The approximate coordinate exchange algorithm for Bayesian optimal design of experiments

Antony Overstall
School of Mathematics & Statistics,
University of Glasgow,
Glasgow
UK

David Woods
Statistical Sciences Research Institute
University of Southampton,
Southampton
UK

Abstract
Optimal Bayesian experimental design typically involves maximising the expectation, with respect to the joint distribution of parameters and responses, of some appropriately chosen utility function. This objective function is usually not available in closed form and the design space can be of high dimensionality. The approximate coordinate exchange algorithm is proposed for this maximisation problem where a Gaussian process emulator is used to approximate the objective function. The algorithm can be used for arbitrary utility functions meaning we can consider fully Bayesian optimal design. It can also be used for those utility functions that result in pseudo-Bayesian designs such as the popular Bayesian D-optimality. The algorithm is demonstrated on a range of examples.
Keywords
Bayesian; coordinate exchange; Gaussian process emulator; optimal experimental design

1 Introduction

1.1 Bayesian optimal design

Optimal experimental design refers to the "best" allocation of the resources for an experiment where there is some degree of uncertainty in the responses. By "best" we mean that the design maximises a specified utility of the experiment. This utility typically depends on the assumed data-generating process, i.e. a statistical model. For Bayesian optimal experimental design, the data-generating process includes a prior distribution for the unknown model parameters. This distribution quantifies our knowledge of the model parameters prior to conducting the experiment and allows us to explicitly use this knowledge to help find an optimal design.

Suppose the experiment consists of n runs where each run consists of a treatment of k factors and the observation of a response. Let $\mathbf{D}$ denote the $n \times k$ design matrix where the ith row, $\mathbf{d}_i$, for $i = 1, \ldots, n$, specifies the settings of the k factors. Furthermore, let $\boldsymbol{\delta} = \text{vec}(\mathbf{D}) \in \mathcal{D}$ denote the $nk \times 1$ vector found by stacking the columns of $\mathbf{D}$ into a

vector and where we let $\mathcal{D}$ denote the q-dimensional design space with $q = nk$. An optimal design is found by maximising the objective function, $U(\boldsymbol{\delta})$, given by the expectation of a utility function with respect to the joint distribution of model parameters and unobserved responses. Mathematically,

$$U(\boldsymbol{\delta}) = \mathrm{E}_{\boldsymbol{\psi},\mathbf{y}}\left[u(\boldsymbol{\psi}, \mathbf{y}, \boldsymbol{\delta})\right] = \int u(\boldsymbol{\psi}, \mathbf{y}, \boldsymbol{\delta}) \, \mathrm{dP}_{\boldsymbol{\psi},\mathbf{y}},$$

where $u(\boldsymbol{\psi}, \mathbf{y}, \boldsymbol{\delta})$ denotes the utility function depending on the unknown model parameters, $\boldsymbol{\psi} \in \Psi$; the unobserved responses, $\mathbf{y} \in \mathcal{Y}$; and the design. In this case, Ψ is the P-dimensional parameter space, and $\mathcal{Y}$ is the n-dimensional sample space. Each element of the $n \times 1$ vector $\mathbf{y}$ represents the response from each run of the experiment. We discuss some common choices of utility function in Section 1.3.

The joint distribution of model parameters and unobserved responses, $\pi(\boldsymbol{\psi}, \mathbf{y}|\boldsymbol{\delta})$, can be factorised as

$$\pi(\boldsymbol{\psi}, \mathbf{y}|\boldsymbol{\delta}) = \pi(\mathbf{y}|\boldsymbol{\psi}, \boldsymbol{\delta})\pi(\boldsymbol{\psi}), \tag{1}$$

where $\pi(\boldsymbol{\psi})$ is the probability density of the prior distribution of $\boldsymbol{\psi}$ and $\pi(\mathbf{y}|\boldsymbol{\psi}, \boldsymbol{\delta})$ represents the joint distribution of the responses conditional on the parameters. Once the responses have been observed, and when considered as a function of $\boldsymbol{\psi}$, $\pi(\mathbf{y}|\boldsymbol{\psi}, \boldsymbol{\delta})$ is called the likelihood function. Note from (1) that the prior distribution for $\boldsymbol{\psi}$ is assumed to not depend on the design, $\boldsymbol{\delta}$, although the methodology proposed in this paper does not rely on this assumption.

1.2 Challenges and existing approaches

In practice, obtaining the optimal design by maximising the expected utility function is associated with two problems.

1. The design space, $\mathcal{D}$, can be of high dimensionality making the optimisation challenging.
2. In all but the most trivial cases, the integration required in the evaluation of $U(\boldsymbol{\delta})$ will be analytically intractable.

Simulation-based methods provide a obvious approach to these problems. Muller (1999) considered the function

$$h(\boldsymbol{\psi}, \mathbf{y}, \boldsymbol{\delta}) \propto u(\boldsymbol{\psi}, \mathbf{y}, \boldsymbol{\delta})\pi(\boldsymbol{\psi}, \mathbf{y}|\boldsymbol{\delta}), \tag{2}$$

as the density of a joint distribution where the marginal mode of $\boldsymbol{\delta}$ is equivalent to the optimal design. Muller (1999) proposed using simulation methods (e.g. Markov chain Monte Carlo) to generate a sample from increasingly tempered versions of the joint distribution given by (2). The optimal design can be approximated by the sample mean of $\boldsymbol{\delta}$. The problem of the high dimensionality of $\mathcal{D}$ still plagues the method of Muller (1999) in terms of specifying an efficient simulation method. Ryan et al. (2014) use this method for designs of one factor ($k = 1$) but where n may be large. To counter the high dimensionality, Ryan et al. (2014) employ a dimension reduction scheme to, essentially, change the problem to determining the first design point and a spacing parameter to find the subsequent design points. The Muller (1999) simulation method is applied to maximising the expected utility over the two-dimensional modified design space where Monte Carlo integration is used to approximate $U(\boldsymbol{\delta})$. See Muller et al. (2004) and Amzal et al. (2006) for further examples of this simulation-based approach.

When $U(\boldsymbol{\delta})$ is analytically tractable, an approach to the problem of a high dimensional design space is the coordinate exchange algorithm (Meyer and Nachtsheim, 1995) which is an example of a cyclic ascent (or Gauss-Seidel) optimisation method (see, for example, Bazaraa et al., 2006, pages 365-368). Here the objective function $U(\boldsymbol{\delta})$ is maximised over each element (or coordinate) of $\mathcal{D}$ sequentially. Thus, for each element $i = 1, \ldots, q$, we cycle through the following steps:

1. Let $\boldsymbol{\delta}^C = \left(\delta_1^C, \ldots, \delta_q^C\right)$ be the current design.
2. Define $U_i(\delta) = U\left(\delta_1^C, \ldots, \delta_{i-1}^C, \delta, \delta_{i+1}^C, \ldots, \delta_q^C\right)$.
3. Let $\delta_i^C = \arg\max_{\delta \in \mathcal{D}_i} U_i(\delta)$, where $\mathcal{D}_i$ corresponds to the ith dimension of $\mathcal{D}$ (which may depend on $\boldsymbol{\delta}_{\setminus i}^C$).

Another approach is the point exchange algorithm. This relies on the specification of a candidate set of design points. At each iteration, the candidate point which results in the greatest improvement in $U(\boldsymbol{\delta})$ is added to the current design, and then the design point, in the augmented current design, which, when removed, results in the smallest reduction in $U(\boldsymbol{\delta})$, is removed. See, for example, Morris (2011, pages 310-311) for a more detailed description of the point exchange algorithm.

Chaloner and Verdinelli (1995) describe how a normal approximation to the posterior distribution of $\boldsymbol{\psi}$ can lead to approximations to $U(\boldsymbol{\delta})$ in which the expectation is only with respect to the marginal distribution of $\boldsymbol{\psi}$ (i.e. the prior distribution). Whilst, typically, still analytically intractable, an appealing feature of these approximations is their relationship to classical optimal experimental design (see Section 1.3). Gotwalt et al. (2009) applied a deterministic quadrature rule to approximate the objective function which is the prior expectation of a particular loss function. They applied their method, in conjunction with the coordinate exchange algorithm, to compartmental and logistic regression models.

In this paper, we extend the idea of Gotwalt et al. (2009) by considering the approximate coordinate exchange algorithm for arbitary data-generating processes and/or utility functions. This is simply application of the coordinate exchange algorithm but where evaluation of the intractable $U_i(\delta)$ is replaced by evaluation of an approximation.

Monte Carlo integration provides a straightforward mechanism for approximating the expected utility and has been applied by, for example, Muller and Parmigiani (1996), Hamada et al. (2001) and Hainy et al. (2013) to find Bayesian optimal designs. However, the applicability of these approaches when $\mathcal{D}$ is of high dimensionality has yet to be demonstrated. Muller and Parmigiani (1996) formed an approximation to $U(\boldsymbol{\delta})$ by fitting a statistical model to Monte Carlo integration evaluations of $U(\boldsymbol{\delta})$. That is, they essentially smoothed the evaluations to give a predictive equation for $U(\boldsymbol{\delta})$ which is then maximised (not using the coordinate exchange algorithm). We follow this approach for approximating $U_i(\delta)$ and, in particular, use a Gaussian process emulator as the smoothing statistical model. At each iteration of the coordinate exchange algorithm, we refit the emulator so that it adapts to the shape of the expected utility as we get closer to the maximum. Gaussian process emulators are a very flexible class of model and their use for approaching optimisation problems can be traced back to, for example, Jones et al. (1998).

1.3 Utility functions

In this section we discuss the utility function, the choice of which should be driven by the ultimate goal of the experiment. For example, the choice of utility function should reflect the fact that we may be interested in prediction or estimating some function of parameters. See Chaloner and Verdinelli (1995) for examples of utility functions. However, for the purpose of demonstrating the methodology in this paper, in the examples of Section 4, we focus on two commonly used utility functions for parameter estimation; Shannon information gain (Lindley 1956) and negative squared error loss.

Suppose that the model parameters can be decomposed as $\boldsymbol{\phi} = (\boldsymbol{\theta}, \boldsymbol{\gamma})$, where $\boldsymbol{\theta}$ denotes the $p \times 1$ vector of parameters of interest and $\boldsymbol{\gamma}$ denotes the $(P-p) \times 1$ vector of nuisance parameters.

1.3.1 Shannon information gain (SIG)

The SIG utility is given by

$$u^S(\boldsymbol{\theta}, \mathbf{y}, \boldsymbol{\delta}) = \log \pi(\boldsymbol{\theta}|\mathbf{y}, \boldsymbol{\delta}) - \log \pi(\boldsymbol{\theta}).$$

The expectation of SIG with respect to the posterior distribution of $\boldsymbol{\theta}$ is the Kullback-Liebler distance (KLD) between the marginal posterior and prior distributions of $\boldsymbol{\theta}$. Therefore maximising $U^S(\boldsymbol{\delta}) = \mathrm{E}_{\boldsymbol{\psi},\mathbf{y}}\left(u^S(\boldsymbol{\theta}, \mathbf{y}, \boldsymbol{\delta})\right)$ is equivalent to maximising the expectation (with respect to the marginal distribution of $\mathbf{y}$) of this KLD. Also note that this is equivalent to minimising the expected entropy of the posterior distribution.

Consider the linear model with conjugate normal-inverse-gamma prior distribution

$$\begin{aligned} \mathbf{y}|\boldsymbol{\beta}, \sigma^2 &\sim \mathrm{N}\left(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I}_n\right), \\ \boldsymbol{\beta}|\sigma^2 &\sim \mathrm{N}\left(\boldsymbol{\mu}_0, \sigma^2\mathbf{V}_0\right), \\ \sigma^2 &\sim \mathrm{IG}\left(\frac{a}{2}, \frac{b}{2}\right), \end{aligned}$$

where $\mathbf{I}_n$ denotes the $n \times n$ identity matrix. The design, $\boldsymbol{\delta}$, specifies the elements of the $n \times p$ model matrix, $\mathbf{X}$. Let $\boldsymbol{\beta}$ be the $p \times 1$ vector of parameters of interest and σ^2 be a nusiance parameter. The expected SIG is

$$U^S(\boldsymbol{\delta}) = \frac{1}{2}\log|\mathbf{V}_0| + \frac{1}{2}\log|\mathbf{V}_0^{-1} + \mathbf{X}^T\mathbf{X}|.$$

Chaloner and Verdinelli (1995) point out that, in some cases, $U(\boldsymbol{\delta})$ is a strictly monotonic function of a simpler function, $\phi(\boldsymbol{\delta})$. Therefore maximising $U(\boldsymbol{\delta})$ is equivalent to maximising the function $\phi(\boldsymbol{\delta})$. Since $U^S(\boldsymbol{\delta})$ is a strictly monotonic function of

$$\phi^S(\boldsymbol{\delta}) = \log|\mathbf{V}_0^{-1} + \mathbf{X}^T\mathbf{X}|, \tag{3}$$

maximising $\phi^S(\boldsymbol{\delta})$ is equivalent to maximising $U^S(\boldsymbol{\delta})$. It can also be shown that if interest lies in both $\boldsymbol{\beta}$ and σ^2, then the expected utility is also a strictly monotonic function of (3).

Additionally, under a non-informative prior distribution for $\boldsymbol{\beta}$ (where $\mathbf{V}_0^{-1} = \mathbf{0}$) and with $n \geq p$, $\phi^S(\boldsymbol{\delta}) = \log|\mathbf{X}^T\mathbf{X}|$, meaning the optimal design under expected SIG is equivalent to the classical D-optimal design (Atkinson et al., 2007, Chapter 10).

1.3.2 Negative squared error loss (NSEL)

The NSEL utility is given by

$$\begin{aligned} u^V(\boldsymbol{\theta}, \mathbf{y}, \boldsymbol{\delta}) &= -\sum_{i=1}^{p} \left[\theta_i - \mathrm{E}(\theta_i|\mathbf{y}, \boldsymbol{\delta})\right]^2, \\ &= -\left[\boldsymbol{\theta} - \mathrm{E}(\boldsymbol{\theta}|\mathbf{y}, \boldsymbol{\delta})\right]^T \left[\boldsymbol{\theta} - \mathrm{E}(\boldsymbol{\theta}|\mathbf{y}, \boldsymbol{\delta})\right]. \end{aligned} \tag{4}$$

The expectation of NSEL with respect to the marginal posterior distribution of $\boldsymbol{\theta}$ is the negative trace of the posterior variance matrix. Therefore maximising $U^V(\boldsymbol{\delta})$ is equivalent to minimising the expected (with respect to the marginal distribution of $\mathbf{y}$) average variance of the parameters of interest. For the linear model with parameters of interest $\boldsymbol{\beta}$, it can be shown that

$$U^V(\boldsymbol{\delta}) = -\frac{b}{a-2}\mathrm{tr}\left\{\left(\mathbf{V}_0^{-1} + \mathbf{X}^T\mathbf{X}\right)^{-1}\right\},$$

so that

$$\phi^V(\boldsymbol{\delta}) = -\mathrm{tr}\left\{\left(\mathbf{V}_0^{-1} + \mathbf{X}^T\mathbf{X}\right)^{-1}\right\}.$$

Under the non-informative prior,

$$\phi^V(\boldsymbol{\delta}) = -\mathrm{tr}\left\{\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\right\},$$

which means, under a linear model with a non-informative prior, the optimal design under NSEL is equivalent to the classical A-optimal design (Atkinson et al., 2007, Chapter 10).

For non-linear models, the expected utility under SIG and NSEL will, typically, be intractable. Chaloner and Verdinelli (1995) and Muller and Parmigiani (1996) use a normal approximation to the posterior distribution of $\boldsymbol{\psi}$ to justify the following approximations

$$\begin{aligned} \hat{\phi}^S(\boldsymbol{\delta}) &= \mathrm{E}_{\boldsymbol{\psi}}\left(\log|\mathcal{I}(\boldsymbol{\psi};\boldsymbol{\delta})|\right) = \int_\Theta \log|\mathcal{I}(\boldsymbol{\psi};\boldsymbol{\delta})|\pi(\boldsymbol{\psi})\mathrm{d}\boldsymbol{\psi}, \\ \hat{\phi}^V(\boldsymbol{\delta}) &= -\mathrm{E}_{\boldsymbol{\psi}}\left(\mathrm{tr}\left\{\mathcal{I}(\boldsymbol{\psi};\boldsymbol{\delta})\right\}^{-1}\right) = -\int_\Theta \mathrm{tr}\left\{\mathcal{I}(\boldsymbol{\psi};\boldsymbol{\delta})^{-1}\right\}\pi(\boldsymbol{\psi})\mathrm{d}\boldsymbol{\psi}, \end{aligned}$$

where $\mathcal{I}(\boldsymbol{\psi};\boldsymbol{\delta})$ denotes the Fisher information. Designs that maximise $\hat{\phi}^S$ and $\hat{\phi}^V$ are used under the classical approach to statistics and are called Bayesian D- and A-optimal designs, respectively, or, collectively, pseudo-Bayesian designs.

Although $\hat{\phi}^S$ and $\hat{\phi}^V$ are certainly simpler expressions than the original expected utilities, they are still, typically, intractable. Lastly, note that $\hat{\phi}^S$ and $\hat{\phi}^V$ still fit into the general framework, from Section 1.1, by specifying the following utility functions

$$\begin{aligned} u^D(\boldsymbol{\psi}, \mathbf{y}, \boldsymbol{\delta}) &= \log|\mathcal{I}(\boldsymbol{\psi};\boldsymbol{\delta})|, \\ u^A(\boldsymbol{\psi}, \mathbf{y}, \boldsymbol{\delta}) &= -\mathrm{tr}\left\{\mathcal{I}(\boldsymbol{\psi};\boldsymbol{\delta})\right\}^{-1}, \end{aligned}$$

respectively, which do not depend on $\mathbf{y}$. We denote the resulting expected utilities as U^D and U^A, respectively.

1.4 Layout of paper

The layout of the remainder of the paper is as follows. In Section 2 we outline the approximate coordinate exchange (ACE) algorithm. In Section 3 we describe how Monte Carlo integration and Gaussian process emulators can be used to approximate the expected utility in the ACE algorithm. The ACE algorithm is then applied to a range of challenging examples in Section 4.

2 Approximate Coordinate Exchange Algorithm

In this section we provide the basic framework of the approximate coordinate exchange algorithm (ACE) algorithm, which is modified from the algorithm used by Gotwalt et al. (2009).

It has two phases. Phase I uses the coordinate exchange algorithm where evaluation of $U(\delta)$ is replaced by evaluation of an approximation, denoted by $\tilde{U}(\delta)$. Gotwalt et al. (2009) noted that Phase I tends to produce clusters of design points. Phase II allows these clusters to be consolidated into a single design point, which is then repeated. This is achieved using the point exchange algorithm where the candidate set is the design found by Phase I and where evaluation of $U(\delta)$ is, again, replaced by evaluation of $\tilde{U}(\delta)$.

Strictly speaking, Phases I and II form an approximate coordinate and point exchange algorithm but we retain the name "approximate coordinate exchange" for brevity.

Specifically, the algorithm is as follows.

1. Choose an initial design $\boldsymbol{\delta}^0$ and set the current design to be $\boldsymbol{\delta}^C = \boldsymbol{\delta}^0$.

Phase I (coordinate exchange)

2. For $i = 1, \ldots, q$,
   (a) Let $\tilde{U}_i(\delta) = \tilde{U}(\delta_1^C, \ldots, \delta_{i-1}^C, \delta, \delta_{i+1}^C, \ldots, \delta_q^C)$ be the approximation to $U_i(\delta)$.
   (b) Let $\delta_i^* = \arg\ \max_{\delta \in \mathcal{D}_i} \tilde{U}_i(\delta)$.
   (c) Set $\boldsymbol{\delta}^C = (\delta_1^C, \ldots, \delta_{i-1}^C, \delta_i^*, \delta_{i+1}^C, \ldots, \delta_q^C)$.
3. Repeat step 2 N_I times until convergence.

Phase II (point exchange)

4. Let $\mathbf{D}$ be the design found by Phase I and let $\mathbf{D}^C = \mathbf{D}$ be the current design.
5. (a) For $i = 1, \ldots, n$, calculate

$$r_i = \tilde{U}(\boldsymbol{\delta}_i^{(1)}),$$

   where $\boldsymbol{\delta}_i^{(1)} = \text{vec}(\mathbf{D}_i^{(1)})$ and

$$\mathbf{D}_i^{(1)} = \begin{pmatrix} \mathbf{D}^C \\ \mathbf{d}_i^C \end{pmatrix}.$$

   (b) Let $j = \{1, \ldots, n | r_j = \max\{r_1, \ldots, r_n\}\}$ and set $\mathbf{D}^{(2)} = \mathbf{D}_j^{(1)}$.
   (c) For $i = 1, \ldots, n+1$, calculate

$$r_i = \tilde{U}(\boldsymbol{\delta}_i^{(3)}),$$

   where $\boldsymbol{\delta}_i^{(3)} = \text{vec}(\mathbf{D}_i^{(3)})$ and

$$\mathbf{D}_i^{(3)} = \begin{pmatrix} \mathbf{d}_1^{(2)} \\ \vdots \\ \mathbf{d}_{i-1}^{(2)} \\ \mathbf{d}_{i+1}^{(2)} \\ \vdots \\ \mathbf{d}_{n+1}^{(2)} \end{pmatrix}.$$

(d) Let $j = \{1, \ldots, n+1 | r_j = \max\{r_1, \ldots, r_{n+1}\}\}$.

(e) Set $\mathbf{D}^C = \mathbf{D}_j^{(3)}$.

6. Repeat step 5 N_{II} times until convergence.

The above algorithm is very general. In Section 3 we describe how it can be implemented by using Monte Carlo integration and Gaussian process emulation.

3 Methodology

3.1 Monte Carlo Integration

In this section we describe how Monte Carlo integration can be used to approximate $U(\boldsymbol{\delta})$.

Suppose $\{\boldsymbol{\psi}_i, \mathbf{y}_i\}_{i=1}^B$ is a sample generated from the joint distribution of $\boldsymbol{\psi}$ and $\mathbf{y}$, given by $\pi(\boldsymbol{\psi}, \mathbf{y}|\boldsymbol{\delta})$, then the Monte Carlo integration approximation to $U(\boldsymbol{\delta})$ is

$$\hat{U}_B(\boldsymbol{\delta}) = \frac{1}{B}\sum_{i=1}^{B} u(\boldsymbol{\psi}_i, \mathbf{y}_i, \boldsymbol{\delta}).$$

The value of $\hat{U}_B(\boldsymbol{\delta})$ is an unbiased estimator of $U(\boldsymbol{\delta})$ and converges to $U(\boldsymbol{\delta})$ by the law of large numbers. If $|\mathrm{E}_{\boldsymbol{\psi},\mathbf{y}}\left(u(\boldsymbol{\psi}, \mathbf{y}, \boldsymbol{\delta})^2\right)| < \infty$, the variance of $\hat{U}_B(\boldsymbol{\delta})$ is given by

$$\mathrm{var}\left(\hat{U}_B(\boldsymbol{\delta})\right) = \frac{1}{B}\int_{\mathcal{Y}}\int_{\Psi}\left(u(\boldsymbol{\psi}, \mathbf{y}, \boldsymbol{\delta}) - U(\boldsymbol{\delta})\right)^2 \pi(\boldsymbol{\psi}, \mathbf{y}|\boldsymbol{\delta})\mathrm{d}\boldsymbol{\psi}\mathrm{d}\mathbf{y}. \tag{5}$$

Note from (5), that by increasing B we increase precision through a smaller variance. However the increased accuracy comes with higher computational expense.

Consider the application of $\hat{U}_B(\boldsymbol{\delta})$ in the ACE algorithm presented in Section 2. In steps 5a and 5c we set $\tilde{U}(\boldsymbol{\delta}) = \hat{U}_B(\boldsymbol{\delta})$. However, as discussed by Robert and Casella (2004, pages 203-204), $\hat{U}_B(\boldsymbol{\delta})$ is not immediately suitable for optimisation, such as the maximisation carried out in step 2. As the joint distribution of $\boldsymbol{\psi}$ and $\mathbf{y}$ depends on $\boldsymbol{\delta}$, a new sample from this distribution will have to be generated at every new value of $\boldsymbol{\delta}$ in an optimisation algorithm. Also since a new sample will be generated each time, $\hat{U}_B(\boldsymbol{\delta})$ will not be a smooth function. Geyer (1996) suggested the use of importance sampling. A sample, $\{\boldsymbol{\psi}_i, \mathbf{y}_i\}_{i=1}^B$, is generated from a distribution, G, with density $g(\boldsymbol{\psi}, \mathbf{y})$, which does not depend on $\boldsymbol{\delta}$. The expected utility is then approximated by

$$\breve{U}_B(\boldsymbol{\delta}) = \frac{1}{B}\sum_{i=1}^{B}\frac{u(\boldsymbol{\psi}_i, \mathbf{y}_i, \boldsymbol{\delta})\pi(\boldsymbol{\psi}_i, \mathbf{y}_i|\boldsymbol{\delta})}{g(\boldsymbol{\psi}_i, \mathbf{y}_i)}.$$

Now the same sample can be used throughout and $\breve{U}_B(\boldsymbol{\delta})$ will be a smooth function. However, specifying an efficient G, appropriate for the whole design space, will be difficult. Additionally using a single sample will introduce bias. For these reasons we do not pursue this idea here.

Instead the approach taken here is to construct a Gaussian process emulator for $U_i(\delta)$ using a "small" number of evaluations of the Monte Carlo integration, $\hat{U}_{i,B}(\delta)$, i.e. we create a smooth approximation to $U_i(\delta)$. We describe the Gaussian process emulator in Section 3.2.

We spend the remainder of this section describing how we specifically approximate the expectation of the utility functions discussed in Section 1.3. In all cases, suppose, as above, that $\{\boldsymbol{\psi}_i, \mathbf{y}_i\}_{i=1}^B$ is a sample generated from the joint distribution of $\boldsymbol{\psi}$ and $\mathbf{y}$.

The SIG utility function can be rewritten as

$$u^S(\boldsymbol{\theta}, \mathbf{y}, \boldsymbol{\delta}) = \log \pi(\mathbf{y}|\boldsymbol{\theta}, \boldsymbol{\delta}) - \log \pi(\mathbf{y}|\boldsymbol{\delta}),$$

where

$$\pi(\mathbf{y}|\boldsymbol{\theta}, \boldsymbol{\delta}) = \int_\Gamma \pi(\mathbf{y}|\boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\delta})\pi(\boldsymbol{\gamma}|\boldsymbol{\theta})\mathrm{d}\boldsymbol{\gamma}, \tag{6}$$

$$\pi(\mathbf{y}|\boldsymbol{\delta}) = \int_\Psi \pi(\mathbf{y}|\boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\delta})\pi(\boldsymbol{\gamma}, \boldsymbol{\theta})\mathrm{d}\boldsymbol{\psi}, \tag{7}$$

where $\pi(\mathbf{y}|\boldsymbol{\theta}, \boldsymbol{\gamma}, \boldsymbol{\delta}) = \pi(\mathbf{y}|\boldsymbol{\psi}, \boldsymbol{\delta})$ and Γ is the parameter space for $\boldsymbol{\gamma}$. Typically, if there exists nuisance parameters, both of these terms will be analytically intractable. If there are no nuisance parameters, then (6) is $\pi(\mathbf{y}|\boldsymbol{\theta}, \boldsymbol{\delta}) = \pi(\mathbf{y}|\boldsymbol{\psi}, \boldsymbol{\delta})$ which is usually available in closed form. However we can approximate both (6) and (7) using Monte Carlo integration by

$$\hat{\pi}_{\tilde{B}}(\mathbf{y}|\boldsymbol{\theta}, \boldsymbol{\delta}) = \frac{1}{\tilde{B}} \sum_{i=1}^{\tilde{B}} \pi(\mathbf{y}|\boldsymbol{\theta}, \tilde{\boldsymbol{\gamma}}_i, \boldsymbol{\delta}),$$

$$\hat{\pi}_{\tilde{B}}(\mathbf{y}|\boldsymbol{\delta}) = \frac{1}{\tilde{B}} \sum_{i=1}^{\tilde{B}} \pi(\mathbf{y}|\tilde{\boldsymbol{\theta}}_i, \tilde{\boldsymbol{\gamma}}_i, \boldsymbol{\delta}),$$

where $\left\{\tilde{\boldsymbol{\theta}}_i, \tilde{\boldsymbol{\gamma}}_i\right\}_{i=1}^{\tilde{B}}$ is a sample generated from the prior distribution of $\boldsymbol{\psi} = (\boldsymbol{\theta}, \boldsymbol{\gamma})$. The tilde symbol ($\sim$) distinguishes this sample from $\{\boldsymbol{\psi}_i, \mathbf{y}_i\}_{i=1}^B$. We plug-in this approximation to the Monte Carlo integration approximation to $U^S(\boldsymbol{\delta})$ to yield the following nested Monte Carlo integration approximation to the expected SIG utility:

$$\hat{U}_B(\boldsymbol{\delta}) = \frac{1}{B} \sum_{i=1}^{B} \left(\log \hat{\pi}_{\tilde{B}}(\mathbf{y}_i|\boldsymbol{\theta}_i, \boldsymbol{\delta}) - \log \hat{\pi}_{\tilde{B}}(\mathbf{y}_i|\boldsymbol{\delta})\right).$$

Note that $\log \hat{\pi}_{\tilde{B}}(\mathbf{y}|\boldsymbol{\theta}, \boldsymbol{\delta})$ and $\log \hat{\pi}_{\tilde{B}}(\mathbf{y}|\boldsymbol{\delta})$ are not unbiased estimators of $\log \pi(\mathbf{y}|\boldsymbol{\theta}, \boldsymbol{\delta})$ and $\log \pi(\mathbf{y}|\boldsymbol{\delta})$, respectively, so $\hat{U}_B(\boldsymbol{\delta})$ is no longer an unbiased estimator of $U^S(\boldsymbol{\delta})$. However (under certain conditions) the bias is of order $\tilde{B}^{-1}$ (Oehlert, 1992) so will be negligible for large $\tilde{B}$.

Now consider the NSEL utility function given by (4). The expectation $\mathrm{E}(\theta_j|\mathbf{y}, \boldsymbol{\delta})$ will, typically, not be analytically tractable. Its value can be approximated via importance sampling

$$\hat{\mathrm{E}}_{\tilde{B}}(\theta_j|\mathbf{y}, \boldsymbol{\delta}) = \frac{\sum_{i=1}^{\tilde{B}} \tilde{\theta}_{ij}\pi(\mathbf{y}|\tilde{\boldsymbol{\theta}}_i, \tilde{\boldsymbol{\gamma}}_i, \boldsymbol{\delta})}{\sum_{i=1}^{\tilde{B}} \pi(\mathbf{y}|\tilde{\boldsymbol{\theta}}_i, \tilde{\boldsymbol{\gamma}}_i, \boldsymbol{\delta})},$$

where $\left\{\tilde{\boldsymbol{\theta}}_i, \tilde{\boldsymbol{\gamma}}_i\right\}_{i=1}^{\tilde{B}}$ is a sample generated from the prior distribution of $\boldsymbol{\psi}$, and $\tilde{\theta}_{ij}$ is the jth element of $\tilde{\boldsymbol{\theta}}_i$. This yields the following nested Monte Carlo integration approximation

$$\hat{U}_B^V(\boldsymbol{\delta}) = -\frac{1}{B} \sum_{i=1}^{B} \sum_{j=1}^{p} \left(\theta_{ij} - \hat{\mathrm{E}}_{\tilde{B}}(\theta_j|\mathbf{y}_i, \boldsymbol{\delta})\right)^2,$$

where θ_{ij} is the jth element of $\boldsymbol{\theta}_i$. Similar to the approximation to the expected SIG, this will not be unbiased since

$$\mathrm{E}_{\mathbf{y}}\left(\hat{\mathrm{E}}_{\tilde{B}}(\theta_j|\mathbf{y},\boldsymbol{\delta})^2\right) \neq \mathrm{E}(\theta_j|\mathbf{y},\boldsymbol{\delta})^2,$$

but again, we assume that this bias is negligible for large $\tilde{B}$.

For both the expected SIG and NSEL utility functions, we set $\tilde{B} = B$ for all examples, although this does not necessarily have to be the case.

Unbiased Monte Carlo integration approximations to the pseudo-Bayesian expected utility functions are given by

$$\hat{U}_B^D(\boldsymbol{\delta}) = \frac{1}{B}\sum_{i=1}^{B} \log|\mathcal{I}(\boldsymbol{\psi}_i;\boldsymbol{\delta})|,$$

and

$$\hat{U}_B^A(\boldsymbol{\delta}) = -\frac{1}{B}\sum_{i=1}^{B} \mathrm{tr}\left(\mathcal{I}(\boldsymbol{\psi}_i;\boldsymbol{\delta})^{-1}\right),$$

respectively, which only require a sample to be generated from the prior distribution.

3.2 Gaussian Process Emulators

In this section we describe how to construct the Gaussian process emulator approximation to $U_i(\delta)$ to use in step 2a of the ACE algorithm.

At the ith iteration of step 2 of the ACE algorithm, let $\zeta = \{\delta_1, \ldots, \delta_Q\} \in \mathcal{D}_i$ and set $\bar{u}_j = \hat{U}_{i,B}(\delta_j)$, for $j = 1, \ldots, Q$. Let $\bar{m}$ and $\bar{s}$ be the sample mean and standard deviation of $(\bar{u}_1, \ldots, \bar{u}_Q)$ and

$$z_j = \frac{\bar{u}_j - \bar{m}}{\bar{s}},$$

for $j = 1, \ldots, Q$. If $\mathbf{z} = (z_1, \ldots, z_Q)$, the Gaussian process assumes that

$$\mathbf{z}|\eta, \rho \sim \mathrm{N}(\mathbf{0}, \mathbf{A}),$$

where $\mathbf{A}$ is a $Q \times Q$ positive-definite matrix with jlth element

$$A_{jl} = \exp\left(-\rho\left(\delta_j - \delta_l\right)^2\right) + \eta I(j = l),$$

$I(E)$ is the indicator function for the event E, and, $\eta > 0$ and $\rho > 0$ are unknown parameters. Let $\bar{u}_0 = \hat{U}_{i,B}(\delta)$, for $\delta \in \mathcal{D}_i$, be the value of the Monte Carlo approximation we wish to predict. It can be shown that

$$\bar{u}_0|\eta, \rho \sim \mathrm{N}(m_0(\delta), v_0(\delta)),$$

where

$$\begin{aligned} m_0(\delta) &= \bar{m} + \bar{s}\mathbf{a}^T\mathbf{A}^{-1}\mathbf{z}, \\ v_0(\delta) &= \bar{s}^2\left(1 + \eta - \mathbf{a}^T\mathbf{A}^{-1}\mathbf{a}\right), \end{aligned}$$

and $\mathbf{a}$ is a $Q \times 1$ vector with jth element $a_j = \exp\left(-\rho\left(\delta - \delta_j\right)^2\right)$. We use $m_0(\delta)$ as a point prediction of $U_i(\delta)$, i.e. in step 2b we set $\tilde{U}_i(\delta) = m_0(\delta)$.

The parameters η and ρ are unknown. A fully Bayesian approach can be taken but we use the plug-in approach (Kennedy and O 'Hagan, 2001) and replace them with fixed estimates. For convenience, we estimate their values using maximum likelihood and the Fisher scoring method (see, for example, Millar, 2011, page 104).

An important decision is the choice of ζ. We could actually find the optimal design which minimises the predictive variance $v_0(\delta)$, however implementing this at every step of the coordinate exchange algorithm would be infeasible and would require knowledge about η and ρ. Instead we let ζ be a one-dimensional Latin hypercube design (see, for example, Fang et al., 2006, Chapter 2). These types of design are commonly used in the computer experiments literature to approximate computationally expensive functions. A new ζ is randomly generated at every iteration of the ACE algorithm.

We must also specify Q, the number of Monte Carlo integration evaluations to use each time. As Q increases, we must obviously use more Monte Carlo integration evaluations which will be computationally expensive. Also to fit a Gaussian process emulator via maximum likelihood requires multiple inversions of $\mathbf{A}$, a $Q \times Q$ matrix. As Q becomes large this can also become computationally expensive since it requires $\mathcal{O}(Q^3)$ operations. In all examples, unless otherwise stated, we use $Q = 20$. This satisfies the rule of thumb for Gaussian process emulators, advocated by Loeppky et al. (2009), that Q should be at least ten times the number of dimensions of the input space (which in this case is one)

3.3 Further Implementation Details

3.3.1 Initial design

In step 1 an initial design, $\boldsymbol{\delta}^0$, is required. Unless otherwise stated, in all implementations of the ACE algorithm, a Latin hypercube design of appropriate dimensions is used.

3.3.2 Maximisation method

In step 2b we maximise $\tilde{U}_i(\delta) = m_0(\delta)$ for $\delta \in \mathcal{D}_i$. In the corresponding step of the algorithm used by Gotwalt et al. (2009), the so-called "Brent" method (Brent, 1973) is used. We found that this method is susceptible to converging to a local maximum, not the global maximum. Since this is a one-dimensional maximisation, we evaluate $m_0(\delta)$ over a very fine grid, of size M, of uniformly generated points of $\mathcal{D}_i$. Throughout, we use $M = 10000$, and, despite this large value, found that $m_0(\delta)$ could be evaluated for every value in the grid in fractions of a second.

3.3.3 Adequacy of Gaussian process emulator

A Gaussian process emulator, similar to all statistical models, can fit inadequately with the result that δ_i^* is a poor approximation to arg $\max_{\delta \in \mathcal{D}_i} U_i(\delta)$ and the algorithm moving to an inferior design. Bastos and O'Hagan (2009) developed diagnostic procedures to assess the adequacy of Gaussian process emulators based on additional test evaluations of $\hat{U}_{i,B}(\delta)$. However the interpretation of these procedures is subjective which will be difficult to implement automatically within the ACE algorithm. Instead, we propose to include a comparison procedure. Replace step 2c by the following two steps

(c) Let $\boldsymbol{\delta}^* = \left(\delta_1^C, \ldots, \delta_{i-1}^C, \delta_i^*, \delta_{i+1}^C, \ldots, \delta_q^C\right)$.

(d) Compare the current design $\boldsymbol{\delta}^C$ and the candidate design $\boldsymbol{\delta}^*$. If accepted, set $\boldsymbol{\delta}^C = \boldsymbol{\delta}^*$; otherwise $\boldsymbol{\delta}^C$ remains unchanged.

Consider the comparison procedure. A simple check of whether $\hat{U}_B(\boldsymbol{\delta}^*) > \hat{U}_B(\boldsymbol{\delta}^C)$ will not suffice due to the stochastic nature of Monte Carlo integration.

Instead, note that we are assessing whether

$$\mathrm{E}\left(u(\boldsymbol{\psi}, \mathbf{y}, \boldsymbol{\delta}^*)\right) > \mathrm{E}\left(u(\boldsymbol{\psi}, \mathbf{y}, \boldsymbol{\delta}^C)\right), \tag{8}$$

where we can generate a sample from $u(\boldsymbol{\psi}, \mathbf{y}, \boldsymbol{\delta}^*)$ and from $u(\boldsymbol{\psi}, \mathbf{y}, \boldsymbol{\delta}^C)$. To answer this question, we perform a Bayesian hypothesis test of (8) using the two samples and accept the candidate design, $\boldsymbol{\delta}^*$, with the associated posterior probability, p^*, of this hypothesis. Specifically, let

$$\begin{aligned} u_i^C &= u(\boldsymbol{\psi}_i, \mathbf{y}_i, \boldsymbol{\delta}^C), \\ u_i^* &= u(\boldsymbol{\psi}_i^*, \mathbf{y}_i^*, \boldsymbol{\delta}^*), \end{aligned}$$

where $\{\boldsymbol{\psi}_i, \mathbf{y}_i\}_{i=1}^B$ and $\{\boldsymbol{\psi}_i^*, \mathbf{y}_i^*\}_{i=1}^B$ are two samples generated from the joint distribution of $\boldsymbol{\psi}$ and $\mathbf{y}$. We assume that, independently,

$$\begin{aligned} u_i^C &\sim \mathrm{N}\left(b_1, v\right), \\ u_i^* &\sim \mathrm{N}\left(b_1 + b_2, v\right), \end{aligned}$$

for $i = 1, \ldots, B$. The hypothesis of (8) is now equivalent to $b_2 > 0$, and under non-informative priors, the posterior probability of this hypothesis being true is

$$p^* = 1 - F\left(-\frac{\sum_{i=1}^B u_i^* - \sum_{i=1}^B u_i^C}{\sqrt{2B\hat{v}}}\right),$$

where $F(\cdot)$ is the distribution function of the t-distribution with $2B-2$ degrees of freedom,

$$\hat{v} = \frac{\sum_{i=1}^B (u_i^C - \bar{u}^C)^2 + \sum_{i=1}^B (u_i^* - \bar{u}^*)^2}{2B - 2},$$

and $\bar{u}^C$ and $\bar{u}^*$ are the sample means of the u_i^C's and u_i^*'s, respectively.

This procedure relies on the assumption of normality of the u_i^C and u_i^* for $i = 1, \ldots, B$. This may not be a reasonable assumption for all cases. However we believe that this test procedure will be more robust than merely relying on the Gaussian process emulator and the assumption of normality allows analytic calculation of p^*. The idea of using hypothesis tests in the optimisation of stochastic functions is not a new one. For an example of using a classical hypothesis test in conjunction with simulated annealing, see Wang and Zhang (2006).

Additionally we also employ this comparison procedure in step 5 by replacing step 5e by

(e) Compare the candidate design $\boldsymbol{\delta}_j^{(3)} = \mathrm{vec}\left(\mathbf{D}_j^{(3)}\right)$ to the current design $\boldsymbol{\delta}^C = \mathrm{vec}\left(\mathbf{D}^C\right)$. If accepted, set $\boldsymbol{\delta}^C = \boldsymbol{\delta}_j^{(3)}$; otherwise leave $\boldsymbol{\delta}^C$ unchanged.

3.3.4 Convergence

The steps of Phases I and II are repeated N_I and N_{II} times, respectively. We found that, for all examples in Section 4, that $N_I = 20$ and $N_{II} = 100$ was sufficient for approximate convergence. We assessed approximate convergence by using a trace plot of the evaluation of $\hat{U}_B(\boldsymbol{\delta})$ for the current design at each iteration of the ACE algorithm. See Section 3.4.2 for examples of such trace plots.

3.3.5 Monte Carlo sample size

Consider the choice of Monte Carlo sample size, B. There is no requirement that the sample size, B, generated from the joint distribution of $\boldsymbol{\psi}$ and $\mathbf{y}$, used in the evaluation of $\hat{U}_B(\boldsymbol{\delta})$, be the same at every iteration of the ACE algorithm. For all implementations, unless specified otherwise, $B = 1000$ for evaluation of $\hat{U}_B(\boldsymbol{\delta})$ in step 2(a) and $B = 20000$ for all other evaluations of $\hat{U}_B(\boldsymbol{\delta})$, i.e. those used in the comparison procedures.

3.3.6 Repeating ACE

As noted by, for example, Goos and Jones (2011, pg 36), the coordinate exchange algorithm may converge to a local maximum. Therefore we repeat the algorithm N times from N different initial designs. Note that N repetitions is trivial to implement using parallel processing computing where the repetitions will run in parallel on N different processors. Each repetition will produce a design. We evaluate $\hat{U}_B(\boldsymbol{\delta})$, for each of these designs, C times, and choose the design which has the highest mean over these C evaluations. Unless otherwise stated, we use $N = C = 20$ for all of the examples in this paper.

3.3.7 Code

The algorithm outlined and proposed in Sections 2 and 3 is implemented in the `R` package `acebayes`. `R` code to reproduce the examples in Section 4 using `acebayes` is available from the authors on request. Additionally, the designs found for all examples in Section 4 are available in `acebayes` to allow fast comparison with designs found using new methodology.

3.4 Toy Examples

3.4.1 Single Poisson observation

To demonstrate the main ideas behind the algorithm presented in Sections 2 and 3, consider the following toy example. We are to make one observation, y, which is assumed to have the following distribution

$$y|\beta \sim \text{Poisson}(e^{\beta x}),$$

where $\delta = x \in [-1, 1]$ is the design variable. We assume that β has a $\text{N}(\beta_0, 1)$ prior distribution. Consider Bayesian D-optimality with the following utility function

$$\begin{aligned} u(\beta, y, x) &= \log \mathcal{I}(\beta; x), \\ &= 2 \log |x| + \beta x, \end{aligned}$$

which leads to the expected utility of

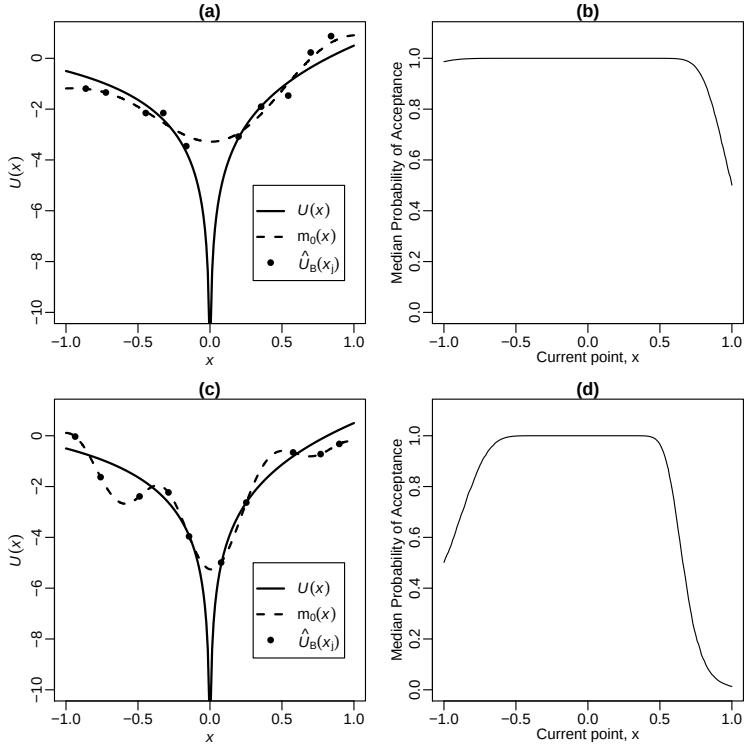
$$U(x) = 2 \log |x| + \beta_0 x.$$

It is easy to see that the optimal design, $\tilde{x}$, is given by

$$\tilde{x} = \begin{cases} 1, & \text{if } \beta_0 > 0, \\ -1, & \text{if } \beta_0 < 0, \end{cases}$$

and if $\beta_0 = 0$, then equally optimal designs are given by $x^* = \pm 1$.

Figure 1: **Plots for the toy example in Section 3.4. Figure (a) shows a plot of the Monte Carlo evaluations, $\hat{U}_B(x_j)$, against $x_j \in \zeta$. Also shown are $U(x)$ and $m_0(x)$ against x. Figure (b) shows the a plot of the median probability of accepting the candidate point against the current point, x. Figures (c) and (d) show the same as Figures (a) and (b) but for a different design, ζ.**

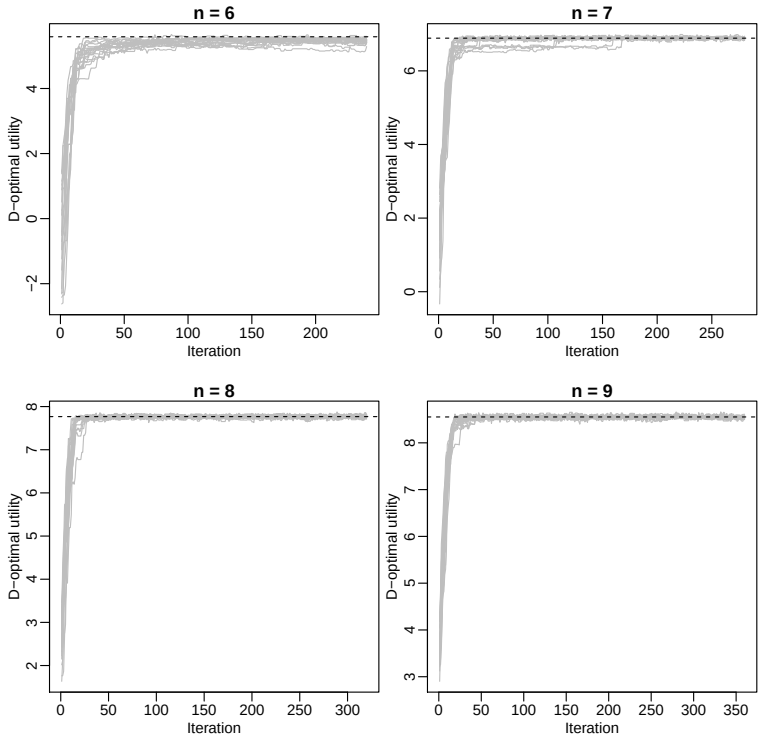


Let $\beta_0 = 0.5$ so that the optimal design is given by $x^* = 1$. We generate a design $\zeta = \{x_1, \ldots, x_Q\}$ of size $Q = 10$ and, for each x_j, evaluate

$$z_j = \hat{U}_B(x_j) = 2 \log |x_j| + \frac{x_j}{B} \sum_{i=1}^{B} \beta_i,$$

where $\{\beta_i\}_{i=1}^{B}$, with $B = 2$, is a sample generated from $\mathrm{N}(\beta_0, 1)$. Figure 1(a) shows a plot of $U(x)$ against x with the points (x_j, u_j), for $j = 1, \ldots, Q$, along with the Gaussian process emulator prediction, $m_0(x)$, as a dashed line. Clearly $m_0(x)$ is maximised at $x = 1$, so this becomes the candidate point to be compared to the current point. Figure 1(b) shows the median (over repeated sampling from the integrand of $\hat{U}_B$) posterior probability of accepting this candidate point plotted against current point x. This probability of acceptance is very close to one for nearly all values of x except for when x becomes close to the optimal design $\tilde{x}$ where the the probability reduces to 1/2. Now suppose we had generated a different design ζ at which to evaluate $\hat{U}_B(x)$. This situation is depicted in Figure 1(c). The Gaussian process emulator could be considered to be inadequate. The estimate of the parameter η is too small resulting in $m_0(x)$ practically interpolating the Monte Carlo integration evaluations. Here $m_0(x)$ is maximised at $x = -1$ and this becomes the candidate point. Again Figure 1(d) shows the median posterior probability of acceptance plotted against current point x. Now we are only likely to accept this candidate if the current point is in areas of low expected utility (i.e. between -0.5 and 0.5). Crucially, we are likely to reject the candidate if the current point is close to the optimal design of $\tilde{x} = 1$ where the probability drops to zero.

Figure 2: **Trace plots of $\hat{U}_B^D(\boldsymbol{\delta}^C)$ at each iteration of the ACE algorithm for each value of n. The dotted horizontal line indicates the optimal value of $U^D(\boldsymbol{\delta})$ as found by Box and Draper (1971).**



3.4.2 Second-order response surface in two factors

The first example we consider is a second-order response surface model. Specifically, for $i = 1, \ldots, n$, where n is the number of runs, we assume that

$$y_i \sim \mathrm{N}\left(\mu_i, \sigma^2\right),$$

where

$$\begin{aligned} \mu_i &= \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{1i}^2 + \beta_4 x_{2i}^2 + \beta_5 x_{1i} x_{2i}, \\ &= \mathbf{x}_i^T \boldsymbol{\beta}, \end{aligned}$$

and $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_5)$. We consider the SIG utility function under non-informative priors. In this case, the expected SIG utility is analytically tractable and the corresponding optimal design is equivalent to the classical D-optimal design. This is found by maximising the following function

$$U^D(\boldsymbol{\delta}) = \log |\mathbf{X}^T \mathbf{X}|,$$

where $\mathbf{X}$ is the $n \times 6$ matrix with ith row given by $\mathbf{x}_i$ and $\boldsymbol{\delta}$ is the $n \times 2$ matrix with ith row given by (x_{1i}, x_{2i}). The design space for each element of $\boldsymbol{\delta}$ is the interval $[-1, 1]$.

Box and Draper (1971) found D-optimal designs for this problem, analytically, for $n = 6, 7, 8$ and 9. To assess the efficacy of the ACE algorithm we attempt to recover these D-optimal designs. To do this we use the ACE algorithm with the following stochastic approximation to $U^D(\boldsymbol{\delta})$,

$$\hat{U}_B^D(\boldsymbol{\delta}) = \frac{1}{B} \sum_{i=1}^{B} u_i,$$

where $u_i = \log |\mathbf{X}^T \mathbf{X}| + \epsilon_i$, with $\epsilon_i \overset{\text{iid}}{\sim} \mathrm{N}(0, 1)$ and $B = 1000$.

Table 1: **Minimum, median and maximum for the $N = 20$ D-efficiencies (%) for each number of runs, n.**

	Number of runs, n			
	6	7	8	9
Minimum	96.5	99.2	99.4	99.6
Median	98.6	99.9	99.9	99.9
Maximum	99.7	100.0	100.0	99.9

To compare the true and approximate optimal designs we use D-efficiency (see, for example, Atkinson et al. 2007, pg 151). Recall from Section 3.3, that we repeat the ACE algorithm $N = 20$ times from different initial designs. Table 1 shows the minimum, median and maximum D-efficiencies over the N replications for each of the four values for n. For each n, all designs obtained are more than 96% efficient and the "best" designs are all more than 99.5% efficient. For each value of n and each repetition, Figure 2 shows a trace plot of the approximate expected SIG utility at each iteration. From these plots, we can identify how the algorithm has approximately converged.

4 Examples

4.1 Compartmental model

Compartmental models are used in Pharmokinetics to study how materials flow through an organism. They have been used extensively to demonstrate optimal design methodology (see, for example, Atkinson et al., 1993; Gotwalt et al., 2009; Ryan et al., 2014). A drug is administered to an individual or animal and then the amount present at a certain body location is measured at a set of n pre-determined sampling times (in hours). There is one design variable: sampling time. Therefore $\boldsymbol{\delta}$ is an $n \times 1$ matrix containing the sampling times: $t_1, \ldots, t_n$.

The general statistical model is as follows. Let y_i denote the amount of drug measured at time t_i and we assume that, independently,

$$y_i \sim \mathrm{N}\left(c(\boldsymbol{\theta})\mu(\boldsymbol{\theta}; t_i), \sigma^2 v(\boldsymbol{\theta}; t_i)\right),$$

where $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3)$ are parameters and of interest, $\sigma^2 > 0$ is a nuisance parameter, c and v are specified functions depending on the application, and

$$\mu(\boldsymbol{\theta}; t_i) = \exp\left(-\theta_1 t_i\right) - \exp\left(-\theta_2 t_i\right).$$

4.1.1 Bayesian D-optimality

Atkinson et al. (1993) and Gotwalt et al. (2009) studied generating Bayesian D-optimal designs for this problem, where $n = 18$,

$$\begin{aligned} c(\boldsymbol{\theta}) &= \theta_3, \\ v(\boldsymbol{\theta}; t_i) &= 1. \end{aligned}$$

The prior distribution for $\boldsymbol{\theta}$ is such that the elements are independent, with

$$\begin{aligned} \theta_1 &\sim \mathrm{U}[0.01884, 0.09884], \\ \theta_2 &\sim \mathrm{U}[0.298, 8.298], \end{aligned}$$

and where θ_3 has a prior point mass at 21.8. The log-determinant of the Fisher information only depends on σ^2 linearly, so its prior does not affect the Bayesian D-optimal design. The design space for t_1 is $[0, 24]$ hours and for t_j is $[t_{j-1}, 24]$. However this constraint does not alter the operation of the ACE algorithm.

Relative to the optimal design found by Atkinson et al. (1993), the mean relative D-efficiencies (over the twenty Monte Carlo integration approximations) of the ACE and Gotwalt et al. (2009) designs are 99.9 and 99.6 (to 1 decimal place), respectively. This shows that the utility of the ACE, Atkinson et al. (1993) and Gotwalt et al. (2009) designs are very similar.

4.1.2 Shannon information gain

We now move onto a slightly more challenging problem involving the compartmental model studied by Ryan et al. (2014). In this case,

$$\begin{aligned} c(\boldsymbol{\theta}) &= \frac{D}{\theta_3}\frac{\theta_2}{\theta_2 - \theta_1}, \\ v(\boldsymbol{\theta}; t_i) &= \left(1 + \frac{\tau^2}{\sigma^2} c(\boldsymbol{\theta})^2 \mu(\boldsymbol{\theta}; t_i)^2\right), \end{aligned}$$

where $D = 400$, $\sigma^2 = 0.1$, and $\tau^2 = 0.01$. The prior distribution for $\boldsymbol{\theta}$ is such that the elements are independent. Each element of the parameters of interest, θ_j, has a log-normal prior distribution where the mean and variance, on the log scale, are M_j and 0.05, respectively, where

$$\begin{aligned} M_1 &= \log(0.1), \\ M_2 &= \log(1), \\ M_3 &= \log(20). \end{aligned}$$

There is a further constraint on the times, $t_1, \ldots, t_n$, where

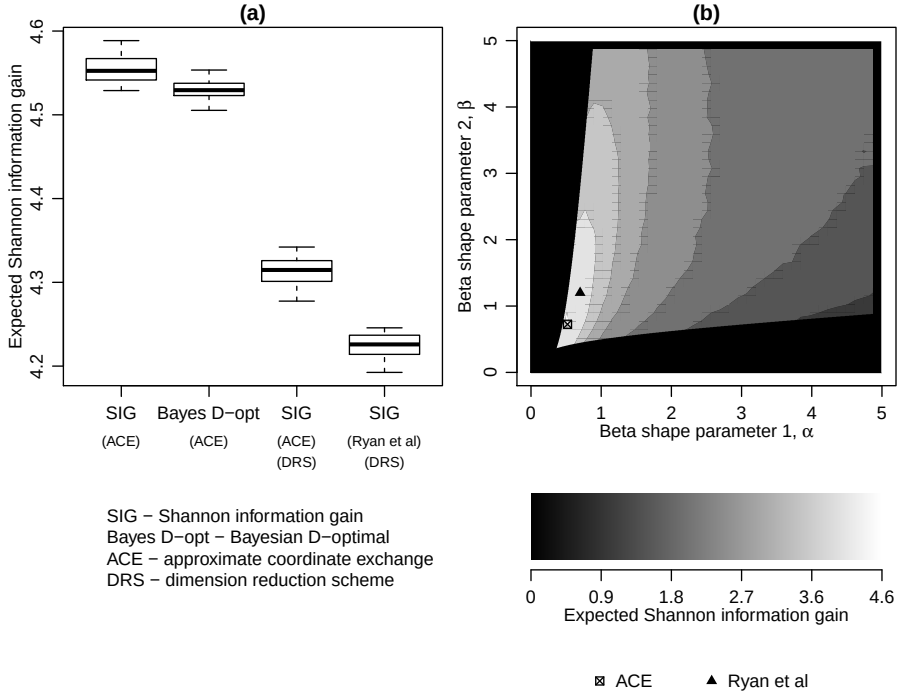
$$\max_{i,j=1,\ldots,n} |t_i - t_j| \leq 0.25,$$

i.e., the sampling times have to be at least fifteen minutes apart. This is easy to incorporate into the ACE algorithm. In Step 2, we maximise $m_0(\delta)$ over the points in $\mathcal{D}_i$ that satisfy the constraint. Phase II of the ACE algorithm is then omitted since we do not want to replicate sampling times.

Ryan et al. (2014) used their dimension-reduction schemes (DRS), in conjunction with the Muller (1999) simulation approach, to find an optimal design, with $n = 15$ times, under the SIG utility function (among other utility functions). They found that the Beta DRS yielded the optimal design. The Beta DRS involves setting the sampling times, $t_1, \ldots, t_n$, to be the scaled percentiles of a Beta(α, β) distribution. This means the dimensionality of the design problem has been reduced from n to 2, i.e., specifying α and β. We use the ACE algorithm to find three designs with which to compare to the design of Ryan et al. (2014):

1. SIG - the design found using the ACE algorithm under the SIG utility function;
2. Bayes D-opt - the design found using the ACE algorithm under the Bayesian D-optimality utility function;

Figure 3: **(a) Boxplots of twenty evaluations of $\hat{U}_B^S$ for the designs found using the ACE algorithm under SIG (unrestricted and using the Beta DRS) and under Bayesian D-optimality, and the Ryan et al. (2014) design which maximises expected SIG using the Beta DRS. (b) Plots of the Beta shape parameters where the shade of the plotting character indicates the value of $\hat{U}_B^S$ of the corresponding design. Also shown are the values of the shape parameters for the Ryan et al. (2014) design and the DRS design found using the ACE algorithm.**



3. SIG (DRS) - the design found using the ACE algorithm under the SIG utility function, where the Beta DRS has been used.

Figure 3(a) shows boxplots of twenty evaluations of $\hat{U}_B^S$ for each of the three designs listed above and the Ryan et al. (2014) design. Figure 3(a) confirms what me might expect: that by not using a DRS we obtain higher values of the expected SIG. The Bayesian D-optimal design, for this example, provides a reasonable approximation to the optimal design under SIG. Now consider the DRS designs. The ACE algorithm finds a design with higher expected utility than the Ryan et al. (2014) design. To investigate this further we generated 40000 pairs of Beta shape parameters, (α, β), from $[0, 5]^2$. For each pair we found the corresponding design and evaluated $\hat{U}_B^S$. Figure 3(b) shows a plot of β against α, where the shade of the plotting character indicates the value of $\hat{U}_B^S$. A black plotting character indicates an expected SIG of zero, whereas lighter shades indicate higher expected SIG. Points that do not satisfy the constraint that the sampling times need to be fifteen minutes apart are plotted in black. Also plotted are the Beta shape parameters corresponding to the Ryan et al. (2014) design and the ACE design. Clearly both designs are located in a region of high expected utility but the location of the ACE design confirms the conclusion from Figure 3(b): that it has higher expected utility.

4.2 Logistic regression in four factors

Consider a first-order logistic regression model in four factors where the responses will be observed in G groups of m runs, i.e. $n = Gm$. Specifically let y_{ij} be the jth response

from the ith group ($j = 1, \ldots, m$ and $i = 1, \ldots, G$), where

$$y_{ij} \sim \text{Bernoulli}(\rho_{ij}),$$

with

$$\begin{aligned} \log\left(\frac{\rho_{ij}}{1-\rho_{ij}}\right) &= \beta_0 + \omega_{i0} + (\beta_1 + \omega_{i1})x_{1i} + (\beta_2 + \omega_{i2})x_{2i} + (\beta_3 + \omega_{i3})x_{3i} + (\beta_4 + \omega_{i4})x_{4i}, \\ &= \mathbf{x}_{ij}^T(\boldsymbol{\beta} + \boldsymbol{\omega}_i). \end{aligned} \tag{9}$$

In (9), $\boldsymbol{\beta} \in \mathbb{R}^5$ are the parameters of interest and $\boldsymbol{\omega}_i \in \mathbb{R}^5$ (for $i = 1, \ldots, G$) are the group-specific (or "random effects") nuisance parameters. Let $\mathbf{X}$ be the $n \times 5$ matrix given by

$$\mathbf{X} = \begin{pmatrix} \mathbf{X}_1 \\ \vdots \\ \mathbf{X}_G \end{pmatrix},$$

where $\mathbf{X}_i$ is the $m \times 5$ matrix with jth row given by $\mathbf{x}_{ij}$, for $i = 1, \ldots, G$. Finally, $\mathbf{D}$ is the $n \times 4$ matrix given by the last four columns of $\mathbf{X}$ and the design space for each element is the interval $[-1, 1]$.

The prior distribution for $\boldsymbol{\beta}$ is such that the elements are independent and distributed as follows

$$\begin{array}{lll} \beta_0 \sim \text{U}[-3,3], & \beta_1 \sim \text{U}[4,10], \\ \beta_2 \sim \text{U}[5,11], & \beta_3 \sim \text{U}[-6,0], \\ \beta_4 \sim \text{U}[-2.5,3.5]. \end{array}$$

We consider two different prior distributions for $\boldsymbol{\omega} = (\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_G)$. Firstly, a prior point mass at $\boldsymbol{\omega} = \mathbf{0}$, thus resulting in a standard logistic regression model where we assume that the G groups are homogeneous. Secondly, a hierarchical prior distribution in which the $\boldsymbol{\omega}_i$ are independent and identically distributed with the rth element having the following distribution

$$\omega_{ir} \sim \text{U}[-s_r, s_r],$$

for $r = 1, \ldots, 5$, with $s_r > 0$ also unknown. The prior distribution for each s_r is assumed to have density

$$\pi(s_r) = \frac{2(L_r - s_r)}{L_r^2},$$

i.e. a triangular distribution, where $(L_1, \ldots, L_5) = (3, 3, 3, 1, 1)$.

4.2.1 Standard Logistic Regression

Generating Bayesian D-optimal designs for the standard logistic regression model (with the same prior distribution for $\boldsymbol{\beta}$) for $n = 16$ and $n = 48$ runs was considered by Woods et al. (2006) and Gotwalt et al. (2009). We begin by comparing the ACE design under Bayesian D-optimality to those of Woods et al. (2006) and Gotwalt et al. (2009).

Gotwalt et al. (2009) published their 16-run optimal design but we also coded their method to find an alternative design. Thus, there are four designs to compare for $n = 16$. The designs are compared by computing the average (over twenty evaluations of $\hat{U}_B(\boldsymbol{\delta})$) relative D-efficiency, relative to the ACE design. These are shown in Table 2. For both values of n, the ACE and Gotwalt et al. (2009) designs (coded by ourselves) perform very similarly, and these are about 10% more efficient than the design of Woods et al. (2006).

Table 2: **Average relative D-efficiencies (relative to the ACE design) of the designs of Woods et al. (2006) and Gotwalt et al. (2009) for logistic regression for $n = 16$ and $n = 48$ runs. For $n = 16$ runs, we use the design published in Gotwalt et al. (2009) and a design found by coding their method ourselves.**

Design	$n = 16$	$n = 48$
ACE	100.0	100.0
Gotwalt et al. (2009) (published)	82.0	-
Gotwalt et al. (2009) (coded ourselves)	99.9	101.3
Woods et al. (2006)	93.3	91.0

Now consider generating optimal designs for the logistic regression model under expected SIG. We generate these designs under a range of different numbers of runs, n, from 6 to 48. For each n, the initial design is the final design found from a previous implementation of the ACE algorithm but under an approximation to the Bayesian D-optimality objective function. We now compare the optimal designs under expected SIG to the Bayesian D-optimal designs. Figure 4(a) shows boxplots of $C = 20$ evaluations of $\hat{U}_B^S$ for the optimal designs under SIG and Bayesian D-optimality. As n increases, the difference between the two different optimal designs, in terms of expected Shannon information gain, decreases. This follows from the result (see, e.g. Gelman et al., 2014, pages 585-588) that, as n increases, under certain regularity conditions, the posterior distribution will converge to a normal distribution and, therefore, the approximation described by Chaloner and Verdinelli (1995) will become more accurate.

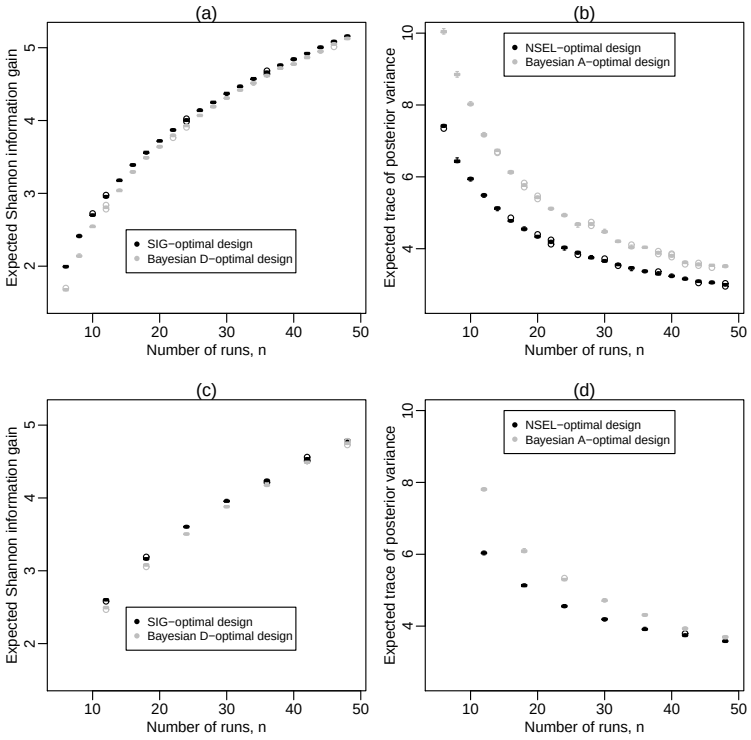
Now consider generating optimal designs under expected NSEL. Under this utility function we repeat the analysis from the preceding section but compare against Bayesian A-optimal designs. Figure 4(b) shows boxplots of twenty evaluations of $-\hat{U}_B^V$ (i.e. Monte Carlo approximation to the expectation of the trace of the posterior variance) for the optimal designs under NSEL and Bayesian A-optimality. Similar to designs under SIG, the difference between the two designs decreases as n increases.

4.2.2 Hierarchical Logistic Regression

For hierarchical logistic regression, we repeat the above exercise using the ACE algorithm to find optimal designs under the same four utility functions as for standard logistic regression. For the SIG utility, due to the computational expense of evaluating $\hat{U}_B^S$, we reduce B to 1000 for the comparison procedure in the ACE algorithm.

Figures 4(c) and (d) shows boxplots of twenty evaluations of $U_B^S(\boldsymbol{\delta})$ and $-U_B^V(\boldsymbol{\delta})$ for the optimal designs found under the SIG and Bayesian D-optimal utility functions, and the NSEL and Bayesian A-optimal utility functions, respectively. Note from comparing Figures 4(a) and (c) how we expect to gain less Shannon information in the presence of group-specific parameters under the hierarchical logistic regression model due to the extra uncertainty involved. A similar conclusion can be drawn from Figures 4(b) and (d) where the expected prior variance is higher under the hierarchical logistic regression model. Similar to the previous section, the difference between the pseudo-Bayesian designs and the fully Bayesian designs decreases as n increases.

Figure 4: **Plots (a) and (b) refer to standard logistic regression and (c) and (d) to hierarchical logistic regression. (a) and (c) show boxplots of twenty evaluations of $\hat{U}_B^S$ for the optimal designs under SIG and Bayesian D-optimality, and (b) and (d) show boxplots of twenty evaluations of $-\hat{U}_B^V$ for the optimal designs under NSEL and Bayesian A-optimality.**



4.3 Binomial Regression under Model Uncertainty

Table 3: **Beetle mortality data of Bliss (1935).**

Dose, x_i	Number exposed, m_i	Number killed, y_i
1.6907	59	6
1.7242	60	13
1.7552	62	18
1.7842	56	28
1.8113	63	52
1.8369	59	52
1.8610	62	61
1.8839	60	60

In this section we consider optimal experimental design under model uncertainty. Consider the beetle mortality data of Bliss (1935) given in Table 3. Here 481 beetles were split into eight groups. For $i = 1, \ldots, n = 8$, the ith group of m_i beetles is administered a dose, x_i, of poison and the number of beetles killed, y_i, is recorded. We follow the case study analysis of O'Hagan and Forster (2004, pages 423-433) where interest lies in producing a model-averaged posterior distribution of the quantity called lethal dose 50 (LD50). LD50 is the dose of poison required to kill 50% of the beetles. We conduct a Bayesian analysis of the beetle mortality data under model uncertainty to evaluate a model-averaged posterior distribution of LD50. We then optimally design a follow-up experiment to refine our current knowledge of LD50.

For $i = 1, \ldots, n$, with $n = 8$, let $y_i \sim \text{Binomial}(m_i, \rho_i)$, where ρ_i is the probability of death for dose x_i. Let $g(\rho_i) = \eta_i$, where $g()$ is the link function and η_i is the linear predictor. We consider six models formed by the Cartesian product of three link functions and two linear predictors. The three link functions are the logit ($g(\rho_i) = \log(\rho_i/(1-\rho_i))$), the c-log-log ($g(\rho_i) = \log(-\log(1-\rho_i))$), or the probit ($g(\rho_i) = \Phi^{-1}(\rho_i)$, where $\Phi()$ is the distribution function of the standard normal distribution). The two linear predictors are either 1st order ($\eta_i = \beta_1 + \beta_2 x_i$) or 2nd order ($\eta_i = \beta_1 + \beta_2 x_i + \beta_3 x_i^2$). Note that O'Hagan and Forster (2004) did not consider the probit link function. Let $m \in \mathcal{M} = \{1, \ldots, 6\}$ denote the model indicator and let $\boldsymbol{\beta}_m$ denote the vector of regression parameters under model m. LD50 is given by

$$z(\boldsymbol{\beta}_m) = \begin{cases} \frac{a-\beta_{m1}}{\beta_{m2}}, & \text{for } m \text{ corresponding to the 1st order linear predictor,} \\ \frac{-\beta_{m2}+\sqrt{\beta_{m2}^2-4\beta_{m3}(\beta_{m1}-a)}}{2\beta_{m3}}, & \text{otherwise,} \end{cases}$$

where

$$a = \begin{cases} \log\left(-\log\left(\frac{1}{2}\right)\right), & \text{for the c-log-log link function,} \\ 0, & \text{otherwise,} \end{cases}$$

and β_{mj} is the jth element of $\boldsymbol{\beta}_m$. Following O'Hagan and Forster (2004), we use unit information prior distributions (Ntzoufras et al., 2003) for $\boldsymbol{\beta}_m|m$ under each model. Additionally we use a uniform prior over the model space, i.e. the prior model probabilities are $\pi(m) = 1/6$, for $m \in \mathcal{M}$. The posterior model probabilities for each model are approximated by using importance sampling to evaluate the marginal likelihood of each model. The approximate posterior model probabilities, $\pi(m|\mathbf{y})$, are shown in Table 4. MCMC samples are generated under each of the six models by using the Metropolis-Hastings algorithm. We can then use the samples to produce a sample from the joint distribution $\boldsymbol{\beta}_m, m|\mathbf{y}$ of regression parameters and model indicator. From this we can derive a sample from the model-averaged posterior distribution of LD50.

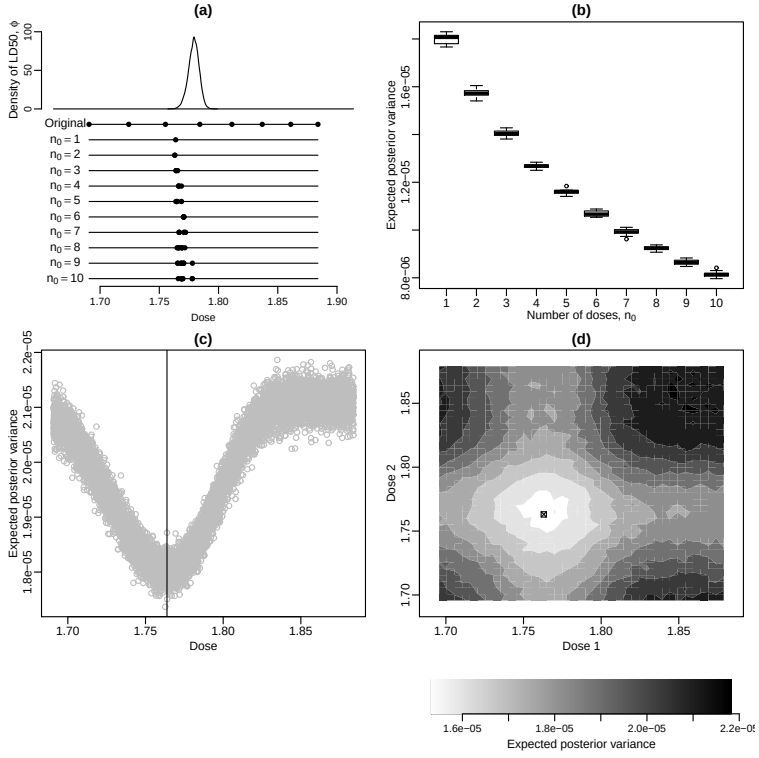
Table 4: **Approximate posterior model probabilities, $\pi(m|\mathbf{y})$, for the beetle mortality data.**

Link Function	Linear Predictor	$\pi(m\|\mathbf{y})$
Logit	1st order	0.0216
Logit	2nd order	0.0686
C-log-log	1st order	0.7580
C-log-log	2nd order	0.0612
Probit	1st order	0.0304
Probit	2nd order	0.0602

We consider the optimal design of a follow-up experiment with n_0 (potentially) new doses of poison. We administer each dose of poison to m_{0i} beetles and in each group record the number, y_{0i}, of beetles that are killed. Let $\mathbf{y}_0$ be the $n_0 \times 1$ vector of the numbers of beetles killed in the follow-up experiment. We assume that the m_{0i} is unknown and has a Poisson(λ) distribution, hence $y_{0i} \sim \text{Poisson}(\lambda\rho_i)$. We choose $\lambda = 60$ (which is consistent with the values of m_i in Table 3) and consider ten different values for n_0: $1, \ldots, 10$. Interest lies in the value of LD50, z. We use the negative squared error loss utility function for z, so the optimal design will minimise the expected posterior variance of z, i.e.

$$U^V(\boldsymbol{\delta}) = -\int_{\mathcal{Y}} \int_{\mathcal{M}} \int_{\mathcal{B}_m} \left(z(\boldsymbol{\beta}_m) - \mathrm{E}\left(z(\boldsymbol{\beta}_m)|\mathbf{y}_0, \mathbf{y}, \boldsymbol{\delta}\right)\right)^2 \mathrm{dP}_{\mathbf{y}_0, m, \boldsymbol{\beta}_m|\mathbf{y}, \boldsymbol{\delta}},$$

Figure 5: **(a) shows the values of the optimal doses for each n_0, the original experimental doses and the posterior density for LD50. (b) shows boxplots of 20 evaluations of $-\hat{U}_B^V(\boldsymbol{\delta})$ for each n_0. (c) shows $-\hat{U}_B^V(\boldsymbol{\delta})$ plotted against dose. (d) shows the two doses plotted against each other where the plotting character shade shows the value of $-\hat{U}_B(\boldsymbol{\delta})$.**



where $\boldsymbol{\delta}$ is the $n_0 \times 1$ vector of doses and $\mathcal{B}_m$ is the parameter space for model m. We can approximate $U^V(\boldsymbol{\delta})$ by

$$\hat{U}_B^V(\boldsymbol{\delta}) = -\frac{1}{B}\sum_{i=1}^{B}\left(z(\boldsymbol{\beta}_{mi}) - \hat{\mathrm{E}}\left(z(\boldsymbol{\beta}_m)|\mathbf{y}_{0i}, \mathbf{y}, \boldsymbol{\delta}\right)\right)^2$$

where $\{\boldsymbol{\beta}_{mi}, m_i, \mathbf{y}_{0i}\}_{i=1}^{B}$ is a sample generated from the joint distribution given by $\pi(\boldsymbol{\beta}_m, m, \mathbf{y}_0|\mathbf{y})$, and

$$\hat{\mathrm{E}}\left(z(\boldsymbol{\beta}_m)|\mathbf{y}_0, \mathbf{y}, \boldsymbol{\delta}\right) = \frac{\sum_{i=1}^{B} z(\tilde{\boldsymbol{\beta}}_{\tilde{m}i})\pi(\mathbf{y}_0|\tilde{\boldsymbol{\beta}}_{\tilde{m}i}, \tilde{m}_i)}{\sum_{i=1}^{B}\pi(\mathbf{y}_0|\tilde{\boldsymbol{\beta}}_{\tilde{m}i}, \tilde{m}_i)},$$

with $\left\{\tilde{\boldsymbol{\beta}}_{\tilde{m}i}, \tilde{m}_i\right\}_{i=1}^{B}$ denoting a sample generated from the joint distribution given by $\pi(\boldsymbol{\beta}_m, m|\mathbf{y})$.

We generate optimal designs under the NSEL utility function for each value of n_0 using the ACE algorithm. The plots in Figure 5 summarise the results. The points on the horizontal lines in Figure 5(a) show the location of the optimal doses for each value of n_0. Also shown in Figure 5(a) are the doses from the original design and the density of the model-averaged posterior distribution for LD50. Note that, for all values of n_0, the doses all lie in the lower tail of the posterior distribution of LD50. For the optimal design of doses, Figure 5(b) shows boxplots of 20 evaluations of $-\hat{U}_B^V(\boldsymbol{\delta})$ (i.e. giving the approximations to the expected posterior variance of LD50) for each value of n_0. Therefore we can see how the expected posterior variance decreases as n_0 increases.

We investigate the tight clustering of the optimal doses seen in Figure 5(a) for $n_0 = 1$

and 2. For each value of n_0, we generate 10000 designs uniformly in the design space. For each design, we evaluate $\hat{U}_B^V(\boldsymbol{\delta})$.

For $n_0 = 1$, Figure 5(c) shows the evaluations of $-\hat{U}_B^V(\boldsymbol{\delta})$ plotted against dose. The vertical line shows the optimal dose (found using the ACE algorithm) which clearly corresponds to the value that minimises the expected posterior variance of LD50. The variance of the model-averaged posterior distribution for LD50 is 2.10×10^{-5}. Therefore we can see from Figure 5(c) that if we had chosen a large dose (near the upper limit of the design space), we would have expected a negligible reduction in the variance of LD50.

For $n_0 = 2$, Figure 5(d) shows the two doses plotted against each other where the shade of the plotting character shows the value of the evaluation of $-\hat{U}_B(\boldsymbol{\delta})$. Also plotted is the optimal dose (found using the ACE algorithm) which appears very close to the value that maximises $U^V(\boldsymbol{\delta})$. Again, if the two doses are chosen to be near the upper limit of the design space, we are again not expected to significantly reduce the variance of LD50.

5 Discussion

In this paper we have proposed the approximate coordinate exchange (ACE) algorithm for maximising the expected utility function with respect to the unknown parameters and responses. This algorithm can be used to find optimal experimental designs and is applicable whenever it is possible to generate from the prior distribution of the model parameters and responses from the statistical model.

The ACE algorithm is demonstrated for a series of examples. Although the statistical models are relatively simple. nevertheless, finding optimal experimental designs under such models is non-trivial. For example, consider the logistic regression model in Sections 4.2. Logistic regression is, now, a trivial model to fit under the Bayesian paradigm using MCMC methods. However, optimal experimental design (classical or Bayesian) is particularly non-trivial. We feel that ours is the first attempt to find optimal designs on such a large scale.

Future work will involve extending the algorithm to statistical models described by some computationally expensive code. This includes statistical models described by the solution to a system of non-linear differential equations.

References

Amzal, B., Bois, F., Parent, E., and Robert, C. (2006), "Bayesian-Optimal Design via Interacting Particle Systems," *Journal of the American Statistical Association*, 101, 773–785.

Atkinson, A., Chaloner, K., Herzberg, A., and Juritz, J. (1993), "Experimental Designs for Properties of a Compartmental Model," *Biometrics*, 49, 325–337.

Atkinson, A., Donev, A., and Tobias, R. (2007), *Optimum Experimental Design, with SAS*, Oxford.

Bastos, L. and O'Hagan, A. (2009), "Diagnostics for Gaussian Process Emulators," *Technometrics*, 51, 425–438.

Bazaraa, M., Sherali, H., and Shetty, C. (2006), *Nonlinear Programming: Theory and Algorithms*, Wiley, 3rd ed.

Bliss, C. (1935), "The calculation of the dosage-mortality curve," *Annals of Applied Biology*, 22, 134–167.

Box, M. and Draper, N. (1971), "Factorial designs, the $|F^T F|$ criterion and some related matters," *Techometrics*, 13, 731–742.

Brent, R. (1973), *Algorithms for Minimization without Derivatives*, Prentice-Hall.

Chaloner, K. and Verdinelli, I. (1995), "Bayesian Experimental Design: A Review," *Statistical Science*, 10, 273–304.

Fang, K., Li, R., and Sudjianto, A. (2006), *Design and Modeling for Computer Experiments*, Chapman and Hall.

Gelman, A., Carlin, J., Stern, H., Dunson, D., Vehtari, A., and Rubin, D. (2014), *Bayesian Data Analysis*, Chapman and Hall, 3rd ed.

Geyer, C. (1996), "Estimation and optimization of functions," in *Markov chain Monte Carlo in Practice*, eds. Gilks, W., Richardson, S., and Spiegelhalter, D., Chapman and Hall.

Goos, P. and Jones, B. (2011), *Optimal Design of Experiments: A Case Study Approach*, Wiley.

Gotwalt, C., Jones, B., and Steinberg, D. (2009), "Fast Computation of Designs Robust to Parameter Uncertainty for Nonlinear Settings," *Technometrics*, 51, 88–95.

Hainy, M., Muller, W., and Wynn, H. (2013), "Approximate Bayesian Computational Design (ABCD), an Introduction," in *MODA 10 - Advances in Model-Oriented Design and Analysis*, eds. Ucinski, D., Atkinson, A., and Patan, M., Springer.

Hamada, M., Martz, H., Reese, C., and Wilson, A. (2001), "Finding Near-Optimal Bayesian Experimental Designs via Genetic Algorithms," *The American Statistician*, 55, 175–181.

Jones, D., Schonlau, M., and Welch, W. (1998), "Efficient global optimization of expensive black-box functions," *Journal of Global Optimization*, 13, 455–492.

Kennedy, M. C. and O 'Hagan, A. (2001), "Bayesian calibration of computer models (with discussion)," *Journal of the Royal Statistical Society, Series B*, 63, 425–464.

Lindley, D. (1956), "On the measure of information provided by an experiment," *Annals of Statistics*, 27, 986–1005.

Loeppky, J., Sacks, J., and Welch, W. (2009), "Choosing the Sample Size of a Computer Experiment: A Practical Guide," *Technometrics*, 51, 366–376.

Meyer, R. and Nachtsheim, C. (1995), "The Coordinate Exchange Algorithm for Constructing Exact Optimal Experimental Designs," *Technometrics*, 37, 60–69.

Millar, R. (2011), *Maximum Likelihood Estimation and Inference*, Wiley.

Morris, M. (2011), *Design of Experiments: An Introduction Based on Linear Models*, Chapman and Hall.

Muller, P. (1999), "Simulation-Based Optimal Design," in *Bayesian Statistics 6*, eds. Bernardo, J. M., Bayarri, M. J., Berger, J. O., Dawid, A. P., Heckerman, D., and Smith, A. F. M., Oxford.

Muller, P. and Parmigiani, G. (1996), "Optimal Design via Curve Fitting of Monte Carlo Experiments," *Journal of the American Statistical Association*, 90, 1322–1330.

Muller, P., Sanso, B., and De Iorio, M. (2004), "Optimal Bayesian Design by Inhomogeneous Markov Chain Simulation," *Journal of the American Statistical Association*, 99, 788–798.

Ntzoufras, I., Dellaportas, P., and Forster, J. J. (2003), "Bayesian Variable and Link Determination for Generalised Linear Models," *Journal of Statistical Planning and Inference*, 111, 165–180.

Oehlert, G. (1992), "A Note on the Delta Method," *The American Statistician*, 46, 27–29.

O'Hagan, A. and Forster, J. (2004), *Kendall's Advanced Theory of Statistics*, vol. 2B: Bayesian Inference, John Wiley & Sons, 2nd ed.

Robert, C. and Casella, G. (2004), *Monte Carlo Statistical Methods*, Springer-Verlag, 2nd ed.

Ryan, E., Drovandi, C., Thompson, M., and Pettitt, A. (2014), "Towards Bayesian experimental design for nonlinear models that require a large number of sampling times," *Computational Statistics and Data Analysis*, 70, 45–60.

Wang, L. and Zhang, L. (2006), "Stochastic optimization using simulated annealing with hypothesis test," *Applied Mathematics and Computation*, 174, 1329–1342.

Woods, D., Lewis, S., Eccleston, J., and Russell, K. (2006), "Designs for Generalized Linear Models With Several Variables and Model Uncertainty," *Technometrics*, 48, 284–292.