td>
<td>0.665</td>
</tr>
<tr>
<td>( p_{S1S2} )</td>
<td>Disease progression (S1 to S2)</td>
<td>beta((22.27,189.83))</td>
<td>0.105</td>
<td>0.021</td>
<td>0.073</td>
<td>0.142</td>
</tr>
<tr>
<td>Annual Costs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( c_{H} )</td>
<td>Healthy individuals</td>
<td>gamma((25,80))</td>
<td>2000</td>
<td>400</td>
<td>1390</td>
<td>2700</td>
</tr>
<tr>
<td>( c_{S1} )</td>
<td>Sick individuals in S1</td>
<td>gamma((25,160))</td>
<td>4000</td>
<td>800</td>
<td>2781</td>
<td>5400</td>
</tr>
<tr>
<td>( c_{S2} )</td>
<td>Sick individuals in S2</td>
<td>gamma((25,600))</td>
<td>15000</td>
<td>3000</td>
<td>10429</td>
<td>20251</td>
</tr>
<tr>
<td>( c_{Trt} )</td>
<td>Annual treatment cost per sick individual (S1 and S2)</td>
<td>gamma((25,400))</td>
<td>10000</td>
<td>2000</td>
<td>6952</td>
<td>13500</td>
</tr>
<tr>
<td>Utility weights</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( u_{S1} )</td>
<td>Sick individuals in S1</td>
<td>beta((24.25,50))</td>
<td>0.327</td>
<td>0.054</td>
<td>0.240</td>
<td>0.418</td>
</tr>
<tr>
<td>( u_{S2} )</td>
<td>Sick individuals in S2</td>
<td>beta((49.5,148.5))</td>
<td>0.250</td>
<td>0.031</td>
<td>0.201</td>
<td>0.302</td>
</tr>
<tr>
<td>Intervention effect</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( u_{Trt} )</td>
<td>Utility for treated individuals in S1</td>
<td>beta((14,15,5))</td>
<td>0.740</td>
<td>0.098</td>
<td>0.564</td>
<td>0.885</td>
</tr>
</tbody>
</table>

SD, standard deviation; LB, lower bound; UB, upper bound.

In addition, two cases of three comparators are examined, which are denoted as base case and alternative case. Three comparators are analyzed by setting treatment I as standard care (without treatment), treatment II and treatment III as new interventions (with treatment). Among them, treatment I and treatment II are sharing the same set of distributions. The binary scalar is used to distinguish them in microsimulation and outcome functions. The other set of distributions is assigned to treatment III. The base case input parameters of three comparators are shown in Table 3.2 and Table 3.3. The modified input parameter values to generate alternative case of three comparators can be found in Appendix B.
### Table 3.2. Base Case Input Parameters of Treatment I & II in Three Comparators

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Distribution</th>
<th>Mean</th>
<th>SD</th>
<th>LB</th>
<th>UB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Annual Transition Probabilities</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p.HS1</td>
<td>Disease onset (H to S1)</td>
<td>beta(21.1,119.56)</td>
<td>0.150</td>
<td>0.030</td>
<td>0.104</td>
<td>0.202</td>
</tr>
<tr>
<td>p.S1H</td>
<td>Recovery (S1 to H)</td>
<td>beta(12.12)</td>
<td>0.500</td>
<td>0.100</td>
<td>0.335</td>
<td>0.665</td>
</tr>
<tr>
<td>p.S1S2</td>
<td>Disease progression (S1 to S2)</td>
<td>beta(22.27,189.83)</td>
<td>0.105</td>
<td>0.021</td>
<td>0.073</td>
<td>0.142</td>
</tr>
<tr>
<td></td>
<td>Annual Costs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c.H</td>
<td>Healthy individuals</td>
<td>gamma(25,80)</td>
<td>2000</td>
<td>400</td>
<td>1390</td>
<td>2760</td>
</tr>
<tr>
<td>c.S1</td>
<td>Sick individuals in S1</td>
<td>gamma(25,160)</td>
<td>4000</td>
<td>800</td>
<td>2781</td>
<td>5400</td>
</tr>
<tr>
<td>c.S2</td>
<td>Sick individuals in S2</td>
<td>gamma(25,600)</td>
<td>15000</td>
<td>3000</td>
<td>10429</td>
<td>20251</td>
</tr>
<tr>
<td>c.Trt</td>
<td>Annual treatment cost per sick individual (S1 and S2)</td>
<td>gamma(25,400)</td>
<td>10000</td>
<td>2000</td>
<td>6952</td>
<td>13500</td>
</tr>
<tr>
<td></td>
<td>Utility weights</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>u.S1</td>
<td>Sick individuals in S1</td>
<td>beta(24.25,50)</td>
<td>0.327</td>
<td>0.054</td>
<td>0.240</td>
<td>0.418</td>
</tr>
<tr>
<td>u.S2</td>
<td>Sick individuals in S2</td>
<td>beta(49.5,148.5)</td>
<td>0.250</td>
<td>0.031</td>
<td>0.201</td>
<td>0.302</td>
</tr>
<tr>
<td></td>
<td>Intervention effect</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>u.Trt</td>
<td>Utility for treated individuals in S1</td>
<td>beta(10.5,3.3)</td>
<td>0.761</td>
<td>0.111</td>
<td>0.558</td>
<td>0.919</td>
</tr>
</tbody>
</table>

### Table 3.3. Base Case Input Parameters of Treatment III in Three Comparators

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Distribution</th>
<th>Mean</th>
<th>SD</th>
<th>LB</th>
<th>UB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Annual Transition Probabilities</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p.HS1</td>
<td>Disease onset (H to S1)</td>
<td>beta(21.1,119.56)</td>
<td>0.150</td>
<td>0.030</td>
<td>0.104</td>
<td>0.202</td>
</tr>
<tr>
<td>p.S1H</td>
<td>Recovery (S1 to H)</td>
<td>beta(12.12)</td>
<td>0.500</td>
<td>0.100</td>
<td>0.335</td>
<td>0.665</td>
</tr>
<tr>
<td>p.S1S2</td>
<td>Disease progression (S1 to S2)</td>
<td>beta(22.27,189.83)</td>
<td>0.105</td>
<td>0.021</td>
<td>0.073</td>
<td>0.142</td>
</tr>
<tr>
<td></td>
<td>Annual Costs</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c.H</td>
<td>Healthy individuals</td>
<td>gamma(25,80)</td>
<td>2000</td>
<td>400</td>
<td>1390</td>
<td>2760</td>
</tr>
<tr>
<td>c.S1</td>
<td>Sick individuals in S1</td>
<td>gamma(25,160)</td>
<td>4000</td>
<td>800</td>
<td>2781</td>
<td>5400</td>
</tr>
<tr>
<td>c.S2</td>
<td>Sick individuals in S2</td>
<td>gamma(25,600)</td>
<td>15000</td>
<td>3000</td>
<td>10429</td>
<td>20251</td>
</tr>
<tr>
<td>c.Trt</td>
<td>Annual treatment cost per sick individual (S1 and S2)</td>
<td>gamma(25,480)</td>
<td>12000</td>
<td>2400</td>
<td>8343</td>
<td>16201</td>
</tr>
<tr>
<td></td>
<td>Utility weights</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>u.S1</td>
<td>Sick individuals in S1</td>
<td>beta(24.25,50)</td>
<td>0.327</td>
<td>0.054</td>
<td>0.240</td>
<td>0.418</td>
</tr>
<tr>
<td>u.S2</td>
<td>Sick individuals in S2</td>
<td>beta(49.5,148.5)</td>
<td>0.250</td>
<td>0.031</td>
<td>0.201</td>
<td>0.302</td>
</tr>
<tr>
<td></td>
<td>Intervention effect</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>u.Trt</td>
<td>Utility for treated individuals in S1</td>
<td>beta(8.2)</td>
<td>0.800</td>
<td>0.121</td>
<td>0.571</td>
<td>0.959</td>
</tr>
</tbody>
</table>
Chapter 4
Computational Study

The main study findings are discussed in this chapter. First, the base case results of two comparators are demonstrated, where different types of metamodels are compared. The approximation errors of a metamodel will increase slightly when using fewer training samples. Introducing nonlinearity in metamodel can reduce the approximation error. Second, the results are extended to the base case of three comparators, where the metamodeling approach shows similar performance of approximating cost-effectiveness acceptability curves (CEACs). Third, effects of different shapes of CEACs and different sampling schemes are illustrated. Last but not least, we test the metamodeling approach on real-world research data from lymphoma treatment and show the results.

4.1 Base Case Results of Two Comparators

First, the metamodeling approach is analyzed on the base case of two comparators (Case A), which is the most common scenario in cost-effectiveness analysis. Metamodels of LM, LM2, LASSO, GAMs and neural networks are trained on 5000, 2000, 1000, 500 and 200 samples individually. The metamodeling approach performs well and the approximated CEACs are close to the true CEAC. The base case approximation errors of two comparators are shown in Table 4.1.

<table>
<thead>
<tr>
<th>Training Size</th>
<th>LM</th>
<th>LM2</th>
<th>LASSO</th>
<th>GAMs</th>
<th>Neural Networks</th>
</tr>
</thead>
<tbody>
<tr>
<td>5000</td>
<td>0.91 (0.89-0.93) \textsuperscript{a}</td>
<td>0.26 (0.25-0.27)</td>
<td>0.28 (0.27-0.29)</td>
<td>0.98 (0.95-1.00)</td>
<td>0.78 (0.66-0.91)</td>
</tr>
<tr>
<td>2000</td>
<td>0.92 (0.89-0.95)</td>
<td>0.27 (0.26-0.28)</td>
<td>0.28 (0.27-0.29)</td>
<td>1.00 (0.97-1.03)</td>
<td>0.89 (0.73-1.04)</td>
</tr>
<tr>
<td>1000</td>
<td>1.02 (0.94-1.09)</td>
<td>0.26 (0.25-0.27)</td>
<td>0.31 (0.29-0.34)</td>
<td>1.09 (1.02-1.17)</td>
<td>0.99 (0.81-1.17)</td>
</tr>
<tr>
<td>500</td>
<td>1.09 (0.99-1.20)</td>
<td>0.27 (0.26-0.29)</td>
<td>0.33 (0.30-0.35)</td>
<td>1.18 (1.07-1.29)</td>
<td>1.27 (1.01-1.54)</td>
</tr>
<tr>
<td>200</td>
<td>1.33 (1.19-1.46)</td>
<td>0.32 (0.29-0.34)</td>
<td>0.42 (0.38-0.47)</td>
<td>1.38 (1.26-1.50)</td>
<td>1.49 (1.25-1.72)</td>
</tr>
</tbody>
</table>

\textsuperscript{a} 95\% confidence interval.
4.1.1 Linear Model (LM)

LM metamodels are applied to approximate CEAC in the base case analysis with two comparators. The metamodeling approach is effective. The approximation error is 0.91% (95% Confidence Interval: 0.89%-0.93%) when using 5000 training samples. The true CEAC and approximated CEACs are shown in Figure 4.1(a), with the black line representing the true CEAC and the curves with light color representing the approximated CEACs.

The metamodeling approach could closely reproduce the true CEAC. Differences between true CEAC and approximated CEACs are plotted to distinguish the nuances. In Figure 4.1(b), the maximum difference is below 0.015.

![Cost-Effectiveness Acceptability Curve](a)

**Figure 4.1.** (a) True cost-effectiveness acceptability curve and approximated cost-effectiveness acceptability curves of Case A from LM metamodels when using 5000 training samples, (b) Difference between true cost-effectiveness acceptability curve and approximated cost-effectiveness acceptability curves of Case A from LM metamodels when using 5000 training samples.

To test whether the metamodeling approach will be affected by the training size, training samples of 2000, 1000, 500 and 200 observations are used to train LM metamodels separately. With smaller training size, the metamodeling approach could still reproduce true CEAC closely. When training the LM metamodel with 200 samples, the approximation error increases slightly but it is below 1.4%. In Figure 4.2(a), the approximated CEACs are in close proximity to the true CEAC, which demonstrates that the metamodeling approach functions well in this scenario. In addition, Figure 4.2(b) supports the observation.
4.1.2 Linear Model with Two-Way Interaction Terms (LM2)

Furthermore, interaction terms are added to introduce nonlinearity in metamodel. LM2 metamodels are performed, which results in approximation error 0.26% (95% CI: 0.25%-0.27%) when using 5000 training samples. The true CEAC and the approximated CEACs are compared in Figure 4.3(a). According to the Figure 4.3(b), the difference between the true CEAC and the approximated CEACs is smaller than that from linear metamodels without interaction terms.

The performances of metamodeling approach are approximately equal when training on 5000, 2000, 1000 and 500 samples, which lead to approximation errors 0.26% or 0.27%. LM2 metamodels achieve approximation error 0.32% when as few as 200 training samples are used. In this test case, 500 training samples are adequate to capture the nonlinearity of microsimulation model output. However, metamodel of the same structure are unable to obtain the similar performance when using 200 training samples.

Introducing nonlinearity in metamodel reduces the approximation error. LM2 metamodels perform better than LM metamodels when using the same number of training samples. In this analysis, the approximation error of LM2 metamodel using 200 training samples is even smaller than that of LM metamodel using 5000 training samples.
4.1.3 LASSO Regression with Two-Way Interaction Terms

LASSO metamodels are conducted for linear model selection, which can select the subset of the most important variables compared to LM2 metamodels. The approximation error is 0.28% (95% CI: 0.27%-0.29%) when training on 5000 samples. It increases to 0.42% (95% CI: 0.38%-0.47%) when using 200 training samples.

The performance of LASSO metamodels is worse than that of LM2 metamodels in this test. The reason is that the sick-sicker microsimulation is an ideal model with an obvious relation between input and output variables. The main sampled input parameters are annual transition probabilities, annual costs and effectiveness weights. Clearly, the incremental cost outputs ($\Delta C$) are related to annual transition probabilities and annual costs, and the incremental effect outputs ($\Delta E$) are relevant to probabilities and effectiveness weights. Since metamodels are trained separately for $\Delta C$ and $\Delta E$, irrelevant input variables are excluded in corresponding metamodels. The effectiveness weights are not involved in metamodels for $\Delta C$. Similarly, the annual costs are not included in metamodels for $\Delta E$. However, the advantages of variable selection in LASSO metamodels will be uncovered in the real-world test case of lymphoma treatment model, where the relation between input and output is blurred.

Similar to the LM2 metamodels, there is a slight increase in approximation errors when training the LASSO metamodels with smaller training sizes. In addition, the
interaction terms in LASSO reduce the approximation errors compared with LM.

4.1.4 Generalized Additive Model (GAM)

The performance of GAMs metamodels is evaluated. Compared to linear regression and LASSO, the structure of GAMs is more complex. A separate function can be calculated for each variable in GAMs, therefore, nonlinearity can be added to each variable independently.

We use gam package in R to fit GAM regression. Smoothing spline is used as basis function for variables. By default setting of the package, each function of variables is specified as 4 degrees of freedom. Interaction effects are not included when fitting GAMs.

Same as the findings above, the approximation errors of GAMs metamodels increase when using fewer training samples. The approximation errors of GAMs metamodels are less than 1.40% for all tested training sizes. It indicates that GAMs metamodels can reproduce the true CEAC fairly well. The performance of GAMs metamodels is similar to that of LM metamodels, but worse than the performance of LM2 and LASSO metamodels. Thus, the complexity of GAMs metamodels does not necessarily contribute to a better result.

4.1.5 Neural Networks

Neural networks metamodels are performed with keras package in R. Since the input parameters are scaled with standardization, the order of magnitude of input parameters becomes one. The neural networks metamodels are implemented to predict $\Delta C$ (order of magnitude as three) and $\Delta E$ (order of magnitude as one) from input parameters separately. Due to the different magnitudes between $\Delta C$ and $\Delta E$ outputs, the structures of corresponding neural networks are different. The neural networks of $\Delta C$ are consisted of two hidden layers with 30 neurons in the first hidden layer and 20 neurons in the second one. The neural networks of $\Delta C$ are trained until 7000 epochs when the value of loss function (MSE) is stable. There are two hidden layers in neural networks of $\Delta E$ with 15 neurons in the first hidden layer and 10 neurons in the second hidden layer. The neural networks of $\Delta E$ are trained until 500 epochs when the value of loss function reaches stability. The structures of neural networks are consistent for all the tested training size in base case as well as alternative cases of two comparators.

The structures of neural networks are more complex compared with linear models, but it does not necessarily improve the results. The approximation error is 0.78% (95%
CI: 0.66%-0.91%) when using 5000 training samples, which is smaller than approximation errors of LM but higher than that of LM2 and LASSO metamodels. The approximation errors of neural networks increase a little bit with fewer training samples. When as few as 200 training samples are used, the approximation error becomes 1.49% (95% CI: 1.25%-1.72%) which is still higher than that of LM2 and LASSO metamodels. The performance is even worse than LM and GAMs metamodels.

In this analysis, complex model like neural networks does not certainly achieve better performance. It can be challenging if there is not enough training samples or not proper structure. It is possible that meaningful approximation cannot be obtained for some neural networks with simple structures, which indicate fewer hidden layers and hidden neurons. In addition, neural networks with complex structures require more time to run. But the performance of tested neural networks is reasonably good, the approximation error is below 1.50% for all tested training sizes.

### 4.2 Base Case Results of Three Comparators

To test whether the metamodeling approach will still work for more comparators, the analysis with two comparators is further extended to three comparators. Though the shapes of CEAC change, metamodels show similar performance of approximating CEACs in the cases with three comparators. The base case approximation errors of three comparators are shown in Table 4.2.

<table>
<thead>
<tr>
<th>Training Size</th>
<th>LM</th>
<th>LM2</th>
<th>LASSO</th>
<th>GAMs</th>
<th>Neural Networks</th>
</tr>
</thead>
<tbody>
<tr>
<td>5000</td>
<td>0.98 (0.94-1.01)</td>
<td>0.26 (0.25-0.28)</td>
<td>0.44 (0.38-0.49)</td>
<td>1.05 (1.01-1.08)</td>
<td>1.11 (0.89-1.32)</td>
</tr>
<tr>
<td>2000</td>
<td>1.01 (0.95-1.07)</td>
<td>0.32 (0.30-0.35)</td>
<td>0.56 (0.47-0.64)</td>
<td>1.02 (0.97-1.08)</td>
<td>1.59 (1.33-1.84)</td>
</tr>
<tr>
<td>1000</td>
<td>1.12 (1.02-1.22)</td>
<td>0.34 (0.31-0.38)</td>
<td>0.75 (0.64-0.87)</td>
<td>1.07 (0.99-1.14)</td>
<td>1.63 (1.19-2.08)</td>
</tr>
<tr>
<td>500</td>
<td>1.38 (1.21-1.55)</td>
<td>0.44 (0.39-0.49)</td>
<td>0.87 (0.72-1.01)</td>
<td>1.23 (1.11-1.35)</td>
<td>2.02 (1.62-2.43)</td>
</tr>
<tr>
<td>200</td>
<td>1.63 (1.35-1.90)</td>
<td>0.59 (0.49-0.68)</td>
<td>1.13 (0.92-1.35)</td>
<td>1.54 (1.39-1.69)</td>
<td>2.65 (2.08-3.22)</td>
</tr>
</tbody>
</table>

#### 4.2.1 Linear Model (LM)

In the analysis with three comparators on 5000 training samples, we observe that LM metamodel generates CEACs with an average approximation error 0.98% (95% CI: 0.94%-1.01%). With fewer training samples, the approximation errors increase slightly. LM metamodels achieve approximation error below 1.65% when as few as 200 training samples are used.
4.2.2 Linear Model with Two-Way Interaction Terms (LM2)

The interaction terms obtain the nonlinearity of microsimulation model output and further reduce the approximation errors. LM2 metamodel attains approximation error as 0.26%, compared with 0.98% of LM metamodel when using 5000 training samples. The approximation error increases to 0.59% when using 200 samples. In this scenario, the approximated CEACs are still close to the true CEAC and the difference between them is below 0.60%. Figure 4.4(a) and Figure 4.4(b) plot the true CEAC and approximated CEACs from LM2 metamodels and the difference between them when using 1000 training samples.

![Figure 4.4](image)

Figure 4.4. (a) True cost-effectiveness acceptability curve and approximated cost-effectiveness acceptability curves of 3 treatments from LM2 metamodels when using 1000 training samples, (b) Difference between true cost-effectiveness acceptability curve and approximated cost-effectiveness acceptability curves of 3 treatments from LM2 metamodels when using 1000 training samples.

4.2.3 LASSO Regression with Two-Way Interaction Terms

Similarly, the advantages of LASSO regression are not obvious in this synthetic test case. The relation between model input and output is apparent, therefore, all the relevant input variables are included in corresponding metamodels. When performing LASSO regression, some input variables are dropped during variable subset selection. The advantages of LASSO regression will be further explored in real-world test case of lymphoma treatment model, where the relation between input and output variables is unclear.

LASSO metamodel results in approximation error 0.44% (95% CI: 0.38%-0.49%) when using 5000 training samples. It increases to 1.13% (95% CI: 0.92%-1.35%) with
200 training samples. Compared to the LM2 metamodels, the approximation errors are larger when using the same training size. However, the approximation errors are smaller in comparison to LM metamodels.

### 4.2.4 Generalized Additive Model (GAM)

Similar to the analysis with two comparators, the GAMs metamodel can reproduce the true CEAC reasonably well. The approximation errors increase from 1.05% when using 5000 training samples to 1.54% when using as few as 200 training samples. However, its performance is worse than that of LM2 and LASSO metamodels. The complex structure of GAMs does not improve the results necessarily.

### 4.2.5 Neural Networks

Neural networks metamodels are performed for $\Delta C_1$ (incremental cost between treatment II and treatment I), $\Delta E_1$ (incremental effect between treatment II and treatment I), $\Delta C_2$ (incremental cost between treatment III and treatment I) and $\Delta E_2$ (incremental effect between treatment III and treatment I) individually.

Due to the different magnitudes between incremental cost and incremental effect outcomes, the structures of corresponding neural networks are different. The neural networks for $\Delta C_1$ and $\Delta C_2$ are composed of three hidden layers with 60 hidden neurons in the first hidden layer, 40 hidden neurons in the second one and 25 hidden neurons in the third one. The neural networks of incremental cost are trained until 6000 epochs when the value of loss function is unchanged. Similarly, the neural networks for $\Delta E_1$ and $\Delta E_2$ are consisted of two hidden layers with 20 hidden neurons in the first hidden layer and 15 neurons in the second one. The neural networks of incremental effect are trained until 500 epochs. The structures of neural networks are consistent for all the tested training size in base case as well as alternative case of three comparators and real-world test case.

Similar to the analysis with two comparators, the structure of neural networks is more complex than linear models. However, the results demonstrate that complex models does not necessarily obtain better performance. The approximation errors of neural networks metamodels are the highest among all tested types of metamodels. The approximation errors of tested neural networks are even four times than that of LM2 metamodels when using the same amount of training samples. The performance of neural networks is sensitive to its structure. Adding more hidden layers and hidden neurons is one way to
increase the complexity of neural networks structure. There is possibility that neural networks with more complicated structures may generate better results.

4.3 Effect of CEAC Shapes

To evaluate the performance of the metamodeling approach on different shapes of CEAC, the parameters are modified to generate different CEAC shapes of two and three comparators. Two alternative cases of two comparators and one alternative case of three comparators are analyzed. LM2 and neural networks metamodels are applied in alternative cases. The results are demonstrated in Table 4.3.

In analysis with two comparators, the LM2 metamodels generate CEACs with approximation error less than 0.35% when using training size of 2000, 1000 and 200. The approximation errors of neural networks metamodels are higher, which is below 1.50% for the tested cases and training sizes.

The model structures of alternative cases is the same as that of base case, however, the performance of alternative cases is better. The reason is related to the CEAC shapes of different cases. The CEAC of base case changes dramatically over the entire range of WTP. However, the CEAC of Case H remains unchanged before the first quartile of WTP and the CEAC of Case E is unchanged after the first quartile of WTP. Approximating true CEAC of Case A could generate more nuances over the entire range of WTP. In general, the metamodeling approach works reasonably well for all the tested cases of two comparators. The approximation errors are below 1.00% for most of the cases.

The approximation errors slightly increase in analysis with three comparators. However, the metamodeling approach still works well, which reproduces the true CEAC with approximation errors under 1.00% when using LM2 metamodels. The approximation errors of alternative case are higher than that of base case when using neural networks metamodels. The structure of neural networks is the same for these two cases to test the effect of different shapes of CEACs. The performance of neural networks metamodels is sensitive to its structure, which may not be perfectly suitable for the alternative case.

The result indicates that the metamodeling approach is robust for all the tested cases. The approximation error is below 3.00% in most of the cases.

4.4 Effect of Sampling Schemes

To test whether the metamodeling approach relies on specific sampling methods, we use additional sampling schemes like Latin hypercube sampling (LHS) and full factorial design.
The structures of neural networks are consistent for different sampling schemes when testing on specific comparators. Results of different sampling schemes are demonstrated in Table 4.4.

The default sampling method is probabilistic sensitivity analysis (PSA), which samples the input variables from distributions. LHS gives uniformly distributed input parameter values. PSA and LHS are performed with training size of 5000, 2000 and 500. Moreover, deterministic training samples are produced from full factorial design. Since there are 7 input variables in metamodels of costs and 6 in metamodels of effectiveness, the full factorial design runs with $2^7$ and $2^6$ observations in corresponding metamodel.

The approximation errors from generating training samples with full factorial design could be a little bit higher. The reason is that information between extreme value of each variables is not available, and the metamodels can not capture enough variation in the parameters.

LHS does not require more samples for more variables. Due to this feature of independence, the results from generating training samples with LHS are less sensitive to training size compared with PSA. The approximation errors change slightly when the training size decreases more than 50% in LM2 metamodels. The performance of neural networks goes worse a little bit when using different training sizes in the analysis of three comparators. To compare LHS and PSA, the structure of neural networks using LHS samples is the same as that using PSA samples. The performance of neural networks is sensitive to its structure and input values. The results indicate that the same structure does not work for LHS samples of three comparators very well. However, the performance of neural networks metamodels using LHS samples is similar to that using PSA samples in the analysis of two comparators.

4.5 Real-World Cost-Effectiveness Analysis Study: Lymphoma Treatment Model

In previous sections, the metamodeling approach has been tested on the synthetic data. Real-world research data from lymphoma treatment [1] are utilized in this section to further evaluate the robustness of the metamodeling approach.

Microsimulation models are developed to compare standard treatment, novel treatment and subtype-based treatments. There are two treatments in the subtype-based strategy, one is gene expression profiling (GEP) test which is the gold standard and the other one is immunohistochemistry (IHC) testing using the Hans algorithm. Only GEP test is considered in the subtype-based treatment. Therefore, there are three comparators in
### Table 4.3. Approximation Errors (%) of Different Shapes of CEAC

<table>
<thead>
<tr>
<th>Metamodels</th>
<th>Training Size</th>
<th>LM2</th>
<th>Neural Networks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2000</td>
<td>1000</td>
<td>200</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Two Comparators</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Base Case (Case A)</td>
<td>0.27 (0.26-0.28)</td>
<td>0.26 (0.25-0.27)</td>
<td>0.32 (0.29-0.34)</td>
</tr>
<tr>
<td>Alternative 1 (Case H)</td>
<td>0.21 (0.20-0.22)</td>
<td>0.22 (0.21-0.24)</td>
<td>0.26 (0.23-0.29)</td>
</tr>
<tr>
<td>Alternative 2 (Case E)</td>
<td>0.23 (0.21-0.25)</td>
<td>0.24 (0.22-0.26)</td>
<td>0.24 (0.21-0.27)</td>
</tr>
<tr>
<td>Three Comparators</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Base Case</td>
<td>0.32 (0.30-0.35)</td>
<td>0.34 (0.31-0.38)</td>
<td>0.59 (0.49-0.68)</td>
</tr>
<tr>
<td>Alternative 1</td>
<td>0.22 (0.21-0.24)</td>
<td>0.37 (0.34-0.39)</td>
<td>0.42 (0.38-0.47)</td>
</tr>
</tbody>
</table>

### Table 4.4. Approximation Errors (%) of Different Sampling Schemes

<table>
<thead>
<tr>
<th>Sampling Scheme</th>
<th>Training Size</th>
<th>PSA</th>
<th>LHS</th>
<th>Full Factorial Design</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5000</td>
<td>2000</td>
<td>500</td>
<td>2000</td>
</tr>
<tr>
<td>Two Comparators</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LM2</td>
<td>0.26 (0.25-0.27)</td>
<td>0.27 (0.26-0.28)</td>
<td>0.27 (0.26-0.29)</td>
<td>0.22 (0.21-0.22)</td>
</tr>
<tr>
<td>Neural Networks</td>
<td>0.78 (0.66-0.91)</td>
<td>0.89 (0.73-1.04)</td>
<td>1.27 (1.01-1.54)</td>
<td>0.97 (0.85-1.08)</td>
</tr>
<tr>
<td>Three Comparators</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LM2</td>
<td>0.26 (0.25-0.28)</td>
<td>0.32 (0.30-0.35)</td>
<td>0.44 (0.39-0.49)</td>
<td>0.22 (0.21-0.23)</td>
</tr>
<tr>
<td>Neural Networks</td>
<td>1.11 (0.89-1.32)</td>
<td>1.59 (1.33-1.84)</td>
<td>2.02 (1.62-2.43)</td>
<td>1.11 (0.94-1.28)</td>
</tr>
<tr>
<td>Training Size</td>
<td>LM</td>
<td>LM2</td>
<td>LASSO</td>
<td>GAMs</td>
</tr>
<tr>
<td>--------------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
</tr>
<tr>
<td>2000</td>
<td>3.64 (3.52-3.75)</td>
<td>–</td>
<td>0.96 (0.92-0.99)</td>
<td>3.51 (3.41-3.61)</td>
</tr>
<tr>
<td>1000</td>
<td>3.71 (3.50-3.92)</td>
<td>–</td>
<td>1.02 (0.98-1.06)</td>
<td>3.60 (3.40-3.79)</td>
</tr>
<tr>
<td>500</td>
<td>3.79 (3.57-4.02)</td>
<td>–</td>
<td>1.10 (1.04-1.15)</td>
<td>3.61 (3.41-3.81)</td>
</tr>
<tr>
<td>200</td>
<td>3.81 (3.38-4.24)</td>
<td>–</td>
<td>1.28 (1.18-1.37)</td>
<td>3.51 (3.22-3.79)</td>
</tr>
</tbody>
</table>

the analysis with real-world research data. CEACs of these three treatments are drawn in Figure 4.5.

![Figure 4.5](image)

**Figure 4.5.** Cost-effectiveness acceptability curves of three lymphoma treatments from published paper [1].

The input parameters are generated from 10,000 PSA iterations and their base values, ranges and distributions are referred to Table 1 in [1]. Though IHC testing is excluded in the comparison, the related input parameters is reserved including the probability and cost of IHC test. Model outputs include total life years, effectiveness (measured with quality-adjusted life-years), direct costs of each treatment strategy, and ICERs compared with standard treatment. To be consistent with the analysis in synthetic data, metamodels are trained to predict incremental cost and incremental effect outcomes and further approximate true CEAC. Similarly, the true CEAC is from independently 10,000 sampled parameter sets.

LM, LM2, LASSO, GAMs and neural networks metamodels are performed individually. Generally, the metamodeling approach could closely reproduce the true CEAC with proper choice of metamodels in this analysis.

The approximation error of LM metamodel is 3.79% (95% CI: 3.57%-4.02%) when
500 training samples are used, and decreases to 3.64% (95% CI: 3.52%-3.75%) when using 2000 training samples. The performance of LM metamodels improves when the training size is increased, which is similar to the analysis of synthetic data. True CEACs and approximated CEACs from LM metamodels when using 500 training samples are plotted in Figure 4.6.

![Cost-Effectiveness Acceptability Curve](image)

**Figure 4.6.** True cost-effectiveness acceptability curves and approximated cost-effectiveness acceptability curves of three lymphoma treatments from LM metamodels when using 500 training samples.

LM2 metamodel cannot obtain meaningful approximation of true CEAC in this test case. Without subset selection of variables, the LM2 metamodel will include all the input variables in the model. Linear regression metamodels cannot distinguish and remove the unrelated input variables in this analysis, which further adds errors to the approximated CEACs.

After adding nonlinearity and performing variable selection, the LASSO metamodel achieves the approximation error of 0.96% (95% CI: 0.92%-0.99%) when using 2000 training samples and 1.10% when using 500 training samples. The advantages of LASSO metamodel are illustrated in this analysis. LASSO metamodel overcomes the disadvantage of linear regression metamodel. It is simpler and more interpretable that involves only a subset of the input variables. True CEACs and approximated CEACs from LASSO metamodels when using 500 training samples are plotted in Figure 4.7.
We use complex models like GAMs and neural networks, which does not necessarily improve the approximation performance. The GAMs metamodel can obtain meaningful approximation of true CEAC, but its performance is worse than that of LASSO metamodels.

![Cost-Effectiveness Acceptability Curve](image)

**Figure 4.7.** True cost-effectiveness acceptability curves and approximated cost-effectiveness acceptability curves of three lymphoma treatments from LASSO metamodels when using 500 training samples.

In this analysis, we perform LM, LM2, LASSO, GAMs and neural networks metamodels on real-world research data from lymphoma treatment. LM metamodel can reproduce the true CEAC. Since there are no interaction effects in LM metamodel, its performance is on the average level. LM2 metamodel cannot achieve meaningful approximation in this test. LM2 metamodel includes all the input variables in the model. Without variable selection, LM2 metamodel is unable to distinguish and remove the unrelated input variables.

The approximation performance of LASSO metamodel is the best among these five types of metamodels. The advantages of LASSO metamodel are clear in this analysis. It can overcome the disadvantage of LM2 metamodel by selecting a subset of related input variables. After adding nonlinearity and applying variable selection, LASSO metamodel can closely reproduce the true CEAC.

The structure of GAMs and neural networks metamodels is more complex, but it
does not necessarily result in a better approximation performance. The performance of GAMs and neural networks metamodels is worse than that of LASSO metamodel. The performance of GAMs metamodel is stable when training size varies from 2000 to 200. The approximation errors of neural networks metamodel increases when using fewer training samples. The accuracy of neural networks metamodel is sensitive to its structure. One possible reason is that the current structure of neural networks metamodel is not complex enough. Adding more hidden layers and hidden neurons can improve the approximation performance. However, it may increase the computational burden which obeys the idea of performing the metamodeling approach. It is challenging to achieve a balance between approximation performance and computation time when using neural networks metamodel.
Chapter 5
Discussion and Conclusion

Conducting probabilistic sensitivity analysis (PSA) for complex simulation models could be computationally challenging. The metamodeling approach is commonly used to simplify the model representation, and to further reduce the computational burden. This study aims to explore whether the metamodeling approach can be used to represent the model uncertainty and approximate cost-effectiveness acceptability curves (CEACs) in cost-effectiveness analysis. We adapt the microsimulation model from a tutorial example [118]. First, we design the overall framework. The model parameters are sampled from recommended statistical distribution [36], and the model outputs are cost and effect for each treatment. Metamodels are built separately for incremental cost and incremental effect between treatments. The true CEAC can be achieved by running the simulation model with 10,000 independently sampled parameter sets (testing set), which is considered as gold standard. Similarly, the approximated CEAC can be generated from running metamodels with the same testing set. To measure the performance of the metamodeling approach, the approximation error is defined as the maximum absolute value of the difference between true CEAC and the approximated CEACs. We repeat the analysis for 20 times to account for sampling randomness, which is presented as 95% confidence interval in the experiment results. Second, variations of CEAC shapes, metamodel types, training sizes and sampling methods are considered to assess the robustness of the metamodeling approach. We modify the values of input parameters to generate different shapes of CEACs. Three cases of two comparators and two cases of three comparators are included. Then we use five types of metamodels to test whether the choice of metamodels will affect the approximation performance. Metamodels of LM, LM2, LASSO2, GAMs and neural networks are performed using 5000, 2000, 1000, 500 and 200 training samples individually. Different sampling methods, including PSA, Latin hypercube sampling (LHS) and full factorial design are applied to evaluate whether the metamodeling approach relies on specific sampling method. Third, the metamodeling approach is tested on real-world research data from lymphoma treatment [1].
In this study, the metamodeling approach is tested to simplify the relation between input and output of a microsimulation model. However, the complexity of microsimulation model is limited where the advantage of using neural networks metamodel has not been thoroughly explored. The accuracy of neural networks metamodel is sensitive to its structure. Adding more hidden layers and hidden neurons can result in a better approximation, while it may increase the computational burden as well. The trade-off between accuracy and computation time of neural networks metamodel is inevitable. Thus, achieving a balance between performance and computation time is challenging when using neural networks metamodels. Moreover, training neural networks with complex structures is time-consuming compared to linear model, which may obey the idea of applying the metamodeling approach.

Whether the metamodeling approach will work for more complex and dynamic simulation model has not been tested. Although microsimulation model is time-consuming, its structure is simpler than agent-based model. The advantage of neural networks metamodels may be revealed when testing on agent-based models.

Experiments are required to explore whether the performance of metamodeling approach will change due to the accuracy of each simulation run. Future research can test the metamodeling approach on simulations with fewer simulated individuals or fewer Monte-Carlo simulation replications.

Additional types of metamodels can be used as well. Gaussian process regression, a non-parametric regression approach, can represent an unknown function flexibly [61]. In case of approximating CEAC, function of incremental cost and incremental effect can be modeled as Gaussian process.

To further reduce the computational burden, multi-output metamodeling approach can be explored to analyze PSA of complex model. With multi-output structure, incremental cost and incremental effect can be predicted with one single metamodel. One potential challenging in multi-output structure is that outputs can be correlated. Thus, introducing regularization terms or performing variable selection is essential in multi-output metamodeling approach.

In conclusion, metamodels could closely approximate the CEAC results in PSA of cost-effectiveness analysis. The metamodeling approach performs well for different shapes of CEAC and different training sizes, ranging from two comparators to three comparators and from 5000 training samples to 200 training samples. When performing the metamodeling approach on the same case, including nonlinearity could improve the approximation performance compared with simple linear models. From the analysis of
effects of different sampling schemes, we can conclude that approximation performance from full factorial design is not as well as the sampling-based methods, like PSA and LHS. Full factorial design generates deterministic training samples, where no information between extreme value of variables can be obtained.
# Appendix A

## Input Parameters for Alternative Cases of Two Comparators

### Table A.1. Input Parameters of Case H

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Distribution</th>
<th>Mean</th>
<th>SD</th>
<th>LB</th>
<th>UB</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Annual Transition Probabilities</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p.HS1</td>
<td>Disease onset (H to S1)</td>
<td>beta(21.1,119.56)</td>
<td>0.150</td>
<td>0.030</td>
<td>0.104</td>
<td>0.202</td>
</tr>
<tr>
<td>p.S1H</td>
<td>Recovery (S1 to H)</td>
<td>beta(12,12)</td>
<td>0.500</td>
<td>0.100</td>
<td>0.335</td>
<td>0.665</td>
</tr>
<tr>
<td>p.S1S2</td>
<td>Disease progression (S1 to S2)</td>
<td>beta(22.27,189.83)</td>
<td>0.105</td>
<td>0.021</td>
<td>0.073</td>
<td>0.142</td>
</tr>
<tr>
<td><strong>Annual Costs</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c.H</td>
<td>Healthy individuals</td>
<td>gamma(25,80)</td>
<td>2000</td>
<td>400</td>
<td>1390</td>
<td>2700</td>
</tr>
<tr>
<td>c.S1</td>
<td>Sick individuals in S1</td>
<td>gamma(25,160)</td>
<td>4000</td>
<td>800</td>
<td>2781</td>
<td>5400</td>
</tr>
<tr>
<td>c.S2</td>
<td>Sick individuals in S2</td>
<td>gamma(25,600)</td>
<td>15000</td>
<td>3000</td>
<td>10429</td>
<td>20251</td>
</tr>
<tr>
<td>c.Trt</td>
<td>Annual treatment cost per sick individual (S1 and S2)</td>
<td>gamma(25,400)</td>
<td>10000</td>
<td>2000</td>
<td>6952</td>
<td>13500</td>
</tr>
<tr>
<td><strong>Utility weights</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>u.S1</td>
<td>Sick individuals in S1</td>
<td>beta(24.25,8.08)</td>
<td>0.750</td>
<td>0.075</td>
<td>0.618</td>
<td>0.864</td>
</tr>
<tr>
<td>u.S2</td>
<td>Sick individuals in S2</td>
<td>beta(49.5,49.5)</td>
<td>0.500</td>
<td>0.050</td>
<td>0.418</td>
<td>0.582</td>
</tr>
<tr>
<td><strong>Intervention effect</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>u.Trt</td>
<td>Utility for treated individuals in S1 (SD)</td>
<td>beta(14.15,2.497)</td>
<td>0.850</td>
<td>0.085</td>
<td>0.689</td>
<td>0.962</td>
</tr>
</tbody>
</table>

### Table A.2. Input Parameters of Treatment I in Case E

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Distribution</th>
<th>Mean</th>
<th>SD</th>
<th>LB</th>
<th>UB</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Annual Transition Probabilities</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p.HS1</td>
<td>Disease onset (H to S1)</td>
<td>beta(7,3)</td>
<td>0.700</td>
<td>0.138</td>
<td>0.450</td>
<td>0.902</td>
</tr>
<tr>
<td>p.S1H</td>
<td>Recovery (S1 to H)</td>
<td>beta(12,12)</td>
<td>0.500</td>
<td>0.100</td>
<td>0.335</td>
<td>0.665</td>
</tr>
<tr>
<td>p.S1S2</td>
<td>Disease progression (S1 to S2)</td>
<td>beta(22.27,189.83)</td>
<td>0.105</td>
<td>0.021</td>
<td>0.073</td>
<td>0.142</td>
</tr>
<tr>
<td><strong>Annual Costs</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c.H</td>
<td>Healthy individuals</td>
<td>gamma(5,80)</td>
<td>400</td>
<td>179</td>
<td>158</td>
<td>732</td>
</tr>
<tr>
<td>c.S1</td>
<td>Sick individuals in S1</td>
<td>gamma(5,160)</td>
<td>800</td>
<td>358</td>
<td>315</td>
<td>1465</td>
</tr>
<tr>
<td>c.S2</td>
<td>Sick individuals in S2</td>
<td>gamma(5,600)</td>
<td>3000</td>
<td>1342</td>
<td>1182</td>
<td>5492</td>
</tr>
<tr>
<td>c.Trt</td>
<td>Annual treatment cost per sick individual (S1 and S2)</td>
<td>gamma(5,4)</td>
<td>20</td>
<td>9</td>
<td>8</td>
<td>37</td>
</tr>
<tr>
<td><strong>Utility weights</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>u.S1</td>
<td>Sick individuals in S1</td>
<td>beta(24.25,50)</td>
<td>0.327</td>
<td>0.054</td>
<td>0.240</td>
<td>0.418</td>
</tr>
<tr>
<td>u.S2</td>
<td>Sick individuals in S2</td>
<td>beta(49.5,148.5)</td>
<td>0.250</td>
<td>0.031</td>
<td>0.201</td>
<td>0.302</td>
</tr>
<tr>
<td><strong>Intervention effect</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>u.Trt</td>
<td>Utility for treated individuals in S1 (SD)</td>
<td>beta(14.15,2.497)</td>
<td>0.850</td>
<td>0.085</td>
<td>0.689</td>
<td>0.962</td>
</tr>
<tr>
<td>Parameter Description</td>
<td>Distribution</td>
<td>Mean</td>
<td>SD</td>
<td>LB</td>
<td>UB</td>
<td></td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------</td>
<td>-------</td>
<td>------</td>
<td>-----</td>
<td>-----</td>
<td></td>
</tr>
<tr>
<td><strong>Annual Transition Probabilities</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p.HS1 Disease onset (H to S1)</td>
<td>beta(21.1,119.56)</td>
<td>0.150</td>
<td>0.030</td>
<td>0.104</td>
<td>0.202</td>
<td></td>
</tr>
<tr>
<td>p.S1H Recovery (S1 to H)</td>
<td>beta(12,12)</td>
<td>0.500</td>
<td>0.100</td>
<td>0.335</td>
<td>0.665</td>
<td></td>
</tr>
<tr>
<td>p.S1S2 Disease progression (S1 to S2)</td>
<td>beta(22.27,189.83)</td>
<td>0.105</td>
<td>0.021</td>
<td>0.073</td>
<td>0.142</td>
<td></td>
</tr>
<tr>
<td><strong>Annual Costs</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c.H Healthy individuals</td>
<td>gamma(20,30)</td>
<td>600</td>
<td>134</td>
<td>398</td>
<td>836</td>
<td></td>
</tr>
<tr>
<td>c.S1 Sick individuals in S1</td>
<td>gamma(20,60)</td>
<td>1200</td>
<td>268</td>
<td>795</td>
<td>1673</td>
<td></td>
</tr>
<tr>
<td>c.S2 Sick individuals in S2</td>
<td>gamma(20,160)</td>
<td>3200</td>
<td>715</td>
<td>2121</td>
<td>4461</td>
<td></td>
</tr>
<tr>
<td>c.Trt Annual treatment cost per sick individual (S1 and S2)</td>
<td>gamma(20,50)</td>
<td>1000</td>
<td>224</td>
<td>663</td>
<td>1394</td>
<td></td>
</tr>
<tr>
<td><strong>Utility weights</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>u.S1 Sick individuals in S1</td>
<td>beta(24.25,50)</td>
<td>0.327</td>
<td>0.054</td>
<td>0.240</td>
<td>0.418</td>
<td></td>
</tr>
<tr>
<td>u.S2 Sick individuals in S2</td>
<td>beta(49.5,148.5)</td>
<td>0.250</td>
<td>0.031</td>
<td>0.201</td>
<td>0.302</td>
<td></td>
</tr>
<tr>
<td><strong>Intervention effect</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>u.Trt Utility for treated individuals in S1 (SD)</td>
<td>beta(14.15,2.497)</td>
<td>0.850</td>
<td>0.085</td>
<td>0.689</td>
<td>0.962</td>
<td></td>
</tr>
</tbody>
</table>
Appendix B
Input Parameters for Alternative Case of Three Comparators

### Table B.1. Alternative Case Input Parameters of Treatment I & II in Three Comparators

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Distribution</th>
<th>Mean</th>
<th>SD</th>
<th>LB</th>
<th>UB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Annual Transition Probabilities</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p.HS1</td>
<td>Disease onset (H to S1)</td>
<td>beta(21.1,119.56)</td>
<td>0.150</td>
<td>0.030</td>
<td>0.104</td>
<td>0.202</td>
</tr>
<tr>
<td>p.S1H</td>
<td>Recovery (S1 to H)</td>
<td>beta(12,12)</td>
<td>0.500</td>
<td>0.100</td>
<td>0.335</td>
<td>0.665</td>
</tr>
<tr>
<td>p.S1S2</td>
<td>Disease progression (S1 to S2)</td>
<td>beta(22.27,189.83)</td>
<td>0.105</td>
<td>0.021</td>
<td>0.073</td>
<td>0.142</td>
</tr>
<tr>
<td></td>
<td><strong>Annual Costs</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c.H</td>
<td>Healthy individuals</td>
<td>gamma(25,80)</td>
<td>2000</td>
<td>400</td>
<td>1390</td>
<td>2700</td>
</tr>
<tr>
<td>c.S1</td>
<td>Sick individuals in S1</td>
<td>gamma(25,160)</td>
<td>4000</td>
<td>800</td>
<td>2781</td>
<td>5400</td>
</tr>
<tr>
<td>c.S2</td>
<td>Sick individuals in S2</td>
<td>gamma(25,600)</td>
<td>15000</td>
<td>3000</td>
<td>10429</td>
<td>20251</td>
</tr>
<tr>
<td>c.Trt</td>
<td>Annual treatment cost per sick individual (S1 and S2)</td>
<td>gamma(25,480)</td>
<td>12000</td>
<td>2400</td>
<td>8343</td>
<td>16201</td>
</tr>
<tr>
<td></td>
<td><strong>Utility weights</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>u.S1</td>
<td>Sick individuals in S1</td>
<td>beta(24.25,10)</td>
<td>0.327</td>
<td>0.054</td>
<td>0.240</td>
<td>0.418</td>
</tr>
<tr>
<td>u.S2</td>
<td>Sick individuals in S2</td>
<td>beta(49.5,148.5)</td>
<td>0.250</td>
<td>0.031</td>
<td>0.201</td>
<td>0.302</td>
</tr>
<tr>
<td></td>
<td><strong>Intervention effect</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>u.Trt</td>
<td>Utility for treated individuals in S1 (SD)</td>
<td>beta(14.15,2.497)</td>
<td>0.850</td>
<td>0.085</td>
<td>0.689</td>
<td>0.962</td>
</tr>
</tbody>
</table>

### Table B.2. Alternative Case Input Parameters of Treatment III in Three Comparators

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Distribution</th>
<th>Mean</th>
<th>SD</th>
<th>LB</th>
<th>UB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Annual Transition Probabilities</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>p.HS1</td>
<td>Disease onset (H to S1)</td>
<td>beta(21.1,119.56)</td>
<td>0.150</td>
<td>0.030</td>
<td>0.104</td>
<td>0.202</td>
</tr>
<tr>
<td>p.S1H</td>
<td>Recovery (S1 to H)</td>
<td>beta(12,12)</td>
<td>0.500</td>
<td>0.100</td>
<td>0.335</td>
<td>0.665</td>
</tr>
<tr>
<td>p.S1S2</td>
<td>Disease progression (S1 to S2)</td>
<td>beta(22.27,189.83)</td>
<td>0.105</td>
<td>0.021</td>
<td>0.073</td>
<td>0.142</td>
</tr>
<tr>
<td></td>
<td><strong>Annual Costs</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c.H</td>
<td>Healthy individuals</td>
<td>gamma(25,80)</td>
<td>2000</td>
<td>400</td>
<td>1390</td>
<td>2700</td>
</tr>
<tr>
<td>c.S1</td>
<td>Sick individuals in S1</td>
<td>gamma(25,160)</td>
<td>4000</td>
<td>800</td>
<td>2781</td>
<td>5400</td>
</tr>
<tr>
<td>c.S2</td>
<td>Sick individuals in S2</td>
<td>gamma(25,600)</td>
<td>15000</td>
<td>3000</td>
<td>10429</td>
<td>20251</td>
</tr>
<tr>
<td>c.Trt</td>
<td>Annual treatment cost per sick individual (S1 and S2)</td>
<td>gamma(25,480)</td>
<td>12000</td>
<td>2400</td>
<td>8343</td>
<td>16201</td>
</tr>
<tr>
<td></td>
<td><strong>Utility weights</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>u.S1</td>
<td>Sick individuals in S1</td>
<td>beta(24.25,10)</td>
<td>0.327</td>
<td>0.054</td>
<td>0.240</td>
<td>0.418</td>
</tr>
<tr>
<td>u.S2</td>
<td>Sick individuals in S2</td>
<td>beta(49.5,148.5)</td>
<td>0.250</td>
<td>0.031</td>
<td>0.201</td>
<td>0.302</td>
</tr>
<tr>
<td></td>
<td><strong>Intervention effect</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>u.Trt</td>
<td>Utility for treated individuals in S1 (SD)</td>
<td>beta(38,2)</td>
<td>0.950</td>
<td>0.034</td>
<td>0.884</td>
<td>0.991</td>
</tr>
</tbody>
</table>
Bibliography


