Published paper

Strong, M, Oakley, JE and Brennan, A (2013) Estimating Multiparameter Partial Expected Value of Perfect Information from a Probabilistic Sensitivity Analysis Sample: A Nonparametric Regression Approach. Med Decis Making. Doi: 10.1177/0272989X13505910

Medical Decision Making
http://mdm.sagepub.com/

Estimating Multiparameter Partial Expected Value of Perfect Information from a Probabilistic Sensitivity Analysis Sample: A Nonparametric Regression Approach

Mark Strong, Jeremy E. Oakley and Alan Brennan

Med Decis Making 2014 34: 311 originally published online 18 November 2013
DOI: 10.1177/0272989X13505910

Estimating Multiparameter Partial Expected Value of Perfect Information from a Probabilistic Sensitivity Analysis Sample: A Nonparametric Regression Approach

Mark Strong, PhD, Jeremy E. Oakley, PhD, Alan Brennan, PhD

The partial expected value of perfect information (EVPI) quantifies the expected benefit of learning the values of uncertain parameters in a decision model. Partial EVPI is commonly estimated via a 2-level Monte Carlo procedure in which parameters of interest are sampled in an outer loop, and then conditional on these, the remaining parameters are sampled in an inner loop. This is computationally demanding and may be difficult if correlation between input parameters results in conditional distributions that are hard to sample from. We describe a novel nonparametric regression-based method for estimating partial EVPI that requires only the probabilistic sensitivity analysis sample (i.e., the set of samples drawn from the joint distribution of the parameters and the corresponding net benefits). The method is applicable in a model of any complexity and with any specification of input parameter distribution. We describe the implementation of the method via 2 nonparametric regression modeling approaches, the Generalized Additive Model and the Gaussian process. We demonstrate in 2 case studies the superior efficiency of the regression method over the 2-level Monte Carlo method. R code is made available to implement the method. ***Key words:*** *value of information; expected value of perfect information; economic evaluation model; nonparametric regression; Bayesian decision theory; computational methods.* ***(Med Decis Making 2014;34:311–326)***

Health economic decision analytic models are used to estimate the expected net benefits of competing decision options. The true values of the input parameters of such models are rarely known with certainty, and it is often useful to quantify the value to the decision maker of reducing uncertainty about the model input parameters. The value of learning an input parameter (or a group of input parameters) can be quantified by its partial expected value of perfect information (partial EVPI).[1–4] The partial EVPI value for an input parameter reveals the sensitivity of the decision to our uncertainty about that input parameter, and as such can be used to inform the design and prioritization of future research.

The partial EVPI for a single parameter (or group of parameters) of interest is typically calculated via a 2-level nested Monte Carlo approach. This requires us to sample values of the input parameter(s) of interest in an outer loop and then to sample values from the joint conditional distribution of the remaining parameters and run the model in an inner loop.[5,6] We recognize 3 important limitations to this method. First, the 2-level method is computationally demanding for all but very simple models because of the

Received 8 February 2013 from the School of Health and Related Research (ScHARR), University of Sheffield, Sheffield, UK (MS, AB), and School of Mathematics and Statistics, University of Sheffield, Sheffield, UK (JEO). This report is independent research supported by the National Institute for Health Research (Mark Strong, postdoctoral fellowship PDF-2012-05-258). The views expressed in this publication are those of the authors and not necessarily those of the National Health Service, the National Institute for Health Research, or the Department of Health. Revision accepted for publication 27 August 2013.

Supplementary material for this article is available on the *Medical Decision Making* Web site at http://mdm.sagepub.com/supplemental.

Address correspondence to Mark Strong PhD, School of Health and Related Research (ScHARR), University of Sheffield, 30 Regent Street, Sheffield S1 4DA, UK; tel +44 (0)114 222 0812; e-mail m.strong@sheffield.ac.uk.

DOI: 10.1177/0272989X13505910

nested loop scheme. Second, the approach requires that the model is run as part of the EVPI calculation process, which may be difficult in certain software applications. Last, a potential problem arises in cases in which correlations exist between parameters. If the parameters of interest are correlated with the remaining parameters, then for the 2-level Monte Carlo method to work, there must be some method of sampling from the distribution of the remaining parameters, conditional on the values of the parameters of interest that have been sampled in the outer loop. If the required conditional distributions are difficult to sample from, say requiring Markov chain Monte Carlo (MCMC), then the computational burden will be substantially further increased.

Our experience is that although probabilistic sensitivity analyses (PSA) have become the norm in many economic evaluations for health technology assessment across the world, it is much less common for partial EVPIs to be estimated. In our view, the reasons for this are partly technical (in terms of the extra demands on the statistical and programming skills of the analyst), partly computational (the additional model development and model running time to implement 2 nested loops rerunning the model on each iteration), and partly structural (in that decision makers and research funding bodies have not always demanded these analyses).

The following scenario is typical of the kinds of problems we have encountered. A probabilistic sensitivity analysis sample (i.e., a set of sampled input parameters with their corresponding model outputs) has been generated for a patient-level simulation model. Each PSA run has required in the order of tens of thousands of patient-level runs of the simulation model to achieve convergence, with considerable computational cost. The analyst now wishes to estimate the partial EVPI value for a subset of input parameters (e.g., those that relate to clinical efficacy). Parameters within this subset of interest may be correlated with other input parameters. To achieve the partial EVPI calculation via the 2-level partial EVPI scheme might then have a computational cost of 1000 outer loops times 1000 inner loops times 10 000 runs of the patient-level simulation model (i.e., 10^{11} model evaluations in total). Not surprisingly, such calculations are often considered too computationally costly.

Recently, computationally efficient methods for calculating partial EVPI have been published,[7,8] but these work only when we require the partial EVPI for each model parameter separately. This restriction to single parameters is potentially problematic because we often expect research to update our knowledge about groups of parameters (for example, a set of relative risks or a group of related costs) rather than just single parameters.

In this article, we present a nonparametric regression-based method for calculating partial EVPI that overcomes the 3 limitations above and can be used to evaluate the partial EVPI of any subset of model parameters without rerunning the model. The article is structured as follows. In the second section, we introduce the nonparametric regression method and describe its general application. In the third section, we demonstrate the method in 2 case studies. In both of our case studies, we assume we have only a single PSA sample but wish to calculate the partial EVPI values of several sets of parameters of interest. The first case study is based on a model that is simple in structure but in which there are correlations between inputs. The second case study is a more complex Markov model. Both models have been used before for illustrative purposes.[5,7,9,10] In the fourth section, we conclude with a discussion of the implications and limitations of the approach.

METHOD

Partial EVPI

The partial expected value of information is the expected difference between the value of the optimal decision based on perfect information about those inputs and the value of the decision made only with prior information. To express this, we introduce some notation.

We assume that we are faced with D decision options, indexed $d = 1, \ldots, D$, and have built a model $\mathrm{NB}(d, \mathbf{x})$ that aims to predict the net benefit of decision option d given a vector of r input parameter values, $\mathbf{x} = (x_1, \ldots, x_r)$.

The true values of the input parameters are assumed to be unknown. We denote the true but unknown parameter values as upper case $\mathbf{X} = (X_1, \ldots, X_r)$ and a sample drawn from the joint distribution of the parameters as $\mathbf{x} = (x_1, \ldots, x_r)$. We denote the vector of p input parameters for which we wish to calculate the partial EVPI as $\mathbf{X}_i$ and the remaining $r - p$ input parameters as $\mathbf{X}_{-i}$. We denote the expectation over the full joint distribution of $\mathbf{X}$ as $E_{\mathbf{X}}$, over the marginal distribution of $\mathbf{X}_i$ as $E_{\mathbf{X}_i}$, and over the conditional distribution of $\mathbf{X}_{-i}|\mathbf{X}_i$ as $E_{\mathbf{X}_{-i}|\mathbf{X}_i}$.

The expected value of our optimal decision, made only with current information, is

$$\max_d E_{\mathbf{X}}\{\mathrm{NB}(d, \mathbf{X})\}. \tag{1}$$

If we knew the value of the inputs of interest, $\mathbf{X}_i$, then the optimal decision would be that with the greatest net benefit, after averaging over the conditional distribution of the remaining unknown inputs $\mathbf{X}_{-i}|\mathbf{X}_i$. The expected net benefit would be

$$\max_d E_{\mathbf{X}_{-i}|\mathbf{X}_i}\{\mathrm{NB}(d, \mathbf{X}_i, \mathbf{X}_{-i})\}. \tag{2}$$

But, because $\mathbf{X}_i$ is unknown, we must average over our current information about $\mathbf{X}_i$, giving

$$E_{\mathbf{X}_i}\left[\max_d E_{\mathbf{X}_{-i}|\mathbf{X}_i}\{\mathrm{NB}(d, \mathbf{X}_i, \mathbf{X}_{-i})\}\right]. \tag{3}$$

The partial EVPI for inputs $\mathbf{X}_i$ is the difference between Equation 3, the expected value of the decision made with perfect information about $\mathbf{X}_i$, and Equation 1, the expected value of the current optimal decision option,[3,4]

$$\begin{aligned}\mathrm{EVPI}(\mathbf{X}_i) = E_{\mathbf{X}_i}&\left[\max_d E_{\mathbf{X}_{-i}|\mathbf{X}_i}\{\mathrm{NB}(d, \mathbf{X}_i, \mathbf{X}_{-i})\}\right] \\ &- \max_d E_{\mathbf{X}}\{\mathrm{NB}(d, \mathbf{X})\}.\end{aligned} \tag{4}$$

We are commonly in a situation in which we cannot evaluate any of the 3 expectations in Equation 4 analytically. Important exceptions are cases in which models are either of linear form (e.g., $Y = \beta_1 X_1 + \beta_2 X_2$) or multilinear (sum-product) form (e.g., $Y = \beta_1 X_1 X_2 + \beta_2 X_3 X_4$; where β_1 and β_2 are constants). In the linear case, both the expectation in Equation 1 and the inner expectation in Equation 3 have an analytic solution, and in the multilinear case, these expectations have an analytic solution if inputs are independent.[11] In the case of correlated inputs, analytic solutions to these 2 expectations will sometimes exist, such as the case in which the inputs have a multivariate Normal distribution. The outer expectation in Equation 3 is more problematic because of the maximization step, and analytic solutions rarely exist. See Brennan and others for a fuller discussion.[5]

A PSA takes N samples from the joint distribution of the input parameters, $\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$, and generates a corresponding set of N net benefits, $\{\mathrm{NB}(d, \mathbf{x}^{(1)}), \ldots, \mathrm{NB}(d, \mathbf{x}^{(N)})\}$, for each decision option d. From this, the Monte Carlo solution to the second term in Equation 4 is simply $\max_d \frac{1}{N}\sum_{n=1}^{N} \mathrm{NB}(d, \mathbf{x}^{(n)})$.

The first term in Equation 4 requires more work, and unless there are analytic solutions to the expectations, the usual approach is to use a nested 2-level Monte Carlo method with K ''outer'' simulations and J ''inner'' simulations.[11] Here, the estimator is given by

$$\frac{1}{K}\sum_{k=1}^{K} \max_d \frac{1}{J}\sum_{j=1}^{J} \mathrm{NB}\left(d, \mathbf{x}_i^{(k)}, \mathbf{x}_{-i}^{(j,k)}\right), \tag{5}$$

where $\mathbf{x}_{-i}^{(j,k)}$ are samples drawn from the conditional distribution $\mathbf{X}_{-i}|\mathbf{X}_i = \mathbf{x}_i^{(k)}$.

Sufficient numbers of runs of both the outer and inner loops are required to ensure that the partial EVPI is estimated with sufficient precision and with an acceptable level of upward bias that is induced by the maximization step. For models that are slow to run, this 2-level scheme can represent a considerable computational burden.

To address the problems of the 2-level method, we focus our attention on the estimation of the inner expectation. To avoid the need for the inner loop simulation, we reframe the estimation of this conditional expectation as a regression problem.

Principles of Estimating Partial EVPI Using Regression

Our target is to estimate the conditional expectation $E_{\mathbf{X}_{-i}|\mathbf{X}_i = \mathbf{x}_i}\{\mathrm{NB}(d, \mathbf{x}_i, \mathbf{X}_{-i})\}$ evaluated at some particular value $\mathbf{x}_i$, because given this, the partial EVPI is easily obtained. To estimate this conditional expected net benefit, we undertake 3 conceptual moves.

First, we recognize that we can express the model output for model run n as a sum of the conditional expectation that we require, and a mean-zero error term,

$$\mathrm{NB}(d, \mathbf{x}^{(n)}) = E_{\mathbf{X}_{-i}|\mathbf{X}_i = \mathbf{x}_i^{(n)}}\{\mathrm{NB}(d, \mathbf{x}_i^{(n)}, \mathbf{X}_{-i})\} + \varepsilon^{(n)}. \tag{6}$$

To see why the error term must have zero mean, we rearrange and take expectations,

$$\begin{aligned}E(\varepsilon) &= E\{\mathrm{NB}(d, \mathbf{X})\} - E_{\mathbf{X}_i}[E_{\mathbf{X}_{-i}|\mathbf{X}_i = \mathbf{x}_i}\{\mathrm{NB}(d, \mathbf{x}_i, \mathbf{X}_{-i})\}] \\ &= E\{\mathrm{NB}(d, \mathbf{X})\} - E\{\mathrm{NB}(d, \mathbf{X}_i, \mathbf{X}_{-i})\} \\ &= E\{\mathrm{NB}(d, \mathbf{X})\} - E\{\mathrm{NB}(d, \mathbf{X})\} = 0.\end{aligned} \tag{7}$$

The second move is to realize that the expectation $E_{\mathbf{X}_{-i}|\mathbf{X}_i = \mathbf{x}_i^{(n)}}\{\mathrm{NB}(d, \mathbf{x}_i^{(n)}, \mathbf{X}_{-i})\}$ takes a different value for each value $\mathbf{x}_i^{(n)}$ and can therefore be thought of as a function of $\mathbf{x}_i$. We do not know the form of this function, but we can denote it as the unknown function $g(d, \mathbf{x}_i)$, allowing us to write for the n^{th} model run

$$\mathrm{NB}(d, \mathbf{x}^{(n)}) = g(d, \mathbf{x}_i^{(n)}) + \varepsilon^{(n)}. \tag{8}$$

The third key idea is that we can treat the N model outputs from the probabilistic sensitivity analysis $\{\mathrm{NB}(d, \mathbf{x}^{(1)}), \ldots, \mathrm{NB}(d, \mathbf{x}^{(N)})\}$ as "noisy" data through which we can learn about the functional form of $g(d, \mathbf{x}_i)$. Within Equation 8, we know $n = 1, \ldots, N$ values for the left-hand side $\mathrm{NB}(d, \mathbf{x}^{(n)})$ and the corresponding $n = 1, \ldots, N$ values of the $\mathbf{x}_i^{(n)}$, and therefore, we can think of this as a regression problem. However, we immediately recognize that the target function $g(d, \mathbf{x}_i)$ has unknown form, and we have no desire to impose any particular form. We could begin by fitting a standard linear model with power and interaction terms to model the nonlinearity between the net benefits and the inputs of interest, but we choose instead to adopt a more flexible nonparametric regression approach.

As an illustration, Figure 1 shows the results from a hypothetical PSA in which we plot the net benefit function, $\mathrm{NB}(x_1, x_2, x_3)$, against a single parameter of interest, x_1. The scatter of points suggests some kind of U-shaped function. The dashed line shows a nonparametric regression of $\mathrm{NB}(x_1, x_2, x_3)$ on x_1. This regression provides an estimate of the expected value $E_{X_2, X_3 | X_1 = x_1} \mathrm{NB}(x_1, X_2, X_3)$ as a function of x_1, that is, it provides the $g(d, \mathbf{x}_i)$ from Equation 8. In this particular illustrative model, the expectation $E_{X_2, X_3 | X_1 = x_1} \mathrm{NB}(x_1, X_2, X_3)$ can be obtained analytically (solid line), showing that the true expectation is very well estimated by the nonparametric regression.

Once we have obtained the regression function estimate, $\hat{g}(d, \mathbf{x})$, for each decision option in our economic model, we can proceed to calculating the partial EVPI. Evaluating $\hat{g}(d, \mathbf{x})$ at $\{\mathbf{x}_i^{(1)}, \ldots, \mathbf{x}_i^{(N)}\}$ gives us $\{\hat{g}(d, \mathbf{x}_i^{(1)}), \ldots, \hat{g}(d, \mathbf{x}_i^{(N)})\}$, which are the estimates of the conditional expectations that we require, and hence we can compute the partial EVPI by

$$\widehat{EVPI}(\mathbf{X}_i) = \frac{1}{N} \sum_{n=1}^{N} \max_d \hat{g}(d, \mathbf{x}_i^{(n)}) - \max_d \frac{1}{N} \sum_{n=1}^{N} \hat{g}(d, \mathbf{x}_i^{(n)}). \tag{9}$$

Note that we use $\max_d \frac{1}{N} \sum_{n=1}^{N} \hat{g}(d, \mathbf{x}_i^{(n)})$ as our Monte Carlo estimator of the second term in Equation 4 rather than $\max_d \frac{1}{N} \sum_{n=1}^{N} \mathrm{NB}(d, \mathbf{x}^{(n)})$. By choosing this as our estimator, we exploit the positive correlation between the 2 terms in Equation 9 and hence estimate the partial EVPI with increased precision.

We also note at this point that EVPI (calculated by any method) is invariant to the reexpression of net benefits as incremental net benefits, relative to some chosen baseline option (which is therefore defined as having an absolute net benefit of zero). This reduces the number of regression problems from D to $D - 1$.

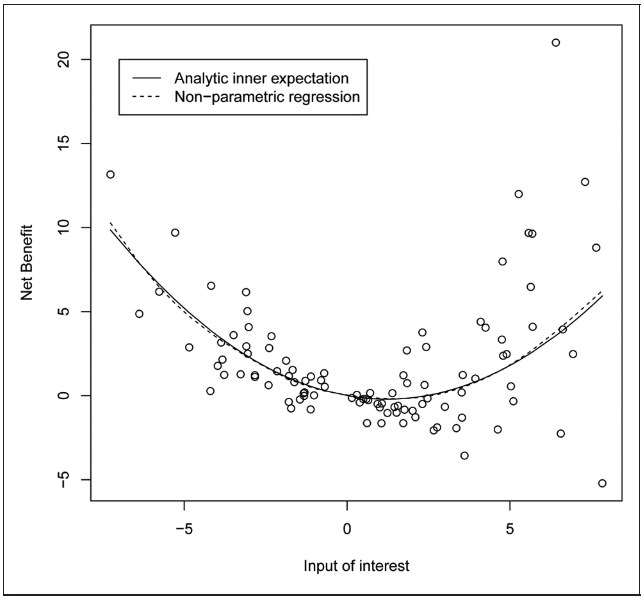

Figure 1. *Net benefit against single input parameter of interest for hypothetical model with 3 parameters.*

In the next sections, we give an overview of 2 particular nonparametric regression methods that are suitable in this context, Gaussian process regression and regression based on a Generalized Additive Model (GAM).

GAM Regression

When we adopt a GAM, we represent the unknown function $g(d, \mathbf{x}_i)$ as the sum of a set of smooth functions of the inputs. In the simplest form of GAM, we have

$$\begin{gathered} \mathrm{NB}(d, \mathbf{x}) = g(d, \mathbf{x}_i) + \varepsilon, \\ g(d, \mathbf{x}_i) = s_1(x_1) + \ldots + s_p(x_p), \end{gathered} \tag{10}$$

where each smoothing function $s_j(x_j)$ is a function of one of the $j = 1, \ldots, p$ model input parameters of interest, and ε is a mean-zero Normally distributed error with constant variance. For an introduction to GAM models, see Hastie and Tibshirani[12] or Wood.[13]

The usual choice for the smoothing functions is some form of spline, a common choice being the cubic spline. A cubic spline represents an arbitrary smooth function as a series of short cubic polynomials joined piecewise, as shown in Figure 2.

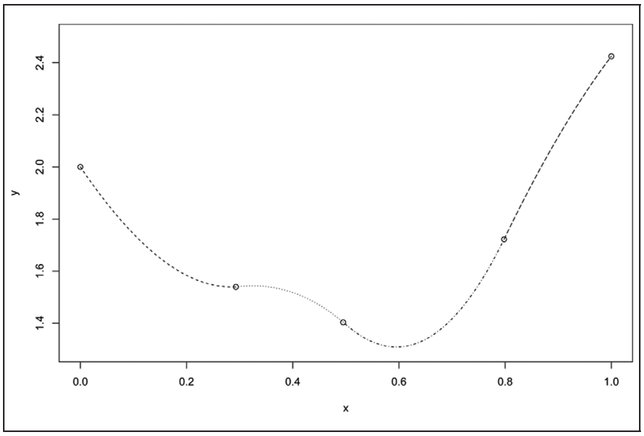

Figure 2. A cubic spline, showing the piecewise construction from 4 sections of cubic polynomial, each with different coefficients.

The cubic spline shown in Figure 2 can also be expressed as the weighted sum of a series of polynomial basis functions, $b_l(x)$, that take values of x across the whole range of x, rather than values in short segments of the range of x (this builds up the spline function in a manner similar to the way in which an arbitrary sound wave can be built up from sine waves of increasing frequency). This allows us to write

$$y = s(x) = \sum_{l=1}^{L} \beta_l b_l(x), \tag{11}$$

for some basis dimension L. The basis dimension controls the degree to which the spline can be ''wiggly'' (we can loosely think of this as akin to determining the number of segments in Figure 2). The basis functions themselves tend to be cumbersome to write out, and the reader is referred to Wood for further details.[13]

By expressing our unknown function (Equation 10) in the same way, we get

$$g(d, \mathbf{x}_i) = \sum_{l_1=1}^{L_1} \beta_{l_1} b_{l_1}(x_1) + \ldots + \sum_{l=1_p}^{L_p} \beta_{l_p} b_{l_p}(x_p). \tag{12}$$

Estimation of the model coefficients is typically via penalized maximum likelihood, in which the penalties are designed to suppress overly wiggly estimates that would result in overfitting. The choice of basis dimension for each spline is usually not important as long as it is sufficiently large to avoid constraining the spline to be overly inflexible (we found any any dimension greater than 3 to be sufficient). In practice, the software in which GAM is implemented makes the choice of basis dimension for each spline automatically.

Although the simple additive model in Equation 10 performs well in many situations, it will not adequately capture interactions between the input parameters of interest that may be a feature of the health economic model. To model interactions, we must include multivariate smoothing functions in our GAM model specification. So, for example, if we expect there to be interactions between inputs x_1 and x_2, then we would specify the model

$$g_d(\mathbf{x}_i) = s_1(x_1, x_2) + s_2(x_3), \ldots, s_{p-1}(x_p). \tag{13}$$

The multivariate smoothing function s_1 is built up using a tensor product construction, which results in the spline's being the sum of all multiplicative combinations of the basis functions for each variable,

$$s_1(x_1, x_2) = \sum_{l_1=1}^{L_1} \sum_{l_2=1}^{L_2} \beta_{l_1 l_2} b_{l_1}(x_1) b_{l_2}(x_2). \tag{14}$$

Modeling a large number of potential interactions does therefore have a cost. Given m inputs that are expected to interact in the economic model, and assuming the same basis dimension, n, for each input variable, the GAM model must estimate n^m coefficients. If n^m approaches the size of the PSA sample, then the GAM method will break down. This is one motivation for the more flexible Gaussian process regression approach described in the next section.

After estimating the GAM model parameters and hence obtaining $\hat{g}(d, \mathbf{x})$, we can evaluate $\hat{g}(d, \mathbf{x})$ at the PSA inputs to give $\{g(d, \mathbf{x}_i^{(1)}), \ldots, g(d, \mathbf{x}_i^{(N)})\}$ and therefore the partial EVPI via Equation 9. The code in Box 1 illustrates the simplicity of the GAM regression approach using the mgcv package in R.[14] In the example, there are 2 decision options, with the vector object INB holding the incremental net benefits from the PSA. The PSA samples from the 2 parameters of interest are held in vector objects x5 and x14. We assume the parameters do not interact in the model. If they did, we would simply replace the model formula INB ~ s(x5) + s(x14) with the tensor product multivariate specification INB ~ te(x5,x14).

Box 1 Example R Code for Estimating Partial Expected Value of Perfect Information via Generalized Additive Model Regression

```
library(mgcv)
model <- gam(INB ~ s(x5) + s(x14))
g.hat <- model$fitted
partial.evpi <- mean(pmax(0,g.hat)) -
  max(0,mean(g.hat))
```

A method for estimating the standard error of the GAM-based approximation of the partial EVPI is given in Appendix A. R functions for computing the partial EVPI via GAM and its standard error are available at http://www.shef.ac.uk/scharr/sections/ph/staff/profiles/mark.

Gaussian Process Regression

The Gaussian process is a highly flexible representation of an unknown function, in our case $g(d, \mathbf{x}_i)$, that again requires no parametric assumptions regarding functional form.[15] When we model the function $g(d, \mathbf{x}_i)$ as a Gaussian process, we assume that we can represent the unknown values of the function evaluated at the PSA inputs, $\{g(d, \mathbf{x}_i^{(1)}), \ldots, g(d, \mathbf{x}_i^{(N)})\}$, via a multivariate Normal distribution. To be more precise, we are representing our beliefs about the function using the multivariate Normal distribution. The function itself is unknown. We will therefore require a method for specifying the mean, variance, and covariance of the distribution that specifies our beliefs about the unknown function $g(d, \mathbf{x}_i)$ given the PSA values $\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$ and $\{\text{NB}(d, \mathbf{x}^{(1)}), \ldots, \text{NB}(d, \mathbf{x}^{(N)})\}$ that we have observed (sampled).

It is very important to note that by representing the unknown function $g(d, \mathbf{x}_i)$ as a Gaussian process, we do not require that the model input parameters $\mathbf{x}_i$ are Normally distributed (Gaussian) or that the net benefits $\text{NB}(d, \mathbf{x})$ are Gaussian. In practice, the main requirement is that $g(d, \mathbf{x}_i)$ is a smooth function of its inputs in the sense that for any $\mathbf{x}_i^{(n)}$ and $\mathbf{x}_i^{(m)}$ that are close, $g(d, \mathbf{x}_i^{(n)})$ and $g(d, \mathbf{x}_i^{(m)})$ are also close. This is a weak requirement and likely to hold in most health economic models because costs and health benefits (e.g., quality-adjusted life-years [QALYs]) are usually continuous functions of the uncertain model input parameters.

Until now, the use of the Gaussian process in health economics has been rare and restricted to the modeling of the net benefit function in the context of a computationally expensive model.[16–18] For a practical introduction to building Gaussian process models, see the Managing Uncertainty in Complex Models toolkit at mucm.aston.ac.uk/MUCM/MUCMToolkit/.

Gaussian Process Regression Model Specification. Recall that our PSA sample consists of N input vectors $\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$ and N corresponding net benefits $\{\text{NB}(d, \mathbf{x}^{(1)}), \ldots, \text{NB}(d, \mathbf{x}^{(N)})\}$ for each decision option d. For each model run $n = 1, \ldots, N$, we have $\text{NB}(d, \mathbf{x}^{(n)}) = g(d, \mathbf{x}_i^{(n)}) + \varepsilon^{(n)}$ from Equation 8. We assume that the vector of unknown values of the function $g(d, \mathbf{x}_i)$ evaluated at the PSA input values, $\{g(d, \mathbf{x}_i^{(1)}), \ldots, g(d, \mathbf{x}_i^{(N)})\}$, jointly follows a multivariate Normal distribution,

$$\{g(d, \mathbf{x}_i^{(1)}), \ldots, g(d, \mathbf{x}_i^{(N)})\} \sim N(H\boldsymbol{\beta}, \sigma^2\Sigma). \tag{15}$$

The mean of the distribution Hβ is a vector of length N and is the matrix product of a design matrix

$$H = \begin{pmatrix} 1 & x_1^{(1)} & \ldots & x_p^{(1)} \\ 1 & x_1^{(2)} & \ldots & x_p^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_1^{(N)} & \ldots & x_p^{(N)} \end{pmatrix} \tag{16}$$

of size $N \times q$, (where $q = p + 1$), and a vector of regressors $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_q)$. The covariance matrix is a product of a scalar variance term σ^2 and a correlation matrix Σ of size $N \times N$.

We require that the correlation matrix Σ describes the smoothness of the function $g(d, \mathbf{x}_i)$ with respect to each input parameter of interest in the set of p inputs that makes up $\mathbf{x}_i$. We therefore define the $\{n, m\}^{th}$ element of Σ to be a function of the p input parameters of interest in the following way,

$$\Sigma^{(n,m)} = \exp\left[-\sum_{j=1}^{p} \{(x_j^{(n)} - x_j^{(m)})/\delta_j\}^2\right]. \tag{17}$$

The superscripts (n) and (m) denote arbitrary runs in the PSA sample, and j indexes the p parameters of interest that make up $\mathbf{x}_i$. The correlation length hyperparameters δ_j describe the smoothness of $g(d, \mathbf{x}_i)$ with respect to each parameter of interest and are estimated from the PSA sample as described below.

Note that the form of the correlation function ensures that diagonal entries in the matrix Σ are equal to 1 as they should be for a valid correlation matrix. To see why, observe that on the diagonal, we have $\Sigma^{(n,n)} = \exp\left[-\sum_{j=1}^{p} \{(x_j^{(n)} - x_j^{(n)})/\delta_j\}^2\right] = \exp(0) = 1.$ The value of $\Sigma^{(n,m)}$, and therefore the correlation between $g(\mathbf{x}_i^{(n)})$ and $g(\mathbf{x}_i^{(m)})$, decreases toward zero as the distance between $\mathbf{x}_i^{(n)}$ and $\mathbf{x}_i^{(m)}$ increases, with the values of δ_j controlling how fast this decay to zero occurs.

Finally, we require a method for learning about $\boldsymbol{\beta}$, σ^2, and δ_j from the net benefits, $\text{NB}(d, \mathbf{x})$. To do this, we must link the Gaussian process model for $g(d, \mathbf{x}_i)$ to the net benefits, $\text{NB}(d, \mathbf{x})$. Recalling Equation 8, the net benefit obtained on the n^{th} PSA model run, $\mathrm{NB}(d,\mathbf{x}^{(n)})$, is considered to be the sum of $g(d,\mathbf{x}_i^{(n)})$ and a noise term $\varepsilon^{(n)}$, which implies that we can write

$$\{\mathrm{NB}(d,\mathbf{x}^{(1)}),\ldots,\mathrm{NB}(d,\mathbf{x}^{(N)})\}\sim N\{H\boldsymbol{\beta},\sigma^2(\Sigma+\nu I)\}, \quad (18)$$

where I is the identity matrix of size N and ν is a ''nugget'' term that controls the variance of a Normally distributed mean zero, constant variance noise term ε.[19,20] For compactness of notation in the remainder of the article, we write $\Sigma^*=\Sigma+\nu I$ and define the vectors $\mathbf{nb}_d=\{\mathrm{NB}(d,\mathbf{x}^{(1)}),\ldots,\mathrm{NB}(d,\mathbf{x}^{(N)})\}$, $\mathbf{g}_d=\{g(d,\mathbf{x}_i^{(1)}),\ldots,g(d,\mathbf{x}_i^{(N)})\}$, and $\hat{\mathbf{g}}_d=\{\hat{g}(d,\mathbf{x}_i^{(1)}),\ldots,\hat{g}(d,\mathbf{x}_i^{(N)})\}$.

Estimation of Hyperparameters β, σ², δj, and ν. The first step is to estimate the correlation lengths δ_j and the nugget term ν from the PSA sample. The most straightforward approach is to find the values $\widehat{\delta}_j$ and $\widehat{\nu}$ that maximize the joint posterior density of δ_j and ν given the net benefits $\mathbf{nb_d}$. This requires numerical methods, and details are given in Appendix B. An R function is available at http://www.shef.ac.uk/scharr/sections/ph/staff/profiles/mark. Given $\widehat{\delta}_j$ and $\widehat{\nu}$ (and hence $\widehat{\Sigma^*}$), the posterior mean of $\boldsymbol{\beta}$, which can be derived analytically, is

$$\hat{\boldsymbol{\beta}}=(H^{\mathrm{T}}\Sigma^{*-1}H)^{-1}H^{\mathrm{T}}\Sigma^{*-1}\mathbf{nb}_d \quad (19)$$

and the posterior mean of σ^2 is

$$\widehat{\sigma}^2=\frac{(\mathbf{nb}_d-H\hat{\boldsymbol{\beta}})^{\mathrm{T}}\Sigma^{*-1}(\mathbf{nb}_d-H\hat{\boldsymbol{\beta}})}{n-q-2}. \quad (20)$$

Estimation of $g(d,\mathbf{x}_i)$. Once we have determined Σ^* and $\hat{\beta}$, we can use the properties of the Normal distribution to obtain the expected value of $\mathbf{g}_d$ conditional on the net benefits $\mathbf{nb}_d$,

$$\hat{\mathbf{g}}_d=H\hat{\boldsymbol{\beta}}+\Sigma\Sigma^{*-1}(\mathbf{nb}_d-H\hat{\boldsymbol{\beta}}). \quad (21)$$

The components of $\hat{\mathbf{g}}_d$ are $\{\hat{g}(d,\mathbf{x}_i^{(1)}),\ldots,\hat{g}(d,\mathbf{x}_i^{(N)})\}$ and hence can be plugged into Equation 9 to give the partial EVPI. A method for estimating the standard error of the Gaussian process regression approximation for the partial EVPI is given in Appendix A. The R code for computing the Gaussian process regression-based partial EVPI and its standard error is available at http://www.shef.ac.uk/scharr/sections/ph/staff/profiles/mark.

Implementation Issues and Regression Diagnostics

We recommended above that net benefits are expressed as incremental net benefits, relative to a chosen baseline option. This not only reduces the number of regression problems from D to $D-1$ but also improves numerical stability, particularly for the Gaussian process method. For the same reason, we also suggest that, for the Gaussian process method, the input parameters of interest are each scaled to lie in the $[0,1]$ interval. This ensures that the smoothness parameters δ_j are estimated on a common scale. EVPI is invariant to linear rescaling of the input parameters.

For both Gaussian process and GAM models, examination of the residuals is useful for assessing the robustness of assumptions. A plot of residuals (i.e., $\mathbf{y}_d-\hat{\mathbf{g}}_d$) against fitted values ($\hat{\mathbf{g}}_d$) allows assessment of the mean-variance relationship and will highlight deviation from the assumption of constant variance. A Normal quantile-quantile plot of residuals will show deviation from the assumption of Normality of the residuals.

CASE STUDIES

Case Study 1: A Simple Decision Tree Model with Correlated Inputs

Case study 1 is based on a hypothetical decision tree model previously used for illustrative purposes.[5,7,9,10] The model predicts net benefit, $\mathrm{NB}(d,\mathbf{x})$, under 2 decision options ($d=1,2$) and can be written in sum product form as

$$\mathrm{NB}(1,\mathbf{x})=\lambda(x_5x_6x_7+x_8x_9x_{10})-(x_1+x_2x_3x_4), \quad (22)$$

$$\mathrm{NB}(2,\mathbf{x})=\lambda(x_{14}x_{15}x_{16}+x_{17}x_{18}x_{19})-(x_{11}+x_{12}x_{13}x_4), \quad (23)$$

where $x_1,\ldots,x_{19}$ are sampled realizations of the uncertain input parameters $X_1,\ldots,X_{19}$ listed in Table 1, and the willingness to pay for 1 unit of health output in QALYs is $\lambda=10,000/QALY$. Note that some components of $\mathbf{x}=(x_1,\ldots,x_{19})$ are redundant in $\mathrm{NB}(d,\mathbf{x})$ for each d.

We assume that our uncertainty about the inputs can be represented by a multivariate Normal distribution, with X_5,X_7,X_{14}, and X_{16} all pairwise correlated with a correlation coefficient of 0.6, and with X_6 and x_{15} correlated with a correlation coefficient of 0.6. All other inputs were assumed independent. In a simple

Table 1 Summary of Means and Standard Deviations for Case Study Model Parameters

Parameter	Mean (*sd*)	
	$d = 1$	$d = 2$
Cost of drug (X_1,X_{11}), £	1000 (1)	1500 (1)
% Admissions (X_2,X_{12})	10 (2)	8 (2)
Days in hospital (X_3,X_{13})	5.20 (1.00)	6.10 (1.00)
Cost per day (X_4), £	400 (200)	400 (200)
% Responding (X_5,X_{14})	70 (10)	80 (10)
Change in utility if respond (X_6,X_{15})	0.30 (0.10)	0.30 (0.05)
Duration of response (X_7,X_{16}), years	3.0 (0.5)	3.0 (1.0)
% Side effects (X_8,X_{17})	25 (10)	20 (5)
Change in utility if side effect (X_9,X_{18})	–0.10 (0.02)	–0.10 (0.02)
Duration of side effect (X_{10},X_{19}), years	0.50 (0.20)	0.50 (0.20)

sum product form model, the assumption of multivariate Normality allows us to compute the inner conditional expectation analytically.

We define 3 parameter sets of interest: set 1 comprising effectiveness parameters X_5 and X_{14}, representing information that could be gained from a trial; set 2 comprising effectiveness and utility parameters X_5,X_6,X_{14} and X_{15}, representing information that could be gained from a trial that also collected utility data; and set 3 comprising duration of response parameters X_7 and X_{16}, representing information that could be gained from the long-term follow-up of trial participants.

Although the case study model is computationally cheap to evaluate, we assume that we are in a position of being able to evaluate the model only 10 000 times. Given this limitation, we calculated partial EVPI using 3 methods. First, we calculated the partial EVPI for each parameter set using a single-loop Monte Carlo approximation for the outer expectation in the first term of the right-hand side of Equation 4 with 10 000 samples from the distribution of the parameters of interest, an analytic solution to the inner conditional expectation, and hence 10 000 model runs. Next, we calculated the partial EVPI values using the standard 2-level Monte Carlo approach with 3 different sets of J inner loop samples and K outer loop samples, where J × K = 10 000 model runs in total (see Table 2 for values of J and K). Third, we computed the partial EVPI values using the GAM regression method with a total of 10 000 PSA samples. Finally, we computed the partial EVPI values using the Gaussian process regression method with the same 10 000 PSA samples.

We compared values with a gold standard measure of partial EVPI calculated using the analytic solution to the inner conditional expectation, and 10^7 outer loop samples. Standard errors for the 2-level Monte Carlo partial EVPI estimates were obtained using the method given in Appendix A. Estimates of partial EVPI using the 2-level Monte Carlo method are upwardly biased for small values of J, due to the maximization step. The estimates of upward bias were obtained using the method presented in Oakley and others[9] (see Appendix A).

For each method, we report the mean time taken to compute the partial EVPI for the 3 parameter sets of interest.

Results for Case Study 1. Regression diagnostic plots for the Gaussian process and GAM models are shown in Figure 3. A random subset of 500 points is shown on each plot. First of note is, for each parameter set, the similarity in the pattern of residuals between the Gaussian process model and the GAM model (reflecting the similarity in estimates of $\mathbf{g}$). In each case, the plots of residuals against fitted values show no worrying heteroscedasticity, and the residual Normal Q-Q plots show no gross deviation from Normality.

Figure 4 shows the values of $\hat{\mathbf{g}}$ obtained via the regression methods against the analytically calculated values of $\mathbf{g}$. Good agreement is seen over the whole range.

Table 2 shows the estimated partial EVPI values for the 3 sets of parameters of interest. The overall EVPI for all 19 parameters is £1047. The top line shows the gold standard estimates, obtained by generating 10^7 samples from the joint distribution of the inputs of interest and then analytically calculating the expected net benefits for each decision option, conditional on these sampled values. The standard errors of the gold standard estimates are small. When we restrict ourselves to only 10 000 model evaluations,

Table 2 Partial Expected Value of Perfect Information (EVPI) Values and Timings for Case Study 1

Sample Size			Partial EVPI (SE; Upward Bias), £			
Outer Loop	**Inner Loop**	**Total**	**Parameter Set 1** $\{X_5, X_{14}\}$	**Parameter Set 2** $\{X_5, X_6, X_{14}, X_{15}\}$	**Parameter Set 3** $\{X_7, X_{16}\}$	**Mean Time**
One-level Monte Carlo[a]						
10^7	—	10^7	247.95 (0.14; unbiased)	840.84 (0.27; unbiased)	536.28 (0.31; unbiased)	1.3 h
One-level Monte Carlo[b]						
10^4	—	10^4	255.15 (4.41; unbiased)	845.73 (8.53; unbiased)	534.80 (9.01; unbiased)	1.0 s
Two-level Monte Carlo[b]						
10^1	10^3	10^4	232.80 (140.28; 1.42)	474.22 (452.56; 0.17)	301.55 (269.00; 0.42)	0.04 s
10^2	10^2	10^4	222.75 (49.34; 14.28)	796.77 (143.94; 1.83)	501.51 (86.71; 5.88)	0.07 s
10^3	10^1	10^4	351.92 (25.20; 130.69)	909.06 (47.95; 20.35)	583.05 (32.46; 62.99)	0.5 s
Two-level Monte Carlo[c]						
10^4	10^3	10^7	243.61 (4.37; 1.34)	834.73 (13.67; 0.25)	552.97 (9.13; 0.74)	34 s
Gaussian process regression[b]						
10^4	—	10^4	234.44 (17.02, 0.82)	830.48 (11.44, 0.65)	541.13 (15.76, 0.49)	170 s
GAM regression[b]						
10^4	—	10^4	234.52 (16.24, 1.76)	832.19 (10.48, 0.25)	540.50 (15.49, 0.51)	0.9 s

a. Reference gold standard.
b. Model runs restricted to 10^4.
c. *J* and *K* chosen to achieve SE and bias of the same order of magnitude as the regression estimates.

but again use the analytic solution to the conditional expectation, the standard errors are unsurprisingly larger. The estimates are still unbiased. In contrast, estimates obtained via the 2-level Monte Carlo approach are biased due to the maximization over quantities that are subject to sampling variability.[9] When restricted to 10 000 model evaluations, there is a clear tradeoff between bias and variance when using the 2-level method, with small values of the inner loop resulting in considerable upward bias.

In comparison, the regression-based estimates all have lower variance than any of the 2-level Monte Carlo estimates when model runs are restricted to 10 000. The upward bias due to the maximization in the first term of Equation 9 is small in each case and comparable with that obtained by the 2-level Monte Carlo method with 1000 inner loop samples. To achieve a similar level of bias and variance to that obtained using the regression method with 10^4 PSA samples, the 2-level Monte Carlo would require approximately 10^7 model runs.

The computational cost of obtaining the gold standard estimate is greatest, because of the large sample size. The 2-level Monte Carlo method is fast in this example because of the simplicity of the model but will typically be slower and will increase as the computational complexity of the model increases. In contrast, the speeds of the Gaussian process and GAM methods are independent of the computational complexity of the model because the model itself is not evaluated during the regression fitting process. The GAM method takes less than 1 s with a PSA sample size of 10^4, whereas the Gaussian process method takes approximately 3 min.

Case Study 2: Three State Markov Model

Case study 2 is an extension of the case study 1 model that incorporates a 20-time cycle Markov model for the response to each intervention. The parameters for mean duration of response (x_7 and x_{16}) are replaced with Markov models of natural history of response to each drug with health states ''responding,'' ''not responding,'' and ''dead.'' The model is

$$\mathrm{NB}(1,\mathbf{x}) = \lambda\left\{\sum_{n=1}^{20}\left(\mathbf{S}_1^{\mathrm{T}}\mathbf{M}_1^n\mathbf{U}_1\right) + x_8x_9x_{10}\right\} - (x_1 + x_2x_3x_4), \tag{24}$$

$$\mathrm{NB}(2,\mathbf{x}) = \lambda\left\{\sum_{n=1}^{20}\left(\mathbf{S}_2^{\mathrm{T}}\mathbf{M}_2^n\mathbf{U}_2\right) + x_{17}x_{18}x_{19}\right\} - (x_{11} + x_{12}x_{13}x_4), \tag{25}$$

where the vectors $\mathbf{S}_d$ and $\mathbf{U}_d$ are defined as $\mathbf{S}_1 = (x_5, 1-x_5, 0)^{\mathrm{T}}$, $\mathbf{S}_2 = (x_{14}, 1-x_{14}, 0)^{\mathrm{T}}$, $\mathbf{U}_1 = (x_6, 0, 0)^{\mathrm{T}}$, and $\mathbf{U}_2 = (x_{15}, 0, 0)^{\mathrm{T}}$and where the transition matrices are defined as

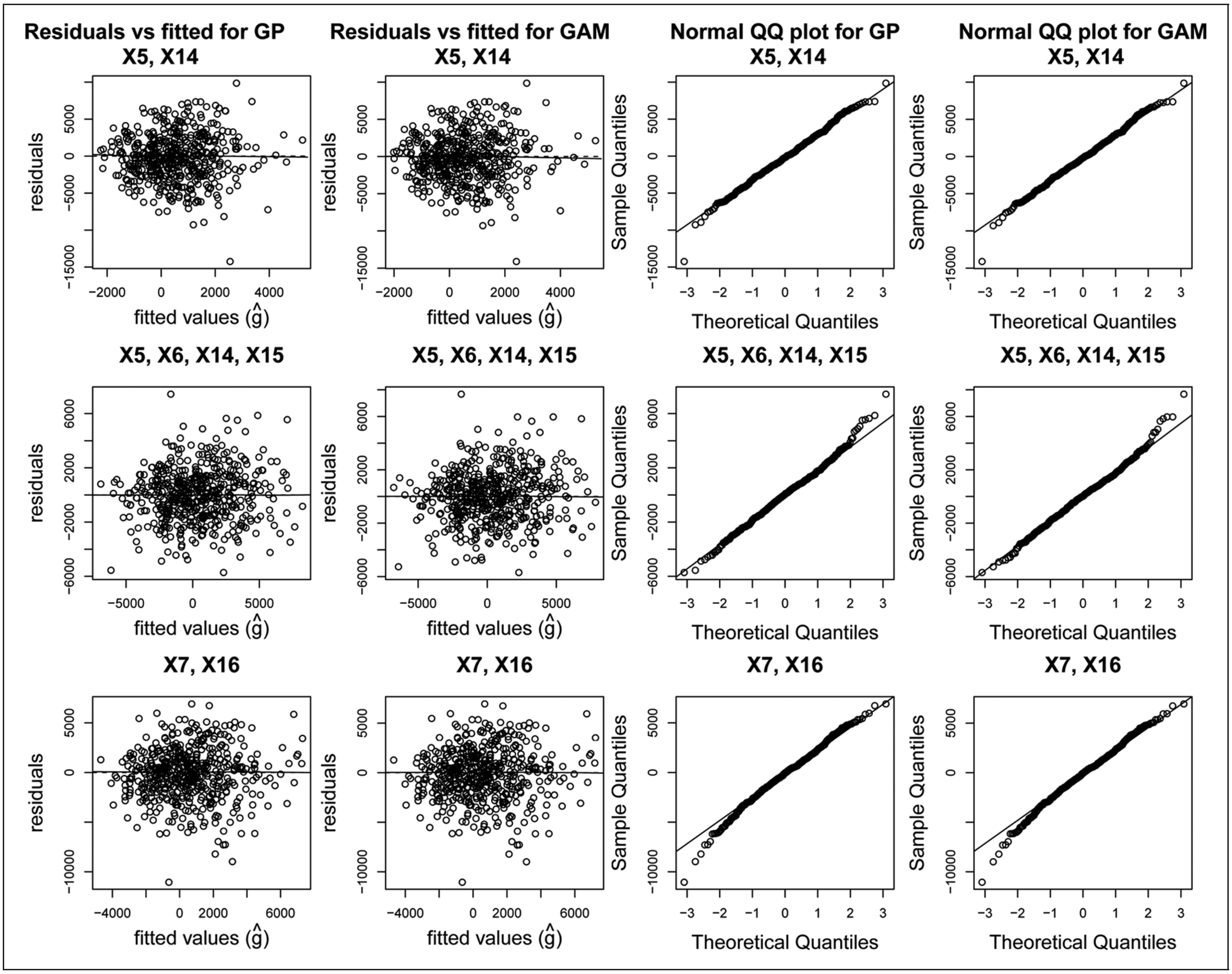

Figure 3. Regression diagnostic plots for case study 1.

$$\mathbf{M}_1 = \begin{pmatrix} x_{20} & x_{21} & x_{22} \\ x_{23} & x_{24} & x_{25} \\ 0 & 0 & 1 \end{pmatrix} \text{ and } \mathbf{M}_2 = \begin{pmatrix} x_{26} & x_{27} & x_{28} \\ x_{29} & x_{30} & x_{31} \\ 0 & 0 & 1 \end{pmatrix}. \quad (26)$$

Uncertainty regarding the transition matrix parameters (X_{20} to X_{31}) was expressed using Dirichlet distributions with $(X_{20}, X_{21}, X_{22}) \sim$ Dirichlet(70,40,10); $(X_{23}, X_{24}, X_{25}) \sim$ Dirichlet(10,100,20); $(X_{26}, X_{27}, X_{28}) \sim$ Dirichlet(70,40,10); and $(X_{29}, X_{30}, X_{31}) \sim$ Dirichlet(10,100,20). Means and standard deviations for the remaining input parameters are as for case study 1 (Table 1), but now instead of assuming Normality for all parameters, we expressed our uncertainty about $X_2, X_5, X_8, X_{12}, X_{14}$ and X_{17} using Beta distributions and our uncertainty about X_3, X_4, X_{10}, X_{13} and X_{19} using Gamma distributions. In contrast with case study 1, it is assumed that each input parameter X_1 to X_{19} is independent of all other parameters in the model.

We again defined 3 parameter sets of interest: set 1 comprising effectiveness parameters X_5 and X_{14}, representing information that could be gained from a trial; set 2 comprising effectiveness and utility parameters X_5, X_6, X_{14} and X_{15}, representing information that could be gained from a trial that also collected utility data; and set 3 comprising the transition matrix parameters X_{20} to X_{31}, representing information that could be gained from the long-term follow-up of trial participants.

Results for Case Study 2. A similar pattern of results is seen for case study 2 as for case study 1.

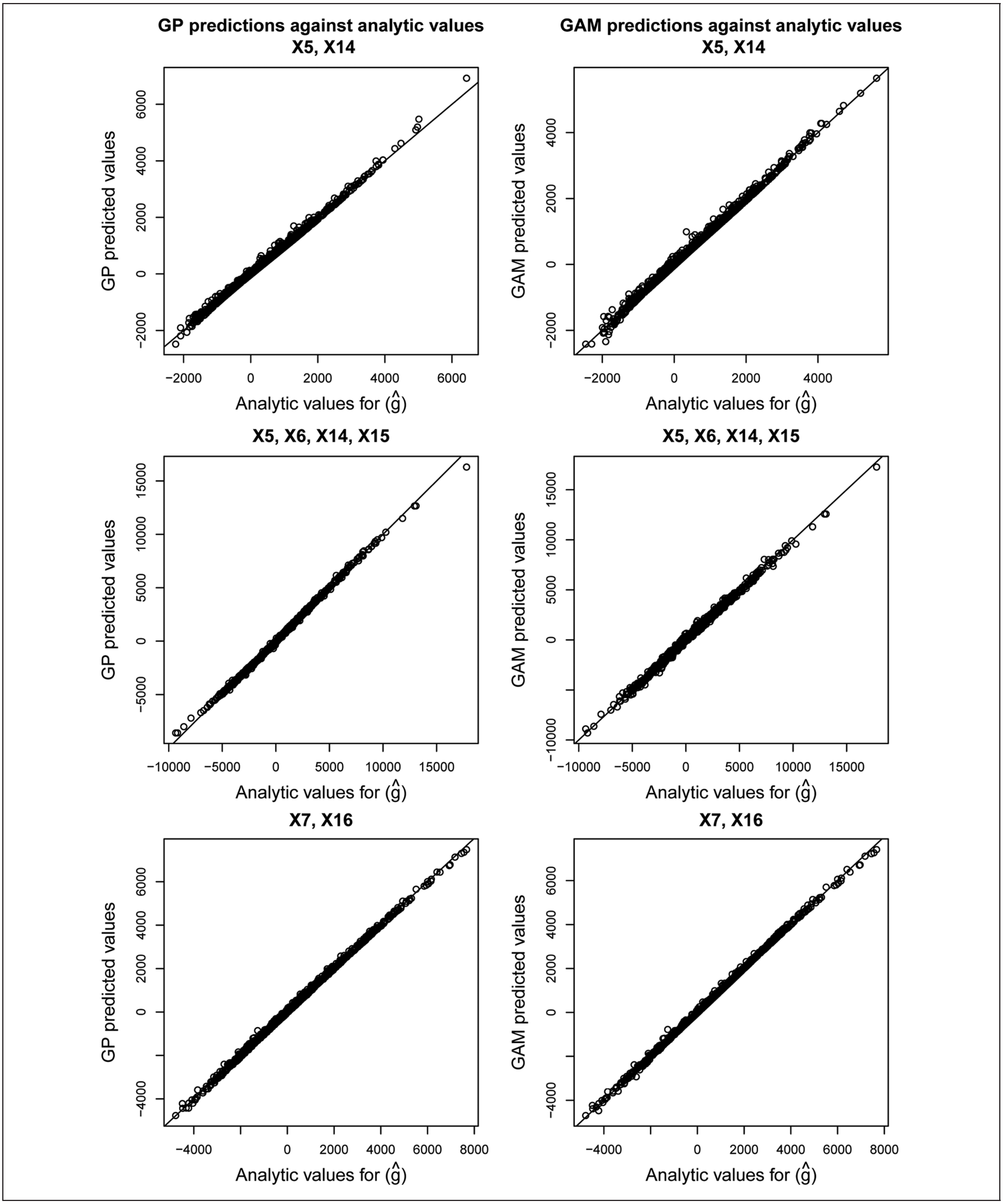

Figure 4. *Gaussian process and Generalized Additive Model regression predictions versus analytic values for case study 1.*

Regression diagnostic plots shown in Figure 5 are similar in character to the those obtained in the first case study, and again, no worrying departures from the model assumptions are indicated.

Figure 6 shows the values of $\hat{\mathbf{g}}$ calculated by the regression methods against the corresponding values obtained by the 2-level Monte Carlo method with 10^8 model runs (defined as our gold standard in this case). Very good agreement is seen over the whole range of $\mathbf{g}$ in each case.

Table 3 shows the estimated partial EVPI values. The overall EVPI is £775. Standard errors for the gold standard 2-level Monte Carlo estimates with 10^8 model runs are small, as are the values of the upward bias. When the number of model evaluations is restricted to 10^4, the regression methods perform considerably better than the 2-level Monte Carlo method, resulting in estimates that have both minimal upward bias and substantially greater precision. To achieve a similar level of bias and variance to that obtained using the regression method with 10^4 PSA samples, the 2-level Monte Carlo would require approximately 10^7 model runs.

With a PSA sample size of 10^4, the GAM takes approximately 1 s and the Gaussian process takes approximately 3 min. In contrast, the 2-level Monte Carlo method with 10^7 model runs takes 1.8 h.

DISCUSSION

Main Result and Implications

The regression-based approach we propose requires only the single set of model evaluations that is generated in a standard probabilistic sensitivity analysis to calculate partial EVPI for any set of inputs. It leads to a considerable gain in precision over the 2-level Monte Carlo method with the same number of model runs while retaining an acceptably small upward bias. The GAM method in particular is straightforward to implement in the freely available software R, thus allowing an analyst to compute partial EVPI for any subset of input parameters quickly and with relative ease.

The regression method allows the complete separation of the EVPI calculation step from the model evaluation step, which may be particularly useful when the model has been built using specialist software (e.g., for discrete event simulation) that does not allow easy implementation of the EVPI step or where those who wish to compute the EVPI do not own (and therefore cannot directly evaluate) the model. The method has the particular advantage that, even in the case of correlated inputs, only the joint distribution of inputs is required. This is in contrast to the 2-level Monte Carlo approach in which we are required to sample values from $p(\mathbf{X}_{-i}|\mathbf{X}_i = \mathbf{x}_i)$, the conditional distribution of the remaining parameters given some sampled parameter vector of interest, a process that an analyst could find challenging without the necessary statistical training.

In terms of computational speed, the regression methods are fast. We see 2 particular scenarios in which this will be useful: when the analyst is faced with a slow patient-level simulation model and in the case in which the partial EVPI calculation would require computationally demanding MCMC sampling under the 2-level scheme.

For health economic decision analysts, the key implication of the nonparametric regression approach is that the computation of partial EVPI has become tractable for any decision problem. We hope that the computation of partial EVPI values now becomes standard practice, and we urge those who write guidance on good modeling practice to promote the routine reporting of EVPI values.

Limitations

There are some limitations of the regression approaches. In general, the GAM method will be more straightforward to implement because of the easy availability of software (e.g., the mgcv package in R). However, if the set of input parameters for which we wish to calculate partial EVPI is moderately large (greater than 6 or so), and if it is expected that those parameters will jointly interact (nonadditively) within the economic model, then it is likely that the number of GAM model parameters that need to be estimated will exceed the number of data points, causing the method to fail. In this case, we would recommend using the Gaussian process approach.

Although the Gaussian process method is relatively easy to implement in R using the functions available at http://www.shef.ac.uk/scharr/sections/ph/staff/profiles/mark, the estimation of the hyperparameters requires numerical optimization, which will be slow if the number of parameters of interest is large. This optimization is not a black box procedure, and as with other numerical methods such as MCMC, the onus is on the user to ensure that convergence is achieved. Second, the Gaussian process method incurs the computational cost of inverting the $N \times N$ matrix Σ^*, which increases in proportion to N^3, where N is the number of PSA samples. This

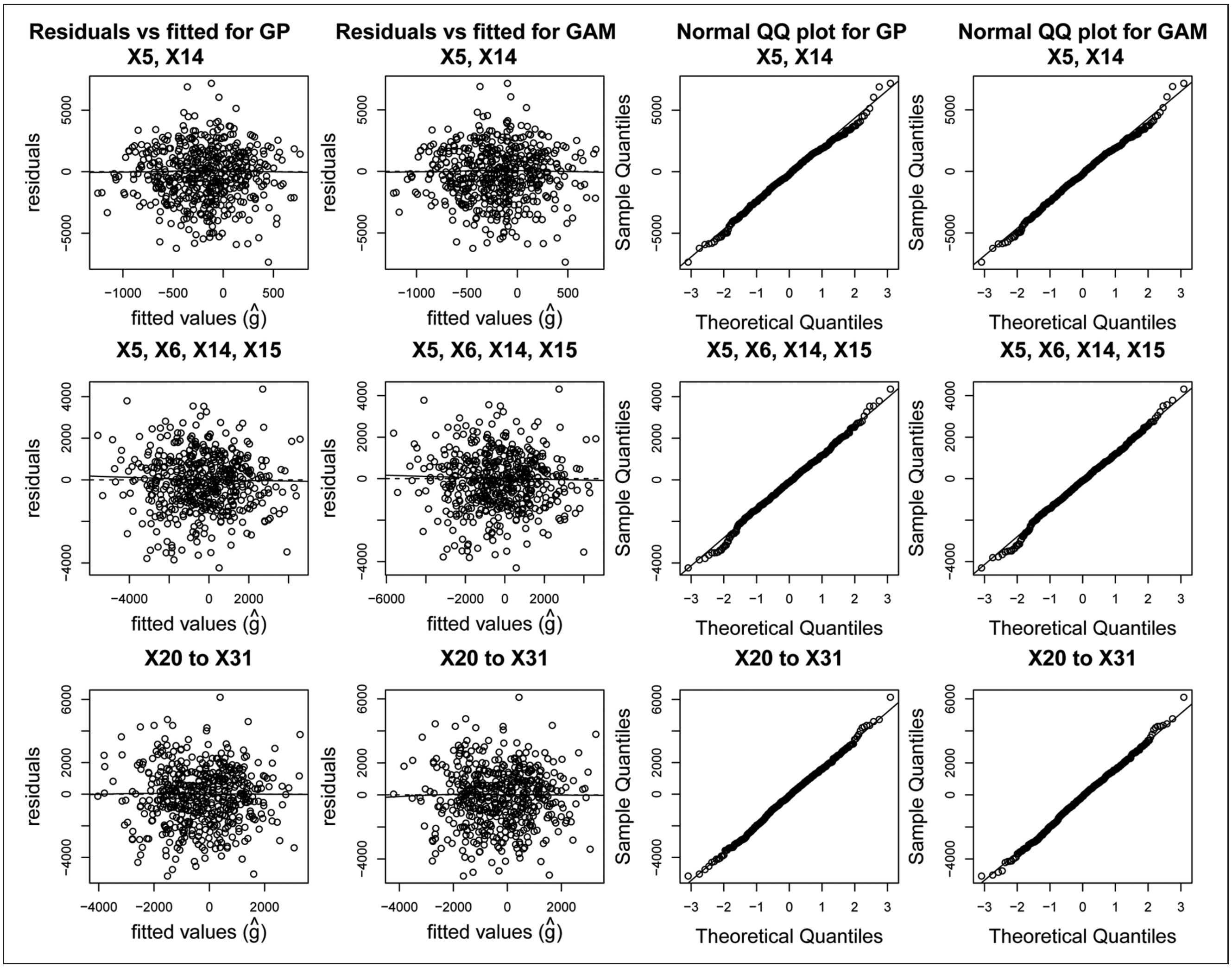

Figure 5. Regression diagnostic plots for case study 2.

places a practical limit on the size of N (currently of the order of tens of thousands), which in turn limits the precision that can be achieved with the Gaussian process method. Finally, the use of the Gaussian process currently requires somewhat more work on the part of the analyst than the GAM approach, even with the functions that we have made available.

Using the Method in Patient-Level Models

In our introduction, we presented a typical scenario in which obtaining partial EVPI via 2-level Monte Carlo was likely to be computationally prohibitive due to the requirement to sample many thousands of patients within each evaluation of the inner loop.

Partial EVPI via the regression method is calculated for a patient-level model in the same manner as it is for a cohort model (i.e., by regressing the PSA sample net benefits on the parameters of interest). We briefly recap here the computation of a PSA for a patient-level model. This is a 2-level process whereby samples are drawn from the PSA level (i.e., population level) parameters in an outer loop, and then, conditional on these samples, individual patients are sampled in an inner loop. The purpose of sampling individual patients is to average over heterogeneity (and/or uncertainty) at the patient level for each sample of population-level input parameters. Convergence is achieved when the patient-level sample size is large enough that, given some arbitrary sample from the PSA (population)–level parameters,

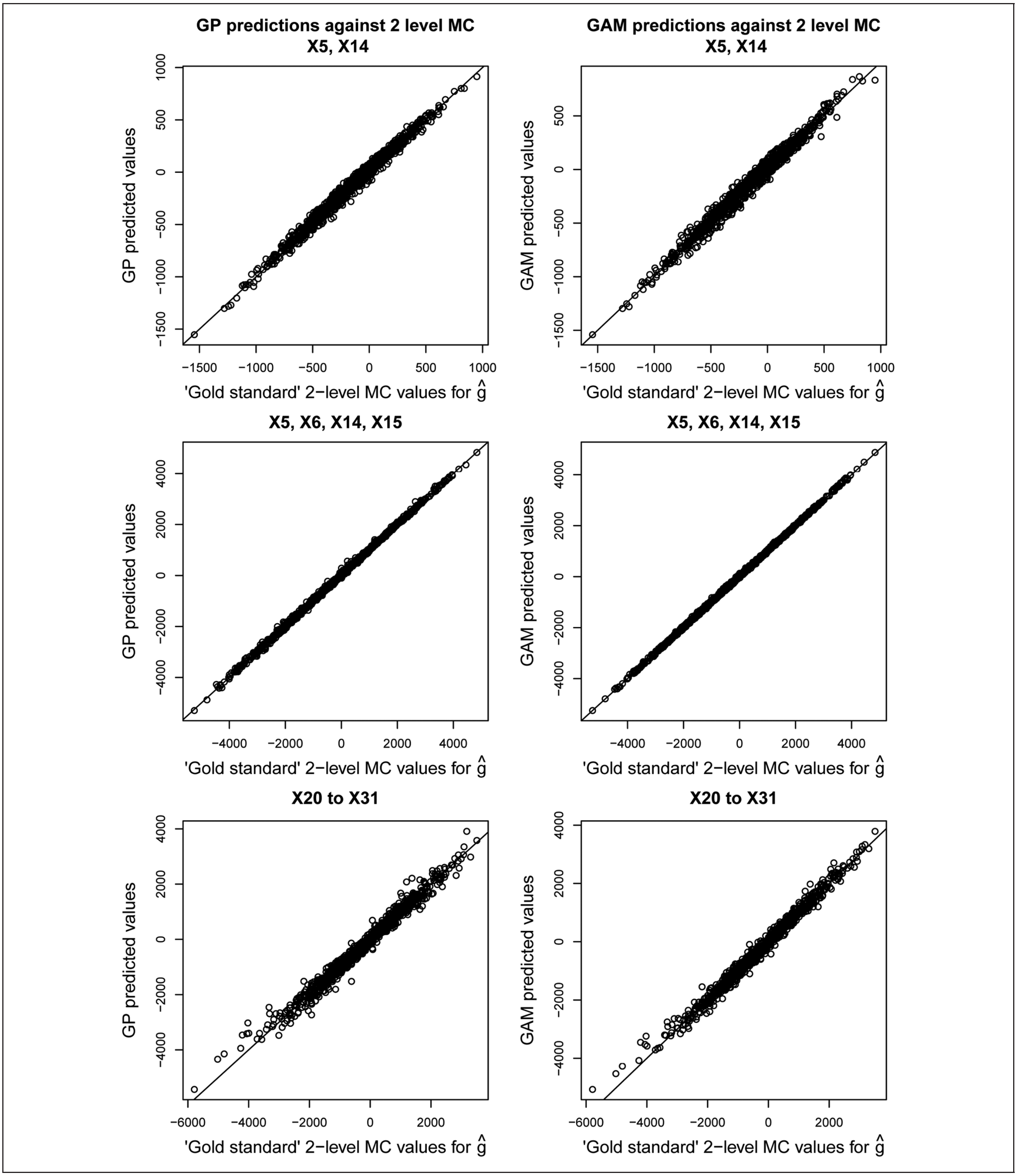

Figure 6. Gaussian process and Generalized Additive Model regression predictions versus those obtained via the gold standard 2-level Monte Carlo method for case study 2.

Table 3 Partial Expected Value of Perfect Information (EVPI) Values and Timings for Case Study 2

Sample Size			Partial EVPI (SE; upward bias), £			
Outer Loop	Inner Loop	Total	Parameter Set 1 $\{X_5, X_{14}\}$	Parameter Set 2 $\{X_5, X_6, X_{14}, X_{15}\}$	Parameter Set 3 $\{X_{20}$ to $X_{31}\}$	Mean Time
Two-level Monte Carlo[a]						
10^4	10^4	10^8	67.95 (1.43; 0.22)	587.14 (9.38; 0.03)	416.80 (6.50; 0.05)	17.7 h
Two-level Monte Carlo[b]						
10^1	10^3	10^4	5.77 (50.66; 2.01)	389.93 (296.6; 0.29)	178.93 (223.02; 0.39)	3.7 s
10^2	10^2	10^4	77.07 (21.96; 19.98)	661.75 (94.34; 2.40)	362.82 (71.78; 4.41)	2.9 s
10^3	10^1	10^4	228.32 (15.11; 148.24)	623.70 (31.43; 21.63)	467.61 (26.06; 42.32)	2.8 s
Two-level Monte Carlo[c]						
10^4	10^3	10^7	68.84 (4.47; 0.22)	595.14 (9.39; 0.13)	426.67 (6.51; 0.30)	1.8 h
GP regression[b]						
10^4	—	10^4	62.36 (10.35; 0.64)	582.32 (8.85; 0.13)	408.17 (10.30; 2.01)	198 s
GAM regression[b]						
10^4	—	10^4	62.53 (9.98; 0.47)	582.03 (8.23; 0.49)	409.80 (10.37; 1.03)	0.9 s

a. Reference gold standard.
b. Model runs restricted to 10^4.
c. *J* and *K* chosen to achieve SE and bias of the same order of magnitude as the regression estimates.

the estimated net benefit is stable. Nonconvergence will introduce additional noise in the estimation of the net benefit for each sample from the PSAlevel parameters.

Now, recall that in our approach, we treat all variability in the net benefit that is not due to the parameters of interest as noise (Equation 8). Any residual variability due to nonconvergence of the patient-level simulation will be treated as noise in the regression and averaged out. Because the regression estimation occurs before the maximization step, the residual first-order uncertainty will not cause an upward bias in the partial EVPI estimate.

Other Uses of the Gaussian Process in Health Economic Decision Modeling

In our method, we modeled the target conditional net benefit as an unknown smooth function of the parameters of interest. The observed net benefits in the PSA sample were treated as noisy data from which to learn about the unknown function. This use of a nonparametric regression method to approximate the (conditional) output of a health economic decision model is subtly different from the use of the Gaussian process in previous work by Stevenson and others,[16] Tappenden and others,[17] and Rojnik and Naversnik.[18] In these previous applications, the Gaussian process was used to model the net benefit itself as an unknown function of all the unknown input parameters, rather than to model the conditional net benefit as a function of the parameters of interest only. The primary purpose for using the Gaussian process was to construct a meta-model or emulator for the health economic decision model to allow a slow model to be replaced by a fast surrogate. Although this approach reduces computation time, the calculation of partial EVPI will typically still require a nested 2-level Monte Carlo approach. More importantly, this use of the Gaussian process does not address the problem of sampling from potentially difficult conditional distributions if input parameters are correlated.

Further Research

Although partial EVPI is useful in highlighting the sensitivity of the decision to any particular subset of input parameters, it represents only an upper bound on the expected value of undertaking research to reduce decision uncertainty. More useful is the expected value of sample information (EVSI), which represents the expected value of undertaking a particular data collection exercise.[11] We are currently working on extending the regression method described above to the computation of EVSI.

Conclusion

In conclusion, the regression-based approach to computing partial EVPI is likely to be of considerable benefit over the traditional 2-level Monte Carlo approach, except perhaps in models that are computationally very cheap to evaluate and in which there are no correlations in the inputs. With the increasing use of patient-level micro-simulation models, we envisage that obtaining partial EVPI via the traditional 2-level Monte Carlo approach will be considered just too time-consuming (in fact, experience suggests that the 2-level Monte Carlo procedure is considered too difficult for even moderately simple cohort models). In contrast, the regression methods we have presented provide a mechanism for rapidly estimating partial EVPI for any set of parameters in a model of any complexity.

REFERENCES

1. Raiffa H. Decision Analysis: Introductory Lectures on Choices Under Uncertainty. Reading, MA: Addison-Wesley; 1968.

2. Claxton K, Posnett J. An economic approach to clinical trial design and research priority-setting. Health Econ. 1996;5(6): 513–24.

3. Felli JC, Hazen GB. Sensitivity analysis and the expected value of perfect information. Med Decis Making. 1998;18(1):95–109.

4. Felli JC, Hazen GB. Erratum: Correction: sensitivity analysis and the expected value of perfect information. Med Decis Making. 2003;23(1):97.

5. Brennan A, Kharroubi S, O'Hagan A, Chilcott J. Calculating partial expected value of perfect information via Monte Carlo sampling algorithms. Med Decis Making. 2007;27(4):448–70.

6. Koerkamp BG, Myriam Hunink MG, Stijnen T, Weinstein MC. Identifying key parameters in cost-effectiveness analysis using value of information: a comparison of methods. Health Econ. 2006;15(4):383–92.

7. Strong M, Oakley JE. An efficient method for computing single-parameter partial expected value of perfect information. Med Decis Making. 2012;33(6):755–66.

8. Sadatsafavi M, Bansback N, Zafari Z, Najafzadeh M, Marra C. Need for speed: an efficient algorithm for calculation of single-parameter expected value of partial perfect information. Value Health. 2013;16(2):438–48.

9. Oakley JE, Brennan A, Tappenden P, Chilcott J. Simulation sample sizes for Monte Carlo partial EVPI calculations. J Health Econ. 2010;29(3):468–77.

10. Kharroubi SA, Brennan A, Strong M. Estimating expected value of sample information for incomplete data models using Bayesian approximation. Med Decis Making. 2011;31(6):839–52.

11. Ades AE, Lu G, Claxton K. Expected value of sample information calculations in medical decision modeling. Med Decis Making. 2004;24(2):207–27.

12. Hastie T, Tibshirani R. Generalized additive models. Stat Sci. 1986;1(3):297–318.

13. Wood SN. Generalized Additive Models: An Introduction with R. Boca Raton, FL: Chapman and Hall/CRC; 2006.

14. R Development Core Team. R: A Language and Environment for Statistical Computing. Vienna, Austria; 2012. Available from: URL: http://www.R-project.org/

15. Rasmussen CE, Williams CKI. Gaussian Processes for Machine Learning. Cambridge, MA: MIT Press; 2006.

16. Stevenson MD, Oakley J, Chilcott JB. Gaussian process modeling in conjunction with individual patient simulation modeling: a case study describing the calculation of cost-effectiveness ratios for the treatment of established osteoporosis. Med Decis Making. 2004;24(1):89–100.

17. Tappenden P, Chilcott JB, Eggington S, Oakley J, McCabe C. Methods for expected value of information analysis in complex health economic models: developments on the health economics of interferon-beta and glatiramer acetate for multiple sclerosis. Health Technol Assess. 2004;8(27):1–78.

18. Rojnik K, Naversnik K. Gaussian process metamodeling in Bayesian value of information analysis: a case of the complex health economic model for breast cancer screening. Value Health. 2008;11(2):240–50.

19. Andrianakis I, Challenor PG. The effect of the nugget on Gaussian process emulators of computer models. Comput Stat Data Anal. 2012;56(12):4215–28.

20. Gramacy R, Lee H. Cases for the nugget in modeling computer experiments. Stat Comput. 2012;22:713–22.