An efficient method for computing partial expected value of perfect information for correlated inputs

Mark Strong[1,§], Jeremy E. Oakley[2]

1. School of Health and Related Research (ScHARR), University of Sheffield, UK.

2. School of Mathematics and Statistics, University of Sheffield, UK.

[§] Corresponding author m.strong@sheffield.ac.uk

September 2011

Abstract

The value of learning an uncertain input in a decision model can be quantified by its partial expected value of perfect information (EVPI). This is commonly estimated via a two level nested Monte Carlo procedure in which the parameter of interest is sampled in an outer loop, and then conditional on this sampled value the remaining parameters are sampled in an inner loop. This two level method can be difficult to implement if the joint distribution of the inner loop parameters conditional on the parameter of interest is not easy to sample from. We present an simple alternative one level method for calculating partial EVPI that avoids the need to sample directly from the potentially problematic conditional distributions. We derive the sampling distribution of our estimator and show in a case study that it is both statistically and computationally more efficient than the two level method.

KEYWORDS: Expected value of perfect information; Economic evaluation model; Monte Carlo methods; Bayesian decision theory; Computational methods

1 Introduction

The value of learning an input to a decision analytic model can be quantified by its partial expected value of perfect information (partial EVPI) (Raiffa, 1968; Claxton and Posnett, 1996; Felli and Hazen, 1998, 2003). The standard two level Monte Carlo approach to calculating partial EVPI is to sample a value of the input parameter 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 (Brennan et al., 2007; Koerkamp et al., 2006). Sufficient numbers of runs of both the outer and inner loops are required to insure that the partial EVPI is estimated with sufficient precision, and with an acceptable level of bias (Oakley et al., 2010).

We recognise two important practical limitations to the standard two level Monte Carlo approach to calculating partial EVPI. Firstly, the nested two level nature of the algorithm with a model run at each inner loop step can be highly computationally demanding for all but very small loop sizes if the model is expensive to run. Secondly, we require a method of sampling from the joint distribution of the inputs (excluding the parameter of interest) conditional on the input parameter of interest. If the input parameter of interest is independent of the remaining parameters then we can simply sample from the unconditional joint distribution of the remaining parameters. Indeed, Ades et al. (2004) show that in certain classes of model, most notably decision tree models with independent inputs, the Monte Carlo inner loop is unnecessary since the target inner expectation has a closed form solution. However, if inputs are not independent we may need to resort to Markov chain Monte Carlo (MCMC) methods if there is no closed form analytic solution to the joint conditional distribution. Including an MCMC step in the algorithm is likely to increase the computational burden considerably, as well as requiring additional programming.

In this paper we present a simple one level 'ordered input' algorithm for calculating partial EVPI that takes into account any dependency in the inputs. The method avoids the need to sample directly from the conditional distributions of the inputs, and instead requires only a single set of the sampled inputs and corresponding outputs in order to calculate partial EVPI values for all input parameters. We derive an expression for the sampling variation of the estimator. We illustrate the method in a case study with two scenarios: inputs that are correlated, but with known closed form solutions for all conditional distributions, and inputs that are correlated where sampling from the conditional distributions requires MCMC.

2 Method

We assume we are faced with D decision options, indexed $d = 1, \ldots, D$, and have built a computer model $y_d = f(d, \boldsymbol{x})$ that aims to predict the net benefit of decision option d given a vector of input parameter values $\boldsymbol{x}$. We denote the true unknown values of the inputs $\boldsymbol{X} = \{X_1, \ldots, X_p\}$, and the uncertain net benefit under decision option d as Y_d. We denote the parameter for which we wish to calculate the partial EVPI as X_i and the remaining parameters as $\boldsymbol{X}_{-i} = \{X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_p\}$. We denote the expectation over the full joint distribution of $\boldsymbol{X}$ as $E_{\boldsymbol{X}}$, over the marginal distribution of X_i as E_{X_i}, and over the conditional distribution of $\boldsymbol{X}_{-i}|X_i$ as $E_{\boldsymbol{X}_{-i}|X_i}$. The partial EVPI for input X_i is

$$EVPI(X_i) = E_{X_i}\left[\max_d E_{\boldsymbol{X}_{-i}|X_i}\left\{f(d, X_i, \boldsymbol{X}_{-i})\right\}\right] - \max_d E_{\boldsymbol{X}}\{f(d, \boldsymbol{X})\}. \quad (1)$$

We wish to evaluate the partial EVPI for each input X_i without sampling directly from the conditional distribution $\boldsymbol{X}_{-i}|X_i$, since this may require computationally intensive numerical methods if inputs are correlated.

Our method for avoiding this difficulty rests on recognising the following. Given a Monte Carlo sample of S input parameter vectors drawn from the joint distribution $p(\boldsymbol{X})$, we can order the set of sample vectors with respect to the parameter of interest X_i, i.e.,

$$\begin{pmatrix} x_1^{(1)} & \ldots & x_i^{(1)} & \ldots & x_p^{(1)} \\ x_1^{(2)} & \ldots & x_i^{(2)} & \ldots & x_p^{(2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^{(S)} & \ldots & x_i^{(S)} & \ldots & x_p^{(S)} \end{pmatrix}, \quad (2)$$

where $x_i^{(1)} \leq x_i^{(2)} \leq \ldots \leq x_i^{(S)}$. Then, for some small integer δ and index k where $\delta < k \leq S - \delta$ the vectors $\boldsymbol{x}_{-i}^{(k-\delta)}, \ldots, \boldsymbol{x}_{-i}^{(k)}, \ldots, \boldsymbol{x}_{-i}^{(k+\delta)}$ are approximate samples from the conditional distribution $\boldsymbol{X}_{-i}|X_i = x_i^{(k)}$ if S is large compared to δ. We can approximate the problematic expectation in Eq. (1) by

$$E_{\boldsymbol{X}_{-i}|X_i}\left\{f(d, X_i, \boldsymbol{X}_{-i})\right\} \simeq \frac{1}{\delta + 1} \sum_{j=k-\delta}^{k+\delta} f\left(d, \boldsymbol{x}^{(j)}\right). \quad (3)$$

The second term in the RHS of Eq. (1) can be estimated simply via Monte Carlo sampling, i.e.

$$\max_d E_{\boldsymbol{X}}\{f(d,\boldsymbol{X})\} \simeq \max_d \frac{1}{N}\sum_{n=1}^{N} f(d,\boldsymbol{X}). \tag{4}$$

2.1 Algorithm for calculating partial EVPI via the one stage 'ordered input' method

We propose the following algorithm for computing the first term in the RHS of Eq. (1). Code for implementing the algorithm in R (R Development Core Team, 2011) is shown in appendix A and is available for download from `http://www.shef.ac.uk/scharr/sections/ph/staff/profiles/mark`.

2.1.1 Stage 1

We first obtain a single Monte Carlo sample $M = \{(\boldsymbol{x}^s, y^s), s = 1,\dots,S\}$ where $\boldsymbol{x}^s$ are drawn from the joint distribution of the inputs, $p(\boldsymbol{X})$, and $y_d^s = f(d,\boldsymbol{x}^s)$ is the evaluation of the model output at $\boldsymbol{x}^s$ for decision option d. Note the use of superscripts to index the randomly drawn sample sets. We let M be the matrix of inputs and corresponding outputs

$$M = \begin{pmatrix} x_1^1 & \dots & x_p^1 & y_1^1 & \dots & y_D^1 \\ x_1^2 & \dots & x_p^2 & y_1^2 & \dots & y_D^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^S & \dots & x_p^S & y_1^S & \dots & y_D^S \end{pmatrix}. \tag{5}$$

2.1.2 Stage 2

For parameter of interest i, we extract the x_i and $y_1,\dots,y_D$ columns and reorder with respect to x_i, giving

$$M^* = \begin{pmatrix} x_i^{(1)} & y_1^{(1)} & \dots & y_D^{(1)} \\ x_i^{(2)} & y_1^{(2)} & \dots & y_D^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ x_i^{(S)} & y_1^{(S)} & \dots & y_D^{(S)} \end{pmatrix}, \tag{6}$$

where $x_i^{(1)} \leq x_i^{(2)} \leq \dots \leq x_i^{(S)}$.

2.1.3 Stage 3

We partition the resulting matrix into $k = 1, \ldots, K$ sub matrices $M^{*(k)}$ of J rows each,

$$M^{*(k)} = \begin{pmatrix} x_i^{(1,k)} & y_1^{(1,k)} & \ldots & y_D^{(1,k)} \\ x_i^{(2,k)} & y_1^{(2,k)} & \ldots & y_D^{(2,k)} \\ \vdots & \vdots & \vdots & \vdots \\ x_i^{(J,k)} & y_1^{(J,k)} & \ldots & y_D^{(J,k)} \end{pmatrix}, \tag{7}$$

retaining the ordering with respect to x_i, and where the row indexed (j, k) in Eq. (7) is the row indexed $(j + (k-1)J)$ in Eq. (6). Note that $J \times K$ must equal the total sample size S.

2.1.4 Stage 4

For each $M^{*(k)}$ we estimate the conditional expectation $\mu_d^{(k)} = E_{\boldsymbol{X}_{-i}|X_i=x_i^{*(k)}} \{f(d, X_i, \boldsymbol{X}_{-i})\}$ for each decision option by

$$\hat{\mu}_d^{(k)} = \frac{1}{J} \sum_{j=1}^{J} y_d^{(j,k)}, \tag{8}$$

where $x_i^{*(k)} = \sum_{j=1}^{J} x_i^{(j,k)}/J$. The maximum $m^{(k)} = \max_d E_{\boldsymbol{X}_{-i}|X_i=x_i^{*(k)}} \{f(d, X_i, \boldsymbol{X}_{-i})\}$ is estimated by

$$\hat{m}^{(k)} = \max_d \hat{\mu}_d^{(k)}. \tag{9}$$

Finally, we estimate the first term in the RHS of Eq. (1) by

$$\bar{\hat{m}} = \frac{1}{K} \sum_{k=1}^{K} \hat{m}^{(k)}. \tag{10}$$

Stages 2 to 4 are repeated for each parameter of interest. Note that only a single set of model runs (stage 1) is required.

See appendix B for a theoretical justification of the algorithm.

2.2 Choosing values for J and K

We assume that we have a fixed number of model evaluations S and wish to choose values for J and K subject to the constraint $J \times K = S$.

Firstly we note that for small values of J the EVPI estimator is upwardly biased due to the maximisation in Eq. (9) (Oakley et al., 2010). Indeed for $J = 1$

(and $K = S$) our ordered input estimator for the first term in the RHS of Eq. (1) reduces to

$$\frac{1}{S}\sum_{s=1}^{S}\max_d(y_d^s), \tag{11}$$

which is the Monte Carlo estimator for the first term in the expression for the *overall* EVPI, $E_{\boldsymbol{X}}\{\max_d f(d, \boldsymbol{X})\} - \max_d E_{\boldsymbol{X}}\{f(d, \boldsymbol{X})\}$.

Secondly we note that for very large values of J, and hence small values of K, the EVPI estimator is downwardly biased, and converges to zero when $J = S$. In this case our ordered input estimator for the first term in the RHS of Eq. (1) reduces to

$$\max_d \frac{1}{S}\sum_{s=1}^{S} y_d^s, \tag{12}$$

which is the Monte Carlo estimator for the second term in the RHS of Eq. (1).

Given that the algorithm is computationally inexpensive we can find appropriate values for J and K empirically by running the algorithm at a range of values of J and K, subject to $J \times K = S$ (in practice we only need choose $J \times K \leq S$). Fig. (1) shows values for the estimated partial EVPI against J (on the $\log_{10}$ scale) for input X_6 in the case study outlined later in the paper. The total number of model evaluations, S, is 1,000,000. Note the upward and downward biases at extreme values of J, but also the large region of stability between $J = 100$ ($K = 10,000$) and $J = 100,000$ ($K = 10$).

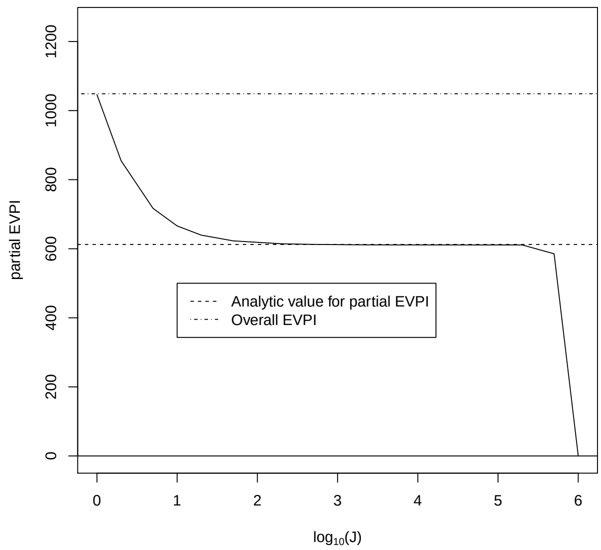

Figure 1: Partial EVPI at values of J ranging from 1 to 10^6 where the total number of model evaluations, S, is 10^6.

2.3 Estimating the precision of the partial EVPI estimator

For the purposes of this section we assume that we can estimate the second term in the RHS of Eq. (1) with sufficient accuracy by choosing large N in Eq. (4), and therefore that this second term does not contribute significantly to the variance of the estimate of the partial EVPI.

If we denote $d_k^* = \arg\max_d \left(\hat{\mu}_d^{(k)}\right)$ we can rewrite Eq. (10) as

$$\begin{aligned} \hat{E}_{X_i}(\hat{m}^{(k)}) = \bar{\hat{m}} &= \frac{1}{K}\sum_{k=1}^{K} \hat{m}^{(k)}, \\ &= \frac{1}{K}\sum_{k=1}^{K} \hat{\mu}_{d_k^*}^{(k)}, \\ &= \frac{1}{K}\sum_{k=1}^{K} \left(\frac{1}{J}\sum_{j=1}^{J} y_{d_k^*}^{(j,k)}\right), \\ &= \frac{1}{S}\sum_{k=1}^{K}\sum_{j=1}^{J} y_{d_k^*}^{(j,k)}. \end{aligned} \tag{13}$$

The variance of $\bar{\hat{m}}$ is

$$\begin{aligned}\mathrm{var}(\bar{\hat{m}}) &= \mathrm{var}\left(\frac{1}{S}\sum_{k=1}^{K}\sum_{j=1}^{J} y_{d_k^*}^{(j,k)}\right), \\ &= \frac{1}{S^2}\sum_{k=1}^{K}\sum_{j=1}^{J} \mathrm{var}\left(y_{d_k^*}^{(j,k)}\right), \end{aligned} \tag{14}$$

since the $y_{d_k^*}^{(j,k)}$ are independent. The estimator for $\mathrm{var}(\bar{\hat{m}})$ is therefore simply

$$\widehat{\mathrm{var}}(\bar{\hat{m}}) = \frac{1}{S(S-1)}\sum_{k=1}^{K}\sum_{j=1}^{J}\left(y_{d_k^*}^{(j,k)} - \bar{\hat{m}}\right)^2. \tag{15}$$

We see therefore that the precision of the estimator does not depend on the individual choices of J and K, but only on $S = J \times K$.

3 Case study

The case study is based on a hypothetical decision tree model previously used for illustrative purposes in Brennan et al. (2007), Oakley et al. (2010) and Kharroubi et al. (2011). The model predicts monetary net benefit, Y_d, under two decision options ($d = 1, 2$) and can be written in sum product form as

$$Y_1 = \lambda(X_5X_6X_7 + X_8X_9X_{10}) - (X_1 + X_2X_3X_4), \tag{16}$$

$$Y_2 = \lambda(X_{14}X_{15}X_{16} + X_{17}X_{18}X_{19}) - (X_{11} + X_{12}X_{13}X_4), \tag{17}$$

where $\boldsymbol{X} = \{X_1, \ldots, X_{19}\}$ are the 19 uncertain input parameters listed in Table 1, and the willingness to pay for one unit of health output in QALYs is $\lambda =$ £$10,000$/QALY. We implemented the model in R (R Development Core Team, 2011).

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%)
Utility Change if respond (X_6, X_{15})	0.30 (0.10)	0.30 (0.05)
Duration of response (years) (X_7, X_{16})	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 (years) (X_{10}, X_{19})	0.50 (0.20)	0.50 (0.20)

Table 1: Summary of input parameters

3.1 Scenario 1: correlated inputs with known conditional distributions

In scenario 1 we assume that a subset of the inputs are correlated, but with a joint distribution such that we can sample from the conditional distributions of the correlated inputs without the need for MCMC. We assume that the inputs are jointly normally distributed, with X_5, X_7, X_{14} and X_{16} all pairwise correlated with a correlation coefficient of 0.6, and with all other inputs independent. In a simple sum product form model the assumption of multivariate normality allows us to compute the inner conditional expectation analytically, as well as allowing us to sample directly from the conditional distribution $\boldsymbol{X}_{-i}|X_i$ in the standard nested two level method, but this will not necessarily be the case in models with additional non-linearity.

We calculated partial EVPI using three methods. Firstly, we calculated the partial EVPI for each parameter using a single loop Monte Carlo approximation for the outer expectation in the first term of the RHS of equation (1) with 10^6 samples from the distribution of the parameter of interest, and an analytic solution to the inner conditional expectation. Next, we calculated the partial EVPI values using the standard two level Monte Carlo approach with 1,000 inner loop samples and 1,000 outer loop samples (i.e 10^6 model evaluations in total). Finally, we computed the partial EVPI values using the ordered sample method with a single set of 10^6 samples and a value of $J = 1,000$.

Standard errors for the two level method estimates were obtained using the method presented in Oakley et al. (2010), and for the ordered input method estimates via Eq. (15). We measured the total computation time for obtaining EVPI values for all 19 parameters. We performed the computations on a single processor core on a 2.93GHz Intel Core i7 machine running 64 bit Linux.

3.1.1 Results for scenario 1

Calculating the expected net benefits for decision options 1 and 2 analytically results in values of £5057.00 and £5584.80 respectively, indicating that decision option 2 is optimal. Running the model with 10^6 Monte Carlo samples from the joint distribution of the input parameters results in option 2 having greater net benefit than option 1 in only 54% of samples, suggesting that the input uncertainty is resulting in considerable decision uncertainty. This is confirmed by the relatively large overall EVPI value of £1046.10.

The partial EVPI values for parameters X_1 to X_4, X_8 to X_{13} and X_{17} to X_{19} were all less than $£0.01$ and therefore considered unimportant in terms of driving the decision uncertainty. Results for the remaining parameters are shown in Table 2. The standard errors of the EVPI values estimated via the ordered input method are considerably smaller than those estimated via the two level method, and computation time is reduced by a factor of five.

Parameter	Partial EVPI (SE), £		
	Analytic conditional expectation	Two level Monte Carlo	Ordered input method
X_5	22.50	9.52 (65.20)	25.29 (3.26)
X_6	612.38	614.76 (33.16)	612.63 (3.15)
X_7	11.56	77.65 (66.38)	14.86 (3.28)
X_{14}	230.94	312.39 (69.59)	233.63 (3.19)
X_{15}	271.52	315.02 (29.52)	273.00 (3.30)
X_{16}	458.97	502.91 (77.98)	462.42 (3.12)
Computation time†		57 seconds	12 seconds

† Computation time is for all 19 input parameters

Table 2: Partial EVPI values for scenario 1

3.2 Scenario 2: correlated inputs with conditional distribution sampling requiring MCMC

In scenario 2 we assume that a subset of the inputs are correlated, but with a joint distribution such that we can only sample from the conditional distributions of the correlated inputs using MCMC. We assume, as in scenario 1, that X_5, X_7, X_{14} and X_{16} are pairwise correlated, but with a more complicated dependency structure based on an unobserved bivariate normal latent variable $\boldsymbol{Z} = (Z_1, Z_2)$ that has expectation zero, variance 1 and correlation 0.6. Conditional on this latent variable, which represents some measure of effectiveness, the proportions of responders (X_5 and X_{14}) are assumed beta distributed, and the durations of response (X_7 and X_{16}) assumed gamma distributed. The hyperparamters of the beta and gamma distributions are defined in terms of $\boldsymbol{Z}$ such that X_5, X_7, X_{14} and X_{16} have the means and standard deviations in table 1.

We calculated partial EVPI for each parameter using a the standard two level Monte Carlo approach with 1,000 inner loop samples and 1,000 outer loop samples (i.e 10^6 model evaluations in total) using OpenBUGS (Lunn et al., 2009) to sample from the conditional distribution of $\boldsymbol{X}_{-i}|X_i$.

3.2.1 Results for scenario 2

Running the model with 10^6 samples from the joint distribution of the input parameters resulted in expected net benefits of £5043.12 and £5549.93 for decision options 1 and 2 respectively, indicating that decision option 2 is optimal, but again with considerable decision uncertainty. Based on this sample, the probability that decision 2 is best is 54% and the overall EVPI £1240.33.

Partial EVPI results are shown in Table 3. Values for parameters X_1 to X_4, X_8 to X_{13} and X_{17} to X_{19} were again all less than £0.01 and are not shown. Standards errors for the partial EVPI values estimated via the order input method are again smaller than those estimated via the two level method. The total time required to compute partial EVPI for all 19 inputs was approximately 2.7 hours. In comparison, the ordered input method with a single set of 10^6 samples and a value of $J = 1,000$ took just 12 seconds, an approximately 1,000 fold reduction in computation time.

Parameter	Partial EVPI (SE), £	
	Two level Monte Carlo with MCMC inner loop	Ordered input method
X_5	102.55 (34.48)	34.65 (3.26)
X_6	610.82 (38.02)	618.80 (3.10)
X_7	132.16 (36.10)	56.25 (3.25)
X_{14}	334.13 (51.94)	368.87 (3.18)
X_{15}	223.09 (25.73)	275.78 (3.25)
X_{16}	554.20 (64.00)	663.25 (3.13)
Computation time†	2.7 hours	12 seconds

† Computation time is for all 19 input parameters

Table 3: Partial EVPI values for scenario 2

4 Conclusion

We have presented a method for calculating the partial expected value of perfect information that is simple to implement, rapid to compute, and does not require an assumption of independence between inputs. The saving in computational time is particularly marked if the alternative is to use a nested two level EVPI approach in which the conditional expectations are estimated using MCMC. The method is straightforward to apply, even with little programming knowledge in a spreadsheet application.

Our approach requires only a single set of model evaluations in order to calculate partial EVPI for all inputs, allowing a complete separation of the EVPI calculation step from the model evaluation step. This separation may be particularly useful when the model has been evaluated using specialist software (e.g. for discrete event or agent based 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.

As presented, the method calculates the partial EVPI for single inputs one at a time. We may however wish to calculate the value of learning groups of inputs simultaneously. Although it is possible to extend our approach to groups of inputs, we quickly come up against the 'curse of dimensionality'. This is because the method relies on partitioning the input space into a large number of 'small' sets such that in each set the parameter of interest lies close to some value. This works well where there is a single parameter of interest, but if we wish to calculate the EVPI for a group of parameters, the samples quickly become much more sparsely located in higher dimensional space. Given a single parameter of interest imagine that we obtain adequate precision if we partition the input space into $K = 1,000$ sets of $J = 1,000$ samples each. With two parameters of interest, we would need to order and partition the space in two dimensions, meaning that to retain the same marginal probabilistic 'size' for each set we now require $K^2 = 1,000,000$ sets of $J = 1,000$ samples each. For groups of inputs, the standard two level approach may be more efficient, or if this is impractical an alternative such as emulation (Oakley and O'Hagan, 2004; Oakley, 2009).

In conclusion, the ordered sample method for calculating partial EVPI is simple enough to be easily implemented in a range of software applications commonly used in cost-effectiveness modelling, reduces computation time considerably when compared with the standard two level Monte Carlo approach, and avoids the need for MCMC in non-linear models with awkward input parameter dependency structures.

Acknowledgements

MS is funded by UK Medical Research Council fellowship grant G0601721.

Appendix A - R code for implementing the algorithm

The `partial.evpi.function` function as written below takes as inputs the costs and effects rather than the net benefits. This allows the partial EVPI to be calculated at any value of willingness to pay, λ.

```
partial.evpi.function<-function(inputs,input.of.interest,costs,effects,lambda,J,K)
{
  S <- nrow(inputs) # number of samples
  if(J*K!=S) stop("The number of samples does not equal J times K")
  D <- ncol(costs) # number of decision options

  nb <- lambda*effects-costs
  baseline <- max(colMeans(nb))
  perfect.info <- mean(apply(nb,1,max))
  evpi <- perfect.info-baseline

  sort.order <- order(inputs[,input.of.interest])
  sort.nb <- nb[sort.order,]

  nb.array <- array(sort.nb,dim=c(J,K,D))
  mean.k <- apply(nb.array,c(2,3),mean)
  partial.info <- mean(apply(mean.k,1,max))
  partial.evpi <- partial.info-baseline
  partial.evpi.index <- partial.evpi/evpi

  return(list(
    baseline = baseline,
    perfect.info = perfect.info,
    evpi = evpi,
    partial.info = partial.info,
    partial.evpi = partial.evpi,
    partial.evpi.index = partial.evpi.index
  ))
}
```

Appendix B - theoretical justification

The ordered algorithm is a method for efficiently computing the inner expectation in the first term of the RHS in equation (1). Dropping the decision option index d for clarity but without loss of generality, our target is $E_{\boldsymbol{X}_{-i}|X_i=x_i^*}\{f(x_i^*, \boldsymbol{X}_{-i})\}$ where x_i^* is a realised value of the parameter of interest, and $\boldsymbol{X}_{-i}$ are the remaining (uncertain) parameters with joint conditional distribution $p(\boldsymbol{X}_{-i}|X_i = x_i^*)$.

Given a sample $\left\{\boldsymbol{x}_{-i}^{(1)}, \ldots, \boldsymbol{x}_{-i}^{(J)}\right\}$ from $p(\boldsymbol{X}_{-i}|X_i = x_i^*)$, the Monte Carlo estimator for $E_{\boldsymbol{X}_{-i}|X_i=x_i^*}\{f(x_i^*, \boldsymbol{X}_{-i})\}$ is

$$\hat{E}_{\boldsymbol{X}_{-i}|X_i=x_i^*}\{f(x_i^*, \boldsymbol{X}_{-i})\} = \frac{1}{J}\sum_{j=1}^{J} f\left(x_i^*, \boldsymbol{x}_{-i}^{(j)}\right). \tag{18}$$

In our ordered approximation method we replace (18) with

$$\hat{E}_{\boldsymbol{X}_{-i}|X_i=x_i^*}\{f(x_i^*, \boldsymbol{X}_{-i})\} = \frac{1}{J}\sum_{j=1}^{J} f\left(x_i^* + \varepsilon_j, \tilde{\boldsymbol{x}}_{-i}^{(j)}\right), \tag{19}$$

where $\{x_i^*+\varepsilon_1, \ldots, x_i^*+\varepsilon_J\} = \{x_i^{(1)}, \ldots, x_i^{(J)}\}$ is an ordered sample from $p(X_i|X_i \in [x_i^* \pm \zeta])$ for some small ζ (and therefore $\bar{\varepsilon} \simeq 0$), and $\tilde{\boldsymbol{x}}_{-i}^{(j)}$ is a sample from $p(\boldsymbol{X}_{-i}|X_i = x_i^* + \varepsilon_j)$

The expression (19) is an unbiased Monte Carlo estimator of

$$\begin{aligned} &E_{X_i\in[x_i^*\pm\zeta]}\left\{E_{\boldsymbol{X}_{-i}|X_i} f(X_i, \boldsymbol{X}_{-i})\right\} \\ &= \int_{\mathcal{X}_{-i}}\int_{\mathcal{X}_i} f(X_i, \boldsymbol{X}_{-i}) p(\boldsymbol{X}_{-i}|X_i) p(X_i|X_i \in [x_i^* \pm \zeta]) dX_i d\boldsymbol{X}_{-i}, \end{aligned} \tag{20}$$

which we can rewrite by introducing an importance sampling ratio as

$$\begin{aligned} &\int_{\mathcal{X}_{-i}}\int_{\mathcal{X}_i} f(X_i, \boldsymbol{X}_{-i}) p(\boldsymbol{X}_{-i}|X_i) p(X_i|X_i \in [x_i^* \pm \zeta]) dX_i d\boldsymbol{X}_{-i} \\ &= \int_{\mathcal{X}_{-i}}\int_{\mathcal{X}_i} f(X_i, \boldsymbol{X}_{-i}) \frac{p(\boldsymbol{X}_{-i}|X_i) p(X_i|X_i \in [x_i^* \pm \zeta])}{p(\boldsymbol{X}_{-i}|X_i) p(X_i|X_i = x_i^*)} p(\boldsymbol{X}_{-i}|X_i) p(X_i|X_i = x_i^*) dX_i d\boldsymbol{X}_{-i} \\ &= \int_{\mathcal{X}_{-i}}\int_{\mathcal{X}_i} f(X_i, \boldsymbol{X}_{-i}) \frac{p(\boldsymbol{X}_{-i}|X_i)}{p(\boldsymbol{X}_{-i}|X_i = x_i^*)} p(X_i|X_i \in [x_i^* \pm \zeta]) dX_i \; p(\boldsymbol{X}_{-i}|X_i = x_i^*) \; d\boldsymbol{X}_{-i}. \end{aligned} \tag{21}$$

We write the terms $f(X_i, \boldsymbol{X}_{-i})\frac{p(\boldsymbol{X}_{-i}|X_i)}{p(\boldsymbol{X}_{-i}|X_i=x_i^*)}$ within the inner integral as a function $g(\cdot)$, i.e.

$$f(X_i, \boldsymbol{X}_{-i})\frac{p(\boldsymbol{X}_{-i}|X_i)}{p(\boldsymbol{X}_{-i}|X_i = x_i^*)} = g(X_i, x_i^*, \boldsymbol{X}_{-i}).$$

If $g(\cdot)$ is approximately linear in the small interval $X_i \in [x_i^* \pm \zeta]$ then we can express $g(X_i, x_i^*, \boldsymbol{X}_{-i})$ as a first order Taylor series expansion about $g(x_i^*, x_i^*, \boldsymbol{X}_{-i})$, giving

$$\begin{aligned}
f(X_i, \boldsymbol{X}_{-i}) \frac{p(\boldsymbol{X}_{-i}|X_i)}{p(\boldsymbol{X}_{-i}|X_i = x_i^*)} &= g(X_i, x_i^*, \boldsymbol{X}_{-i}), \\
&\simeq g\left(x_i^*, x_i^*, \boldsymbol{X}_{-i}\right) + (X_i - x_i^*) \frac{\partial g\left(X_i, x_i^*, \boldsymbol{X}_{-i}\right)}{\partial X_i}\Big|_{X_i = x_i^*} \\
&= f(x_i^*, \boldsymbol{X}_{-i}) + (X_i - x_i^*) \frac{\partial g\left(X_i, x_i^*, \boldsymbol{X}_{-i}\right)}{\partial X_i}\Big|_{X_i = x_i^*}.
\end{aligned}$$

Substituting back into (21) with $c = \frac{\partial g\left(X_i, x_i^*, \boldsymbol{X}_{-i}\right)}{\partial X_i}\Big|_{X_i = x_i^*}$ gives

$$\begin{aligned}
&\int_{\mathcal{X}_{-i}} \int_{\mathcal{X}_i} f(X_i, \boldsymbol{X}_{-i}) \frac{p(\boldsymbol{X}_{-i}|X_i)}{p(\boldsymbol{X}_{-i}|X_i = x_i^*)} p(X_i|X_i \in [x_i^* \pm \zeta]) dX_i \; p(\boldsymbol{X}_{-i}|X_i = x_i^*) \; d\boldsymbol{X}_{-i} \\
&\simeq \int_{\mathcal{X}_{-i}} \int_{\mathcal{X}_i} \left\{ f(x_i^*, \boldsymbol{X}_{-i}) + c(X_i - x_i^*) \right\} p(X_i|X_i \in [x_i^* \pm \zeta]) \; dX_i \; p(\boldsymbol{X}_{-i}|X_i = x_i^*) \; d\boldsymbol{X}_{-i}.
\end{aligned}$$

Since $\int_{\mathcal{X}_i} c(X_i - x_i^*) p(X_i|X_i \in [x_i^* \pm \zeta]) dX_i = E_{X_i \in [x_i^* \pm \zeta]}\{c(X_i - x_i^*)\} \simeq 0$ and $\int_{\mathcal{X}_i} p(X_i|X_i \in [x_i^* \pm \zeta]) \; dX_i = 1$, then

$$\begin{aligned}
&\int_{\mathcal{X}_{-i}} \int_{\mathcal{X}_i} \left\{ f(x_i^*, \boldsymbol{X}_{-i}) + c(X_i - x_i^*) \right\} p(X_i|X_i \in [x_i^* \pm \zeta]) \; dX_i \; p(\boldsymbol{X}_{-i}|X_i = x_i^*) \; d\boldsymbol{X}_{-i}, \\
&= \int_{\mathcal{X}_{-i}} f(x_i^*, \boldsymbol{X}_{-i}) p(\boldsymbol{X}_{-i}|X_i = x_i^*) \; d\boldsymbol{X}_{-i}, \\
&= E_{\boldsymbol{X}_{-i}|X_i = x_i^*}\{f(x_i^*, \boldsymbol{X}_{-i})\}.
\end{aligned}$$

Hence, we have shown that as long as $g(X_i, x_i^*, \boldsymbol{X}_{-i}) = f(X_i, \boldsymbol{X}_{-i}) \frac{p(\boldsymbol{X}_{-i}|X_i)}{p(\boldsymbol{X}_{-i}|X_i = x_i^*)}$ is sufficiently smooth such that it is approximately linear in some small interval $X_i \in [x_i^* \pm \zeta]$, the ordered approximation method (19) will provide a good estimate of our target conditional expectation $E_{\boldsymbol{X}_{-i}|X_i = x_i^*}\{f(x_i^*, \boldsymbol{X}_{-i})\}$.

References

Ades, A. E., Lu, G. and Claxton, K. (2004). Expected value of sample information calculations in medical decision modeling, *Medical Decision Making*, **24 (2)**: 207–227.

Brennan, A., Kharroubi, S., O'Hagan, A. and Chilcott, J. (2007). Calculating partial expected value of perfect information via Monte Carlo sampling algorithms, *Medical Decision Making*, **27 (4)**: 448–470.

Claxton, K. and Posnett, J. (1996). An economic approach to clinical trial design and research priority-setting, *Health Economics*, **5 (6)**: 513–524.

Felli, J. C. and Hazen, G. B. (1998). Sensitivity analysis and the expected value of perfect information, *Medical Decision Making*, **18 (1)**: 95–109.

Felli, J. C. and Hazen, G. B. (2003). Erratum: Correction: Sensitivity analysis and the expected value of perfect information, *Medical Decision Making*, **23 (1)**: 97.

Kharroubi, S. A., Brennan, A. and Strong, M. (2011). Estimating expected value of sample information for incomplete data models using Bayesian approximation, *Medical Decision Making*, . Epub ahead of print http://mdm.sagepub.com/content/early/2011/04/21/0272989X11399920.abstract.

Koerkamp, B. G., Myriam Hunink, M. G., Stijnen, T. and Weinstein, M. C. (2006). Identifying key parameters in cost-effectiveness analysis using value of information: a comparison of methods, *Health Economics*, **15 (4)**: 383–392.

Lunn, D., Spiegelhalter, D., Thomas, A. and Best, N. (2009). The BUGS project: Evolution, critique, and future directions, *Statistics in Medicine*, **28**: 3049–3067.

Oakley, J. E. (2009). Decision-theoretic sensitivity analysis for complex computer models, *Technometrics*, **51 (2)**: 121–129.

Oakley, J. E., Brennan, A., Tappenden, P. and Chilcott, J. (2010). Simulation sample sizes for Monte Carlo partial EVPI calculations, *Journal of Health Economics*, **29 (3)**: 468–477.

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

R Development Core Team (2011). *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0.

Raiffa, H. (1968). *Decision Analysis. Introductory Lectures on Choices Under Uncertainty*, Reading, Massachusetts: Addison-Wesley.