



Department for Applied Statistics
Johannes Kepler University Linz



IFAS Research Paper Series 2013-64

Likelihood-free Simulation-based Optimal Design

Markus Hainy, Werner G. Müller, Helga Wagner

May 2013

Abstract. Simulation-based optimal design techniques are a convenient tool for solving a particular class of optimal design problems. The goal is to find the optimal configuration of factor settings with respect to an expected utility criterion. This criterion depends on the specified probability model for the data and on the assumed prior distribution for the model parameters. We develop new simulation-based optimal design methods which incorporate likelihood-free approaches and utilize them in novel applications.

Most simulation-based design strategies solve the intractable expected utility integral at a specific design point by using Monte Carlo simulations from the probability model. Optimizing the criterion over the design points is carried out in a separate step. Müller (1999) introduces an MCMC algorithm which simultaneously addresses the simulation as well as the optimization problem. In principle, the optimal design can be found by detecting the utility mode of the sampled design points. Several improvements have been suggested to facilitate this task for multidimensional design problems (see e.g. Amzal et al. 2006).

We aim to extend this simulation-based design methodology to design problems where the likelihood of the probability model is of an unknown analytical form but it is possible to simulate from the probability model. We further assume that prior observations are available. In such a setting it seems natural to employ approximate Bayesian computation (ABC) techniques in order to be able to simulate from the conditional probability model. We provide a thorough review of adjacent literature and we investigate the benefits and the limitations of our design methodology for a particular paradigmatic example.

Keywords. Simulation based optimal design, approximate Bayesian computation, Markov chain Monte Carlo.

1. Introduction

In the past decades simulation techniques, particularly the use of Markov chain Monte Carlo methods, have revolutionized statistical inference (cf. Robert and Casella 2004). There has, however, been little impact of this revolution on the experimental design literature other than the pioneering work initiated by Peter Müller (cf. Müller 1999 and Müller et al. 2004) and his followers. The main

reason may be that Müller's method has been limited to a specific choice of design criteria and has effectively been applicable only for finding rather small size designs. Another drawback has been the necessity to be able to explicitly specify the likelihood function for the problem, which can be an obstacle for complex settings faced in real applications.

With the advent of so-called likelihood-free (or approximate Bayesian computation - ABC) methods, the latter issue can be overcome, and we therefore propose to employ those techniques also for finding optimal experimental designs. There are essentially two ways of accomplishing this: the first is to marry ABC with Müller's essentially MCMC-based methods, the second is a more integrated albeit more complex implementation avoiding the MCMC step. While the latter has been put forward in Hainy et al. (2013), the former was alluded to in Hainy et al. (2012) and forms the essence of the present report.

Here we first in Section 2 give a thorough review of the essentials of simulation-based optimal design as well as the various improvements and modifications lately suggested. This ends with a particular emphasis on sequential Monte Carlo methods, which will eventually lend themselves naturally for solving adaptive design problems. Section 3 then presents modifications for including prior observations for the static and sequential design setup. Our main contribution is in Section 4, where we first review ABC methods that are relevant for our purposes and then suggest two MCMC-based optimal design procedures building upon them. Section 5 eventually discusses the type of design criteria / utility functions we can encompass and related implementation issues. Finally in Section 6 we illustrate our methods on a toy example which can be easily related to and understood from classic optimal design theory.

Note that shortly before finalizing this report we have learned of the unpublished paper by Drovandi and Pettitt (2012), wherein similar ideas have been developed independently. However, while the basic concept of fusing Müller's MCMC-based methods with ABC is essentially the same, our approach differs in various ways, particularly on how the posterior for the utility function is generated. Furthermore, we additionally suggest ways of how the methodology can be turned sequential so as to be made useful for adaptive design situations. In our future work we expect to provide comparisons of these differing variants of likelihood-free simulation-based optimal designs.

2. Simulation-based optimal design

2.1. Expected utility maximization

First we consider the case where no prior observations are available. The (future) data $\mathbf{z} \in \mathcal{Y}$ are sampled according to a known probability model which can be described by the likelihood function $p(\mathbf{z}|\boldsymbol{\theta}, \mathbf{d})$. The likelihood function depends on the parameters $\boldsymbol{\theta} \in \Theta$ and the chosen design $\mathbf{d} \in \Xi$. We assume that the parameters follow a prior distribution $p(\boldsymbol{\theta})$ which does not depend on the design \mathbf{d} . The posterior distribution of the parameters is denoted by $p(\boldsymbol{\theta}|\mathbf{z}, \mathbf{d}) \propto p(\mathbf{z}|\boldsymbol{\theta}, \mathbf{d})p(\boldsymbol{\theta})$, and $p(\mathbf{z}|\mathbf{d}) = \int_{\boldsymbol{\theta} \in \Theta} p(\mathbf{z}|\boldsymbol{\theta}, \mathbf{d})p(\boldsymbol{\theta})d\boldsymbol{\theta}$ is the prior predictive distribution.

The general aim of simulation-based optimal design is to find the optimal configuration $\mathbf{d}^* = \arg \sup_{\mathbf{d}} U(\mathbf{d})$ for the expected utility integral

$$\begin{aligned} U(\mathbf{d}) &= \int_{\mathbf{z} \in \mathcal{Y}} \left(\int_{\boldsymbol{\theta} \in \Theta} u(\mathbf{z}, \mathbf{d}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathbf{z}, \mathbf{d}) d\boldsymbol{\theta} \right) p(\mathbf{z}|\mathbf{d}) d\mathbf{z} \\ &= \int_{\mathbf{z} \in \mathcal{Y}} \int_{\boldsymbol{\theta} \in \Theta} u(\mathbf{z}, \mathbf{d}, \boldsymbol{\theta}) p(\mathbf{z}, \boldsymbol{\theta}|\mathbf{d}) d\boldsymbol{\theta} d\mathbf{z} \\ &= \int_{\mathbf{z} \in \mathcal{Y}} \int_{\boldsymbol{\theta} \in \Theta} u(\mathbf{z}, \mathbf{d}, \boldsymbol{\theta}) p(\mathbf{z}|\boldsymbol{\theta}, \mathbf{d}) p(\boldsymbol{\theta}) d\boldsymbol{\theta} d\mathbf{z}. \end{aligned} \quad (2.1)$$

The utility function $u(\cdot)$ may depend on the data \mathbf{z} , the design \mathbf{d} , and the parameters $\boldsymbol{\theta}$. This is the setting considered e.g. by Müller (1999) and Müller et al. (2004).

In many cases the integrals are analytically intractable and numerical solution techniques fail due to the high dimension of the variables. However, if it is possible to obtain a sample $\{\mathbf{z}^{(t)}, \boldsymbol{\theta}^{(t)}, t = 1, \dots, T\}$ from $p(\mathbf{z}, \boldsymbol{\theta}|\mathbf{d})$, for example by sampling $\boldsymbol{\theta}$ from its prior distribution and $\mathbf{z}|\boldsymbol{\theta}, \mathbf{d}$ from the probability model, a straightforward way to approximate $U(\mathbf{d})$ is to use Monte Carlo integration:

$$U(\mathbf{d}) \approx \hat{U}(\mathbf{d}) = \frac{1}{T} \sum_{t=1}^T u(\mathbf{z}^{(t)}, \mathbf{d}, \boldsymbol{\theta}^{(t)}).$$

In a second step, stochastic optimization algorithms may then be performed on $\hat{U}(\mathbf{d})$ to find the optimum. Possible algorithms include steepest-ascent-type algorithms, global optimization methods such as simulated annealing, genetic algorithms of various kinds, global random search or exchange algorithms, see e.g. Zhigljavsky and Žilinskas (2008). Depending on the computational effort to obtain the draws $\{\mathbf{z}^{(t)}, \boldsymbol{\theta}^{(t)}, t = 1, \dots, T\}$ and to evaluate the function $u(\mathbf{z}^{(t)}, \mathbf{d}, \boldsymbol{\theta}^{(t)})$, this can be time-consuming because every optimization step requires several evaluations of $\hat{U}(\mathbf{d})$. Specific

stochastic optimization algorithms have been developed that minimize the computational costs by reducing the number of evaluations, for example the *simultaneous perturbation stochastic approximation* (SPSA) algorithm proposed by Spall (1998a,b). For a discussion of these methods in the context of simulation-based design see e.g. Huan and Marzouk (2011). However, even these improved algorithms may still be slow, depending on the situation, and they may exhibit slow convergence for certain shapes of the utility function.

2.2. MCMC algorithm

Müller (1999) proposes an alternative way to tackle this optimization problem that combines the simulation as well as the optimization steps. To implement a stochastic search the integrand in (2.1) is regarded as proportional to a joint probability distribution of the variables \mathbf{z} , \mathbf{d} , and $\boldsymbol{\theta}$:

$$h(\mathbf{z}, \mathbf{d}, \boldsymbol{\theta}) \propto u(\mathbf{z}, \mathbf{d}, \boldsymbol{\theta})p(\mathbf{z}|\boldsymbol{\theta}, \mathbf{d})p(\boldsymbol{\theta})\mu(\mathbf{d}),$$

where $\mu(\mathbf{d})$ is some (usually uniform) measure on the design region. If $u(\cdot)$ is positive and bounded, then $h(\cdot)$ is a proper pdf and it is straightforward to use Markov chain Monte Carlo (MCMC) methods such as Metropolis Hastings (MH) to generate draws from this distribution.

Integrating over $\boldsymbol{\theta}$ and \mathbf{z} , we get

$$U(\mathbf{d}) \propto \int_{\mathbf{z} \in \mathcal{Y}} \int_{\boldsymbol{\theta} \in \Theta} h(\mathbf{z}, \mathbf{d}, \boldsymbol{\theta}) d\boldsymbol{\theta} d\mathbf{z},$$

so the marginal distribution of \mathbf{d} is proportional to the expected utility function. Therefore, a strategy to find the optimum design is to sample from $h(\mathbf{z}, \mathbf{d}, \boldsymbol{\theta})$, retain the draws of \mathbf{d} , and then search for the mode of the marginal distribution of \mathbf{d} by inspecting the draws.

There are many ways to generate draws from $h(\mathbf{z}, \mathbf{d}, \boldsymbol{\theta})$, for example via Gibbs sampling or by using a hybrid Gibbs/MH sampler. For a review of MCMC sampling schemes see Tierney (1994).

Müller (1999) uses a simple MH scheme with the following proposal distribution for a new draw $\boldsymbol{\vartheta}' = (\mathbf{z}', \mathbf{d}', \boldsymbol{\theta}')$ from $\boldsymbol{\vartheta}^{(t-1)} = (\mathbf{z}^{(t-1)}, \mathbf{d}^{(t-1)}, \boldsymbol{\theta}^{(t-1)})$

$$q(\boldsymbol{\vartheta}'|\boldsymbol{\vartheta}^{(t-1)}) = p(\mathbf{z}'|\boldsymbol{\theta}', \mathbf{d}')p(\boldsymbol{\theta}')g(\mathbf{d}'|\mathbf{d}^{(t-1)}),$$

where $g(\mathbf{d}'|\mathbf{d}^{(t-1)})$ is a random walk proposal for \mathbf{d} , and the joint distribution of \mathbf{z} and $\boldsymbol{\theta}$ is used as independence proposal for the data and the parameters. Usually it is not hard to sample from this

proposal. Specifying the proposal distribution in this way leads to the acceptance probability

$$\alpha = \min \left(1, \frac{u(\boldsymbol{\theta}')}{u(\boldsymbol{\theta}^{(t-1)})} \frac{g(\mathbf{d}^{(t-1)}|\mathbf{d}')}{g(\mathbf{d}'|\mathbf{d}^{(t-1)})} \right).$$

Note that the terms $p(\mathbf{z}'|\boldsymbol{\theta}', \mathbf{d}')$ and $p(\mathbf{z}^{(t-1)}|\boldsymbol{\theta}^{(t-1)}, \mathbf{d}^{(t-1)})p(\boldsymbol{\theta}^{(t-1)})$ do not appear in this formula.

However, if $p(\mathbf{z}, \boldsymbol{\theta}|\mathbf{d})$ is rather flat or $u(\cdot)$ is peaked in regions where $p(\mathbf{z}, \boldsymbol{\theta}|\mathbf{d}) = p(\mathbf{z}|\boldsymbol{\theta}, \mathbf{d})p(\boldsymbol{\theta})$ is relatively low, then this proposal distribution will lead to a poorly performing sampler. Thus, in some cases it might be preferable to use customized proposal densities, which better reflect the shape of the true pdf $h(\mathbf{z}, \mathbf{d}, \boldsymbol{\theta})$. It is also possible to set up a (hybrid) Gibbs scheme with conditional draws of \mathbf{z} , \mathbf{d} , and $\boldsymbol{\theta}$. Jacquier et al. (2011) implement both strategies.

There are several advantages of the MCMC scheme over other standard stochastic optimization approaches: First, as Müller (1999) states, it provides a unified framework for simulation and optimization. Second, only one draw of \mathbf{z} and $\boldsymbol{\theta}$ is sampled at each new design point \mathbf{d} , rather than T , thereby relieving the computational burden. Further Solonen et al. (2012) note that the output of the MCMC sampler provides a “map” of useful designs and not just one single maximum design.

2.3. Simulated annealing

A drawback of the basic MCMC algorithm is that it may be very hard to find the mode of the marginal distribution of \mathbf{d} from the MCMC draws, especially in higher-dimensional design spaces and if the expected utility surface is rather flat.

To facilitate the detection of the mode Müller (1999) suggests a method which is similar to simulated annealing (see e.g. van Laarhoven and Aarts 1992, Kirkpatrick et al. 1983) in that the target distribution is the expected utility function taken to a high power. Then all draws of the MCMC sampler will cluster tightly around the mode, and the mean of the draws will be a good approximation to the mode. In order to achieve that, the target distribution is modified in the following way (augmented model):

$$h_J(\mathbf{z}_{1:J}, \mathbf{d}, \boldsymbol{\theta}_{1:J}) \propto \prod_{j=1}^J \phi(\mathbf{z}_j, \mathbf{d}, \boldsymbol{\theta}_j) \mu(\mathbf{d}), \quad (2.2)$$

where

$$\phi(\mathbf{z}_j, \mathbf{d}, \boldsymbol{\theta}_j) = u(\mathbf{z}_j, \mathbf{d}, \boldsymbol{\theta}_j) p(\mathbf{z}_j|\boldsymbol{\theta}_j, \mathbf{d}) p(\boldsymbol{\theta}_j). \quad (2.3)$$

Integrating over $\mathbf{z}_1, \dots, \mathbf{z}_J$ and $\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_J$ leads to a marginal distribution proportional to $U^J(\mathbf{d})$. Therefore, $1/J$ might be regarded as annealing temperature. As $J \rightarrow \infty$ $U^J(\mathbf{d})$ collapses to a point mass at $\mathbf{d} = \mathbf{d}^*$. Thus, for every draw of \mathbf{d} one has to take J independent draws of \mathbf{z} and $\boldsymbol{\theta}$ and multiply the utilities.

If there are several modes, there is a danger of getting trapped in one local mode if J is too large. Müller et al. (2004) propose an extension of the basic algorithm where J is gradually increased over time as the algorithm progresses, thereby forming an inhomogeneous Markov chain of the variables. In the beginning the MCMC sampler explores the whole design space so that no local mode is missed. Subsequently the MCMC draws concentrate more and more around one of the highest modes. The acceptance probability formula at time t is

$$\alpha = \min \left(1, \frac{\prod_{j=1}^{J(t)} u(\mathbf{z}'_j, \mathbf{d}'_j, \boldsymbol{\theta}'_j)}{\prod_{j=1}^{J(t)} u(\mathbf{z}_j^{(t-1)}, \mathbf{d}_j^{(t-1)}, \boldsymbol{\theta}_j^{(t-1)})} \frac{g(\mathbf{d}^{(t-1)}|\mathbf{d}')}{g(\mathbf{d}'|\mathbf{d}^{(t-1)})} \right).$$

Müller et al. (2004) show that the inhomogeneous Markov chain produced by this sampler is still strongly ergodic if the “cooling” schedule in J is logarithmic. If $J(t) > J(t-1)$, then values $\{(\mathbf{z}_j^{(t-1)}, \boldsymbol{\theta}_j^{(t-1)}), j = J(t-1) + 1, \dots, J(t)\}$ are generated by resampling $\{(\mathbf{z}_j^{(t-1)}, \boldsymbol{\theta}_j^{(t-1)}), j = 1, \dots, J(t-1)\}$.

2.4. Particle methods

Particle methods are a flexible alternative to MCMC algorithms for sampling from target distributions that change over time. This is also the case for our distribution of interest $h_J(\mathbf{z}_{1:J}, \mathbf{d}, \boldsymbol{\theta}_{1:J})$ if J is increasing over time. Amzal et al. (2006) present two interacting particle systems algorithms for the goal of sampling from $h_J(\cdot)$: a classical *sampling importance resampling* algorithm and a computationally more efficient *resampling-Markov* algorithm with weight updates. A different approach is taken by Kück et al. who employ *sequential Monte Carlo (SMC)* ideas going back to Del Moral et al. (2006).

Sampling importance resampling algorithm

Particle systems usually consist of N particles which are propagated over time. The general sampling importance resampling algorithm is composed of three steps at each time t ($t = 1, \dots, T$).

The **first step** is a common importance sampling step to obtain an approximate weighted sample from the target distribution. For each particle $n = 1, \dots, N$, at time t a new tuple $\boldsymbol{\vartheta}_{n,J(t)}^{(t)} = (\mathbf{z}_{n,1:J(t)}^{(t)}, \mathbf{d}_n^{(t)}, \boldsymbol{\theta}_{n,1:J(t)}^{(t)})$ is generated by sampling from the importance function

$$q_{IS}(\boldsymbol{\vartheta}_{n,J(t)}^{(t)} | \mathbf{d}_n^{(t-1)}) = g_{IS}(\mathbf{d}_n^{(t)} | \mathbf{d}_n^{(t-1)}) \prod_{j=1}^{J(t)} p(\mathbf{z}_{n,j}^{(t)} | \boldsymbol{\theta}_{n,j}^{(t)}, \mathbf{d}_n^{(t)}) p(\boldsymbol{\theta}_{n,j}^{(t)}). \quad (2.4)$$

Therefore, the importance weights are computed as

$$w_n^{(t)} = \frac{h_{J(t)}(\boldsymbol{\vartheta}_{n,J(t)}^{(t)})}{q_{IS}(\boldsymbol{\vartheta}_{n,J(t)}^{(t)} | \mathbf{d}_n^{(t-1)})} \propto \frac{\prod_{j=1}^{J(t)} u(\mathbf{z}_{n,j}^{(t)}, \mathbf{d}_n^{(t)}, \boldsymbol{\theta}_{n,j}^{(t)})}{g_{IS}(\mathbf{d}_n^{(t)} | \mathbf{d}_n^{(t-1)})}, \quad n = 1, \dots, N. \quad (2.5)$$

After normalizing the weights, i.e. $W_n^{(t)} = w_n^{(t)} / \sum_{i=1}^N w_i^{(t)}$, a weighted sample is obtained which may serve as a discrete approximation to the target distribution:

$$h_{J(t)}(d\boldsymbol{\vartheta}) \approx \sum_{n=1}^N W_n^{(t)} \delta_{\boldsymbol{\vartheta}_{n,J(t)}^{(t)}}(d\boldsymbol{\vartheta}).$$

Here, $\delta_X(dx)$ denotes the delta-Dirac measure that concentrates all the mass at point X . Moreover, one can use the approximation

$$\mathbb{E}_{h_{J(t)}}(\varphi) = \int \varphi(\boldsymbol{\vartheta}) h_{J(t)}(\boldsymbol{\vartheta}) d\boldsymbol{\vartheta} \approx \sum_{n=1}^N W_n^{(t)} \varphi(\boldsymbol{\vartheta}_{n,J(t)}^{(t)}) = \hat{\mathbb{E}}_{h_{J(t)}}(\varphi)$$

for any measurable and bounded function φ . The variance of this estimator is approximately

$$\mathbb{V}_{q_{IS}}(\hat{\mathbb{E}}_{h_{J(t)}}(\varphi)) \approx \mathbb{V}_{h_{J(t)}}(\varphi) \left(1 + \mathbb{V}_{q_{IS}}(w_n^{(t)}(\boldsymbol{\vartheta}_{n,J(t)}^{(t)})) \right) / N,$$

see Liu (2001). If an i.i.d. sample $\{\tilde{\boldsymbol{\vartheta}}_{1,J(t)}, \dots, \tilde{\boldsymbol{\vartheta}}_{N,J(t)}\}$ from the target distribution $h_{J(t)}$ were available, the variance of the estimator would be $\mathbb{V}_{h_{J(t)}}(\hat{\mathbb{E}}_{h_{J(t)}}(\varphi)) = \mathbb{V}_{h_{J(t)}}(\varphi)/N$. Hence, the variance of the estimator is directly proportional to the variance of the weights. The quantity

$$\text{ESS} = N / \left(1 + \mathbb{V}_{q_{IS}}(w_n^{(t)}(\boldsymbol{\vartheta}_{n,J(t)}^{(t)})) \right)$$

is therefore termed the *effective sample size*. Roughly speaking, it says that N weighted samples drawn from the importance distribution are worth ESS i.i.d. samples drawn directly from the target distribution. The effective sample size can be conveniently estimated by

$$\widehat{\text{ESS}} = 1 / \sum_{n=1}^N \left(W_n^{(t)} \right)^2.$$

Amzal et al. (2006) recommend to use an importance distribution that closely resembles the target distribution, so the variance of $g(\mathbf{d}_n^{(t)} | \mathbf{d}_n^{(t-1)})$ decreases to 0 with rate $1/J$.

The **second step** is a selection or resampling step. The particles $\{\boldsymbol{\vartheta}_{n,J(t)}^{(t)}, n = 1, \dots, N\}$, are resampled according to a multinomial distribution with weights $W_n^{(t)}$. The new sample $\{\bar{\boldsymbol{\vartheta}}_{n,J(t)}^{(t)} = (\bar{\mathbf{z}}_{n,1:J(t)}^{(t)}, \bar{\mathbf{d}}_n^{(t)}, \bar{\boldsymbol{\theta}}_{n,1:J(t)}^{(t)}), n = 1, \dots, N\}$ is also an approximate sample from the target distribution with equal weights $W_n^{(t)} = 1/N$. On the one hand, the resampling step reduces the ESS, on the other hand, each resampling step is an additional approximation that might deteriorate the approximation quality. Some particles with low weights drop out of the sample after the resampling step, leading to a concentration of particles around few design points.

As a **third step**, Amzal et al. (2006) employ a Metropolis-Hastings step to enrich the sample. It would be sufficient to implement only the first (and second) step for the sampler to be valid. However, as mentioned above, this strategy often leads to a strong concentration of design points because many particles are discarded over time. In order to allow the sampler to explore new regions, an MH step is added. Amzal et al. (2006) suggest to use a proposal distribution for the design points $g_{MH}(\mathbf{d}'_n | \bar{\mathbf{d}}_n^{(t)})$ that has a relatively high variance to make it possible to break out and detect other modes. Using the joint distribution of \mathbf{z} and $\boldsymbol{\theta}$ again as independence proposal for these variables (i.e. $\mathbf{z}'_{n,1:J(t)}, \boldsymbol{\theta}'_{n,1:J(t)} | \mathbf{d}'_n \sim \prod_{j=1}^{J(t)} p(\mathbf{z}'_{n,j} | \boldsymbol{\theta}'_{n,j}, \mathbf{d}'_n) p(\boldsymbol{\theta}'_{n,j})$), the acceptance probability for the MH step at particle n ($n = 1, \dots, N$) is given as

$$\alpha = \min \left(1, \frac{\prod_{j=1}^{J(t)} u(\mathbf{z}'_{n,j}, \mathbf{d}'_n, \boldsymbol{\theta}'_{n,j})}{\prod_{j=1}^{J(t)} u(\bar{\mathbf{z}}_{n,j}^{(t)}, \bar{\mathbf{d}}_n^{(t)}, \bar{\boldsymbol{\theta}}_{n,j}^{(t)})} \frac{g(\bar{\mathbf{d}}_n^{(t)} | \mathbf{d}'_n)}{g(\mathbf{d}'_n | \bar{\mathbf{d}}_n^{(t)})} \right).$$

Resampling-Markov algorithm

Amzal et al. (2006) also propose a computationally less demanding alternative to the sampling importance resampling algorithm. After initialization the importance sampling step is omitted in all subsequent iterations of the sampler. Assume that at time $t - 1$ a sample approximately drawn from $h_{J(t-1)}$ is available. If $J(t) > J(t - 1)$ additional values $\{(\mathbf{z}_{n,j}^{(t-1)}, \boldsymbol{\theta}_{n,j}^{(t-1)}), j = J(t - 1) + 1, \dots, J(t)\}$ for each particle are drawn from $\mathbf{z}_{n,J(t-1)+1:J(t)}^{(t-1)}, \boldsymbol{\theta}_{n,J(t-1)+1:J(t)}^{(t-1)} | \mathbf{d}_n^{(t-1)} \sim$

$\prod_{j=J(t-1)+1}^{J(t)} p(\mathbf{z}_{n,j}^{(t-1)} | \boldsymbol{\theta}_{n,j}^{(t-1)}, \mathbf{d}_n^{(t-1)}) p(\boldsymbol{\theta}_{n,j}^{(t-1)})$ and the weights are computed

$$w_n^{(t)} \propto w_n^{(t-1)} \prod_{j=J(t-1)+1}^{J(t)} u(\mathbf{z}_{n,j}^{(t-1)}, \mathbf{d}_n^{(t-1)}, \boldsymbol{\theta}_{n,j}^{(t-1)}). \quad (2.6)$$

The resampling and the Metropolis-Hastings steps are performed as before. After each resampling step all the weights are $1/N$, $w_n^{(t-1)} = 1/N$ in the formula above. The weight at iteration t is just proportional to the product of the utilities of the newly sampled values. This algorithm avoids the complete (importance) sampling of all variables at every iteration. However, it may be harder to explore complex utility surfaces with this method.

Sequential Monte Carlo algorithm

Kück et al. apply the sequential Monte Carlo sampler framework of Del Moral et al. (2006). An auxiliary augmented target distribution

$$\tilde{\pi}_t(\mathbf{x}^{(1:t)}) = \pi_t(\mathbf{x}^{(t)}) \prod_{k=t-1}^1 L_k(\mathbf{x}^{(k+1)}, \mathbf{x}^{(k)})$$

is constructed which assumes the distribution $\pi_t(\mathbf{x}^{(t)})$ as its marginal. Importance sampling is performed on the complete sample $\mathbf{x}^{(1:t)}$, even though the actual interest is only in the marginal distribution of $\mathbf{x}^{(t)}$. The kernel function $L_k(\cdot, \cdot)$ is called *backward kernel*. The choice of $L_k(\cdot, \cdot)$ is arbitrary but may affect sampling efficiency. For a discussion of expedient backward kernels see Del Moral et al. (2006). The importance distribution of the complete sample $\mathbf{x}^{(1:t)}$ is designed in a similar manner as the augmented target distribution:

$$q_t(\mathbf{x}^{(1:t)}) = q_1(\mathbf{x}^{(1)}) \prod_{k=2}^t K_k(\mathbf{x}^{(k-1)}, \mathbf{x}^{(k)}).$$

For some particle $n \in \{1, \dots, N\}$, $\mathbf{x}_n^{(t)}$ is sampled from the *forward kernel* $K_t(\mathbf{x}_n^{(t-1)}, \mathbf{x}_n^{(t)})$ and therefore only depends on $\mathbf{x}_n^{(t-1)}$ and not on the whole particle history. It is possible to calculate the weights at time t recursively:

$$w_n^{(t)}(\mathbf{x}_n^{(1:t)}) = w_n^{(t-1)}(\mathbf{x}_n^{(1:t-1)}) \frac{\pi_t(\mathbf{x}_n^{(t)})}{\pi_{t-1}(\mathbf{x}_n^{(t-1)})} \frac{L_{t-1}(\mathbf{x}_n^{(t)}, \mathbf{x}_n^{(t-1)})}{K_t(\mathbf{x}_n^{(t-1)}, \mathbf{x}_n^{(t)})}.$$

If the target densities are only known up to a normalizing constant, the weights have to be normalized at the end to obtain an approximation to $\tilde{\pi}(\mathbf{x}^{(1:t)})$. Since $\pi(\mathbf{x}^{(t)})$ is just the marginal distribution of $\tilde{\pi}(\mathbf{x}^{(1:t)})$, these weights can also be used to approximate the distribution of $\mathbf{x}^{(t)}$.

SMC algorithms offer a flexible way to slowly “approach” a complex target distribution via a sequence of intermediate target distributions. The weights have to be adjusted properly at each time step. The idea is to start with a target distribution π_1 which is easy to approximate by an importance distribution. If the target distribution changes only gradually from time $t - 1$ to time t , then slight perturbations of the sample induced by the forward kernel should also be good approximations to the new target distribution. The error of sampling from the wrong distribution is captured by the weights.

The variance of the weights usually increases over time so that the ESS goes down. A common strategy to remedy this problem is to perform a resampling step as soon as the ESS falls below some threshold value (often $N/2$).

Kück et al. apply the SMC methodology in a particular situation with Shannon information as utility function. In this application the parameters $\boldsymbol{\theta}$ can be integrated out of the expected utility integral (2.1) which simplifies the problem to some extent. In a more general setting where this integration is not feasible the following sequence of target distributions is sampled:

$$h_{J(t), \nu_t}(\mathbf{z}_{1:J(t)}, \mathbf{d}, \boldsymbol{\theta}_{1:J(t)}) \propto \left(\prod_{j=1}^{J(t)-1} \phi(\mathbf{z}_j, \mathbf{d}, \boldsymbol{\theta}_j) \right) \phi(\mathbf{z}_{J(t)}, \mathbf{d}, \boldsymbol{\theta}_{J(t)})^{\nu_t} \mu(\mathbf{d}).$$

The monotonically increasing inverse annealing temperature $J(t)$ is integer-valued, and $J(t) \leq J(t-1) + 1$, so J increases by at most 1 at each time step. In order to allow for more finely graduated non-integer jumps of the inverse annealing temperature, Kück et al. introduce the variable $\nu_t \in [0, 1]$. The original target density (2.2) is obtained if $\nu_t = 1$. Whenever ν_t reaches 1, the dimension of the target distribution increases, $J(t) = J(t-1) + 1$, and ν_t is set back to 0 again. In all other cases $J(t) = J(t-1)$.

The forward kernel suggested by Kück et al. is

$$K_t \left(\boldsymbol{\vartheta}_{n, J(t-1)}^{(t-1)}, \boldsymbol{\vartheta}_{n, J(t)}^{(t)} \right) = \mathcal{K}_{J(t)-1, 1} \left(\boldsymbol{\vartheta}_{n, J(t)-1}^{(t-1)}, \boldsymbol{\vartheta}_{n, J(t)-1}^{(t)} \right) q_{\nu_t} \left(\mathbf{z}_{n, J(t)}^{(t)}, \boldsymbol{\theta}_{n, J(t)}^{(t)} | \mathbf{d}_n^{(t)} \right),$$

where $\boldsymbol{\vartheta}_{n, j}^{(t)} = (\mathbf{z}_{n, 1:j}^{(t)}, \mathbf{d}_n^{(t)}, \boldsymbol{\theta}_{n, 1:j}^{(t)})$, $\mathcal{K}_{J(t)-1, 1}$ is the Metropolis-Hastings kernel with invariant distribution $h_{J(t)-1, 1}$, and $q_{\nu_t} \left(\mathbf{z}_{n, J(t)}^{(t)}, \boldsymbol{\theta}_{n, J(t)}^{(t)} | \mathbf{d}_n^{(t)} \right)$ is an independence proposal for $\mathbf{z}_{n, J(t)}^{(t)}$ and $\boldsymbol{\theta}_{n, J(t)}^{(t)}$.

Selecting the backward kernel as

$$L_{t-1} \left(\boldsymbol{\vartheta}_{n,J(t)}^{(t)}, \boldsymbol{\vartheta}_{n,J(t-1)}^{(t-1)} \right) = \mathcal{K}_{n,J(t)-1,1} \left(\boldsymbol{\vartheta}_{n,J(t)-1}^{(t-1)}, \boldsymbol{\vartheta}_{n,J(t)-1}^{(t)} \right) \cdot \frac{h_{J(t-1),\nu_{t-1}} \left(\boldsymbol{\vartheta}_{n,J(t-1)}^{(t-1)} \right)}{h_{J(t)-1,1} \left(\boldsymbol{\vartheta}_{n,J(t)-1}^{(t)} \right)}$$

leads to the incremental weights

$$w_n^{(t)} \propto w_n^{(t-1)} \frac{\phi \left(\mathbf{z}_{n,J(t)}^{(t)}, \mathbf{d}^{(t)}, \boldsymbol{\theta}_{n,J(t)}^{(t)} \right)^{\nu_t}}{q_{\nu_t} \left(\mathbf{z}_{n,J(t)}^{(t)}, \mathbf{d}^{(t)}, \boldsymbol{\theta}_{n,J(t)}^{(t)} \right)}. \quad (2.7)$$

A slowly evolving target distribution usually implies a better fitting importance distribution, so the ESS deteriorates less rapidly and there have to be fewer resampling steps.

Kück et al. use $q_{\nu_t} \left(\mathbf{z}_{n,J(t)}^{(t)}, \boldsymbol{\theta}_{n,J(t)}^{(t)} | \mathbf{d}_n^{(t)} \right) = p \left(\mathbf{z}_{n,J(t)}^{(t)}, \boldsymbol{\theta}_{n,J(t)}^{(t)} | \mathbf{d}_n^{(t)} \right)$. They mention that this choice of proposal distribution seems to work well when $\nu_t = 1$. If $\nu_t < 1$, the importance weights tend to exhibit a high variance. Furthermore, $\mathbf{z}_{n,J(t)}^{(t)}$ and $\boldsymbol{\theta}_{n,J(t)}^{(t)}$ are sampled independently from the previous sample at all time steps. Ideally, they should only be sampled independently when $J(t)$ makes a jump, i.e. when $J(t) = J(t-1) + 1$. Kück et al. therefore propose another sampling scheme in which they use different forward and backward kernels depending on whether $J(t) = J(t-1) + 1$ or $J(t) = J(t-1)$.

3. Simulation-based optimal design with prior observations

The algorithms presented in Section 2 have to be modified slightly if prior observations $\mathbf{y}_{1:s} = \{\mathbf{y}_i, i = 1, \dots, s\}$ measured at the design points $\boldsymbol{\xi}_{1:s} = \{\boldsymbol{\xi}_i, i = 1, \dots, s\}$ are available for s measurements collected in the past (cf. Müller et al. 2004). If we assume that

$$p(\boldsymbol{\theta} | \{\mathbf{z}, \mathbf{y}_{1:s}\}, \{\mathbf{d}, \boldsymbol{\xi}_{1:s}\}) \propto p(\mathbf{z} | \boldsymbol{\theta}, \mathbf{d}) \left(\prod_{i=1}^s p(\mathbf{y}_i | \boldsymbol{\theta}, \boldsymbol{\xi}_i) \right) p(\boldsymbol{\theta}) = p(\mathbf{z} | \boldsymbol{\theta}, \mathbf{d}) p(\mathbf{y}_{1:s} | \boldsymbol{\theta}, \boldsymbol{\xi}_{1:s}) p(\boldsymbol{\theta}),$$

then the expected utility integral (2.1) changes to

$$\begin{aligned} U(\mathbf{d}) &= \int_{\mathbf{z} \in \mathcal{Y}} \left(\int_{\boldsymbol{\theta} \in \Theta} u(\{\mathbf{z}, \mathbf{y}_{1:s}\}, \{\mathbf{d}, \boldsymbol{\xi}_{1:s}\}, \boldsymbol{\theta}) p(\boldsymbol{\theta} | \{\mathbf{z}, \mathbf{y}_{1:s}\}, \{\mathbf{d}, \boldsymbol{\xi}_{1:s}\}) d\boldsymbol{\theta} \right) p(\mathbf{z} | \mathbf{y}_{1:s}, \{\mathbf{d}, \boldsymbol{\xi}_{1:s}\}) d\mathbf{z} \\ &= \int_{\mathbf{z} \in \mathcal{Y}} \left(\int_{\boldsymbol{\theta} \in \Theta} u(\{\mathbf{z}, \mathbf{y}_{1:s}\}, \{\mathbf{d}, \boldsymbol{\xi}_{1:s}\}, \boldsymbol{\theta}) \frac{p(\mathbf{z} | \boldsymbol{\theta}, \mathbf{d}) (\prod_{i=1}^s p(\mathbf{y}_i | \boldsymbol{\theta}, \boldsymbol{\xi}_i)) p(\boldsymbol{\theta})}{p(\{\mathbf{z}, \mathbf{y}_{1:s}\} | \{\mathbf{d}, \boldsymbol{\xi}_{1:s}\})} d\boldsymbol{\theta} \right) \times \\ &\quad p(\mathbf{z} | \mathbf{y}_{1:s}, \{\mathbf{d}, \boldsymbol{\xi}_{1:s}\}) d\mathbf{z} \\ &= \int_{\mathbf{z} \in \mathcal{Y}} \left(\int_{\boldsymbol{\theta} \in \Theta} u(\{\mathbf{z}, \mathbf{y}_{1:s}\}, \{\mathbf{d}, \boldsymbol{\xi}_{1:s}\}, \boldsymbol{\theta}) \frac{p(\mathbf{z} | \boldsymbol{\theta}, \mathbf{d})}{p(\mathbf{z} | \mathbf{y}_{1:s}, \{\mathbf{d}, \boldsymbol{\xi}_{1:s}\})} \frac{(\prod_{i=1}^s p(\mathbf{y}_i | \boldsymbol{\theta}, \boldsymbol{\xi}_i)) p(\boldsymbol{\theta})}{p(\mathbf{y}_{1:s} | \boldsymbol{\xi}_{1:s})} d\boldsymbol{\theta} \right) \times \end{aligned}$$

$$\begin{aligned}
& p(\mathbf{z}|\mathbf{y}_{1:s}, \{\mathbf{d}, \boldsymbol{\xi}_{1:s}\})d\mathbf{z} \\
&= \int_{\mathbf{z} \in \mathcal{Y}} \int_{\boldsymbol{\theta} \in \Theta} u(\{\mathbf{z}, \mathbf{y}_{1:s}\}, \{\mathbf{d}, \boldsymbol{\xi}_{1:s}\}, \boldsymbol{\theta}) p(\mathbf{z}|\boldsymbol{\theta}, \mathbf{d}) p(\boldsymbol{\theta}|\mathbf{y}_{1:s}, \boldsymbol{\xi}_{1:s}) d\boldsymbol{\theta} d\mathbf{z}.
\end{aligned}$$

If it is possible to sample from the posterior distribution $p(\boldsymbol{\theta}|\mathbf{y}_{1:s}, \boldsymbol{\xi}_{1:s})$, then all the algorithms introduced in the previous section can still be used. The only difference is that instead of $p(\boldsymbol{\theta})$ one has to use $p(\boldsymbol{\theta}|\mathbf{y}_{1:s}, \boldsymbol{\xi}_{1:s})$ for sampling the parameters. However, MCMC and importance sampling algorithms are usually employed when a closed-form representation of the posterior distribution cannot be obtained. Hence, the algorithms presented in Section 2 have to be generalized. Instead of

$$q(\mathbf{z}'_{1:J}, \mathbf{d}', \boldsymbol{\theta}'_{1:J}|\mathbf{d}^{(t-1)}) = \prod_{j=1}^J p(\mathbf{z}'_j|\boldsymbol{\theta}'_j, \mathbf{d}') p(\boldsymbol{\theta}'_j|\mathbf{y}_{1:s}, \boldsymbol{\xi}_{1:s}) g(\mathbf{d}'|\mathbf{d}^{(t-1)})$$

the more general proposal distribution

$$q_k(\mathbf{z}'_{1:J}, \mathbf{d}', \boldsymbol{\theta}'_{1:J}|\mathbf{d}^{(t-1)}) = \prod_{j=1}^J p(\mathbf{z}'_j|\boldsymbol{\theta}'_j, \mathbf{d}') k(\boldsymbol{\theta}'_j|\mathbf{y}_{1:s}, \boldsymbol{\xi}_{1:s}) g(\mathbf{d}'|\mathbf{d}^{(t-1)}).$$

is used. Thus $\boldsymbol{\theta}'_j$ is not sampled from the correct posterior, but from an approximate proposal distribution which should resemble the posterior distribution as closely as possible. Common choices for these proposals are normal or t random walk or independence proposals, where the scale is proportional to the inverse of the Hessian of the likelihood or the unnormalized posterior.

The target distribution changes from (2.2) to

$$\begin{aligned}
h_J(\mathbf{z}_{1:J}, \mathbf{d}, \boldsymbol{\theta}_{1:J}) &\propto \prod_{j=1}^J u(\{\mathbf{z}_j, \mathbf{y}_{1:s}\}, \{\mathbf{d}, \boldsymbol{\xi}_{1:s}\}, \boldsymbol{\theta}_j) p(\mathbf{z}_j|\boldsymbol{\theta}_j, \mathbf{d}) p(\boldsymbol{\theta}_j|\mathbf{y}_{1:s}, \boldsymbol{\xi}_{1:s}) \mu(\mathbf{d}) \propto \\
&\propto \prod_{j=1}^J u(\{\mathbf{z}_j, \mathbf{y}_{1:s}\}, \{\mathbf{d}, \boldsymbol{\xi}_{1:s}\}, \boldsymbol{\theta}_j) p(\mathbf{z}_j|\boldsymbol{\theta}_j, \mathbf{d}) p(\mathbf{y}_{1:s}|\boldsymbol{\theta}_j, \boldsymbol{\xi}_{1:s}) p(\boldsymbol{\theta}_j) \mu(\mathbf{d}). \quad (3.1)
\end{aligned}$$

In a standard MCMC annealing algorithm this would lead to the acceptance probability (cf. Müller et al. 2004)

$$\begin{aligned}
\alpha = \min \left(1, \frac{\prod_{j=1}^{J(t)} u(\{\mathbf{z}'_j, \mathbf{y}_{1:s}\}, \{\mathbf{d}', \boldsymbol{\xi}_{1:s}\}, \boldsymbol{\theta}'_j)}{\prod_{j=1}^{J(t)} u(\{\mathbf{z}_j^{(t-1)}, \mathbf{y}_{1:s}\}, \{\mathbf{d}^{(t-1)}, \boldsymbol{\xi}_{1:s}\}, \boldsymbol{\theta}_j^{(t-1)})} \frac{\prod_{j=1}^{J(t)} p(\mathbf{y}_{1:s}|\boldsymbol{\theta}'_j, \boldsymbol{\xi}_{1:s}) p(\boldsymbol{\theta}'_j)}{\prod_{j=1}^{J(t)} p(\mathbf{y}_{1:s}|\boldsymbol{\theta}_j^{(t-1)}, \boldsymbol{\xi}_{1:s}) p(\boldsymbol{\theta}_j^{(t-1)})} \times \right. \\
\left. \frac{\prod_{j=1}^{J(t)} k(\boldsymbol{\theta}_j^{(t-1)}|\mathbf{y}_{1:s}, \boldsymbol{\xi}_{1:s}) g(\mathbf{d}^{(t-1)}|\mathbf{d}')}{\prod_{j=1}^{J(t)} k(\boldsymbol{\theta}'_j|\mathbf{y}_{1:s}, \boldsymbol{\xi}_{1:s}) g(\mathbf{d}'|\mathbf{d}^{(t-1)})} \right). \quad (3.2)
\end{aligned}$$

Similar modifications are necessary for applying the sampling importance resampling algorithm.

The importance distribution (2.4) could be changed to

$$q_{IS,k}(\boldsymbol{\vartheta}_{n,J(t)}^{(t)}|\mathbf{d}_n^{(t-1)}) = g_{IS}(\mathbf{d}_n^{(t)}|\mathbf{d}_n^{(t-1)}) \prod_{j=1}^{J(t)} p(\mathbf{z}_{n,j}^{(t)}|\boldsymbol{\theta}_{n,j}^{(t)}, \mathbf{d}_n^{(t)}) k(\boldsymbol{\theta}_{n,j}^{(t)}|\mathbf{y}_{1:s}, \boldsymbol{\xi}_{1:s}),$$

implying a change of the weights from (2.5) to

$$w_n^{(t)} = \frac{h_{J(t)}(\boldsymbol{\vartheta}_{n,J(t)}^{(t)})}{q_{IS}(\boldsymbol{\vartheta}_{n,J(t)}^{(t)}|\mathbf{d}_n^{(t-1)})} \propto \frac{\prod_{j=1}^{J(t)} u(\{\mathbf{z}_{n,j}^{(t)}, \mathbf{y}_{1:s}\}, \{\mathbf{d}_n^{(t)}, \boldsymbol{\xi}_{1:s}\}, \boldsymbol{\theta}_{n,j}^{(t)}) p(\mathbf{y}_{1:s}|\boldsymbol{\theta}_{n,j}^{(t)}, \boldsymbol{\xi}_{1:s}) p(\boldsymbol{\theta}_{n,j}^{(t)})}{g_{IS}(\mathbf{d}_n^{(t)}|\mathbf{d}_n^{(t-1)}) \prod_{j=1}^{J(t)} k(\boldsymbol{\theta}_{n,j}^{(t)}|\mathbf{y}_{1:s}, \boldsymbol{\xi}_{1:s})}. \quad (3.3)$$

The acceptance probability for the MH step corresponds to formula (3.2).

Replacing $p(\boldsymbol{\theta}_{n,j}^{(t-1)})$ by $k(\boldsymbol{\theta}_{n,j}^{(t-1)}|\mathbf{y}_{1:s}, \boldsymbol{\xi}_{1:s})$ in the proposal distribution of the resampling-Markov algorithm, the weights (2.6) change to

$$w_n^{(t)} \propto w_n^{(t-1)} \prod_{j=J(t-1)+1}^{J(t)} \frac{u(\{\mathbf{z}_{n,j}^{(t-1)}, \mathbf{y}_{1:s}\}, \{\mathbf{d}_n^{(t-1)}, \boldsymbol{\xi}_{1:s}\}, \boldsymbol{\theta}_{n,j}^{(t-1)}) p(\mathbf{y}_{1:s}|\boldsymbol{\theta}_{n,j}^{(t-1)}, \boldsymbol{\xi}_{1:s}) p(\boldsymbol{\theta}_{n,j}^{(t-1)})}{k(\boldsymbol{\theta}_{n,j}^{(t-1)}|\mathbf{y}_{1:s}, \boldsymbol{\xi}_{1:s})}. \quad (3.4)$$

The SMC algorithm in Section 2.4 also needs only minor adjustments if prior observations are included. The function $\phi(\mathbf{z}_j, \mathbf{d}, \boldsymbol{\theta}_j)$ defined by (2.3) is modified to

$$\phi(\mathbf{z}_j, \mathbf{d}, \boldsymbol{\theta}_j) = u(\{\mathbf{z}_j, \mathbf{y}_{1:s}\}, \{\mathbf{d}, \boldsymbol{\xi}_{1:s}\}, \boldsymbol{\theta}_j) p(\mathbf{z}_j|\boldsymbol{\theta}_j, \mathbf{d}) p(\mathbf{y}_{1:s}|\boldsymbol{\theta}_j, \boldsymbol{\xi}_{1:s}) p(\boldsymbol{\theta}_j),$$

and the MH kernels $\mathcal{K}_{J(t)-1,1}$ and independence proposals q_{ν_t} have to be adapted accordingly.

4. Approximate Bayesian computing (ABC)

4.1. General methodology

In complex models where an explicit form of the likelihood function $p(\mathbf{z}|\boldsymbol{\theta}, \mathbf{d})p(\mathbf{y}_{1:s}|\boldsymbol{\theta}, \boldsymbol{\xi}_{1:s})$ is not available or it is very cumbersome to evaluate, one may have to resort to *likelihood-free (LF)* methods, also called *approximate Bayesian computing (ABC)*. These methods can be applied if simulating the data from the probability model is feasible for every parameter $\boldsymbol{\theta}$. Instances where likelihood-free methods have been successfully applied include biogenetics (Marjoram et al. 2003, Beaumont et al. 2002), ecology (Jabot and Chave 2009), quantile distributions (Drovandi and Pettitt 2011), models for extremes (Erhardt and Smith 2012, Bortot et al. 2007), and many other cases. For some further examples see Sisson and Fan (2011).

The easiest case is likelihood-free rejection sampling. Sampling from the posterior distribution $p(\boldsymbol{\theta}|\mathbf{y})$ can be accomplished by drawing the parameters from the prior distribution, $\boldsymbol{\theta}' \sim p(\boldsymbol{\theta})$, drawing a variable \mathbf{x} from the probability model, $\mathbf{x}' \sim p(\mathbf{x}|\boldsymbol{\theta}')$, and accepting $\boldsymbol{\theta}'$ if $\mathbf{x}' \approx \mathbf{y}$. Thus,

direct sampling from the posterior distribution is replaced by sampling from the prior distribution. The efficiency of this approach crucially depends on the similarity between the posterior and the prior distribution, i.e. the information gain of the posterior compared to the prior distribution.

Likelihood-free MCMC builds upon the same general idea. The original posterior target distribution $p(\boldsymbol{\theta}|\mathbf{y})$ is approximated by the marginal of the augmented distribution

$$p_{LF}(\boldsymbol{\theta}, \mathbf{x}|\mathbf{y}) \propto p_{\epsilon}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})p(\mathbf{x}|\boldsymbol{\theta})p(\boldsymbol{\theta}). \quad (4.1)$$

The variable \mathbf{x} , which is sampled together with $\boldsymbol{\theta}$, is added to the posterior arguments. Integrating over \mathbf{x} leads to the original posterior distribution if $p_{\epsilon}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$ is a point mass at the point $\mathbf{x} = \mathbf{y}$. Since this event has a very small probability for higher-dimensional discrete distributions and probability zero in the case of continuous distributions, a compromise has to be found between exactness and practicality by adjusting the “narrowness” of $p_{\epsilon}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$. The function $p_{\epsilon}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$ is usually assumed to be a smoothing kernel density function: $p_{\epsilon}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = (1/\epsilon)K(|\mathbf{T}(\mathbf{x}) - \mathbf{T}(\mathbf{y})|/\epsilon)$, where $\mathbf{T}(\cdot)$ is some low-dimensional statistic of \mathbf{y} and \mathbf{x} , respectively. The parameter ϵ controls the tightness of $p_{\epsilon}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})$. If \mathbf{T} is a sufficient statistic for the parameters of the probability model, integrating over $\mathbf{T}(\mathbf{x})$ yields the same distribution as integrating out \mathbf{x} .

Popular choices for $K(\cdot)$ are uniform, Gaussian, or Epanechnikov kernels. The uniform kernel is defined by

$$p_{\epsilon}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) \propto \mathbb{I}_{A_{\epsilon, \mathbf{y}}}(\mathbf{x}),$$

where $\mathbb{I}_{A_{\epsilon, \mathbf{y}}}(\mathbf{x})$ is the indicator function which takes the value 1 if $\mathbf{x} \in A_{\epsilon, \mathbf{y}}$ and 0 otherwise. The neighborhood set $A_{\epsilon, \mathbf{y}}$ is defined as $A_{\epsilon, \mathbf{y}} = \{\mathbf{x} \in \mathcal{Y} : \rho(\mathbf{T}(\mathbf{x}), \mathbf{T}(\mathbf{y})) \leq \epsilon\}$, where ρ denotes a distance measure between $\mathbf{T}(\mathbf{x})$ and $\mathbf{T}(\mathbf{y})$.

The reason for complicating the model artificially is that for the proposal distribution

$$q_{LF}(\boldsymbol{\theta}', \mathbf{x}'|\boldsymbol{\theta}^{(t-1)}, \mathbf{x}^{(t-1)}) = q_{\boldsymbol{\theta}}(\boldsymbol{\theta}'|\boldsymbol{\theta}^{(t-1)})p(\mathbf{x}'|\boldsymbol{\theta}'),$$

the likelihood terms $p(\mathbf{x}^{(t-1)}|\boldsymbol{\theta}^{(t-1)})$ and $p(\mathbf{x}'|\boldsymbol{\theta}')$ cancel out in the acceptance probability

$$\begin{aligned} \alpha &= \min \left(1, \frac{p_{\epsilon}(\mathbf{y}'|\mathbf{x}', \boldsymbol{\theta}')p(\mathbf{x}'|\boldsymbol{\theta}')p(\boldsymbol{\theta}')}{p_{\epsilon}(\mathbf{y}^{(t-1)}|\mathbf{x}^{(t-1)}, \boldsymbol{\theta}^{(t-1)})p(\mathbf{x}^{(t-1)}|\boldsymbol{\theta}^{(t-1)})p(\boldsymbol{\theta}^{(t-1)})} \frac{q_{\boldsymbol{\theta}}(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{\theta}')p(\mathbf{x}^{(t-1)}|\boldsymbol{\theta}^{(t-1)})}{q_{\boldsymbol{\theta}}(\boldsymbol{\theta}'|\boldsymbol{\theta}^{(t-1)})p(\mathbf{x}'|\boldsymbol{\theta}')} \right) \\ &= \min \left(1, \frac{p_{\epsilon}(\mathbf{y}'|\mathbf{x}', \boldsymbol{\theta}')p(\boldsymbol{\theta}')}{p_{\epsilon}(\mathbf{y}^{(t-1)}|\mathbf{x}^{(t-1)}, \boldsymbol{\theta}^{(t-1)})p(\boldsymbol{\theta}^{(t-1)})} \frac{q_{\boldsymbol{\theta}}(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{\theta}')}{q_{\boldsymbol{\theta}}(\boldsymbol{\theta}'|\boldsymbol{\theta}^{(t-1)})} \right). \end{aligned}$$

and it is no longer necessary to evaluate the likelihood. As in the simple rejection sampling case, if prior and posterior distribution have very different shapes, the rejection rates will be very close to 1.

In order to reduce the variance of the MH acceptance ratio and thereby improve the sampler's performance, Del Moral et al. (2012) propose an augmented likelihood-free posterior distribution given by

$$p_{LF}(\boldsymbol{\theta}, \mathbf{x}_{1:M}|\mathbf{y}) \propto \left[\frac{1}{M} \sum_{m=1}^M p_{\epsilon}(\mathbf{y}|\mathbf{x}_m, \boldsymbol{\theta}) \right] \left[\prod_{m=1}^M p(\mathbf{x}_m|\boldsymbol{\theta}) \right] p(\boldsymbol{\theta}),$$

see also Sisson and Fan (2011). If $M \rightarrow \infty$, then $\frac{1}{M} \sum_{m=1}^M p_{\epsilon}(\mathbf{y}|\mathbf{x}_m, \boldsymbol{\theta}) \rightarrow \int p_{\epsilon}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) p(\mathbf{x}|\boldsymbol{\theta}) d\mathbf{x}$, and the sampler is similar to the “marginal” sampler of the marginal of the likelihood-free posterior (4.1) where \mathbf{x} is integrated out.

Choosing a proposal distribution of the form

$$q(\boldsymbol{\theta}', \mathbf{x}'_{1:M}|\boldsymbol{\theta}^{(t-1)}, \mathbf{x}_{1:M}^{(t-1)}) = q_{\boldsymbol{\theta}}(\boldsymbol{\theta}'|\boldsymbol{\theta}^{(t-1)}) \prod_{m=1}^M p(\mathbf{x}'_m|\boldsymbol{\theta}')$$

gives the MH acceptance probability

$$\alpha = \min \left(1, \frac{\sum_{m=1}^M p_{\epsilon}(\mathbf{y}'|\mathbf{x}'_m, \boldsymbol{\theta}')}{\sum_{m=1}^M p_{\epsilon}(\mathbf{y}^{(t-1)}|\mathbf{x}_m^{(t-1)}, \boldsymbol{\theta}^{(t-1)})} \frac{p(\boldsymbol{\theta}')}{p(\boldsymbol{\theta}^{(t-1)})} \frac{q_{\boldsymbol{\theta}}(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{\theta}')}{q_{\boldsymbol{\theta}}(\boldsymbol{\theta}'|\boldsymbol{\theta}^{(t-1)})} \right).$$

Note that this algorithm reduces to the original ABC algorithm if $M = 1$.

A comprehensive account of likelihood-free MCMC is given in Sisson and Fan (2011).

4.2. ABC for simulation-based optimal design

There are two possible ways to deal with non-available likelihood functions in simulation-based optimal design problems when prior observations are present. If these prior observations are accounted for in the target distribution, the likelihood functions do not cancel out in the MH acceptance probability formula (see (3.2)) or in the weights or weight updates of (sequential) importance samplers (see (3.3) and (3.4)).

Modifying the target distribution

One solution would be to augment the target distribution (3.1) to

$$h_{LF,J}(\boldsymbol{\vartheta}_j, \mathbf{x}_{1:s,1:M,1:J}) \propto \prod_{j=1}^J u(\{\mathbf{z}_j, \mathbf{y}_{1:s}\}, \{\mathbf{d}, \boldsymbol{\xi}_{1:s}\}, \boldsymbol{\theta}_j) p(\mathbf{z}_j | \boldsymbol{\theta}_j, \mathbf{d}) \times \\ \left[\frac{1}{M} \sum_{m=1}^M p_{\epsilon}(\mathbf{y}_{1:s} | \mathbf{x}_{1:s,m,j}, \boldsymbol{\theta}_j) \right] \left[\prod_{m=1}^M p(\mathbf{x}_{1:s,m,j} | \boldsymbol{\theta}_j, \boldsymbol{\xi}_{1:s}) \right] p(\boldsymbol{\theta}_j) \mu(\mathbf{d}),$$

where $\mathbf{x}_{1:s,1:M}$ denotes a collection of M draws of $\mathbf{x}_{1:s}$.

A sensible and convenient choice for the proposal distribution is

$$q_{LF}(\boldsymbol{\vartheta}'_j, \mathbf{x}'_{1:s,1:M,1:J} | \mathbf{d}^{(t-1)}) = g(\mathbf{d}' | \mathbf{d}^{(t-1)}) \prod_{j=1}^J p(\mathbf{z}'_j | \boldsymbol{\theta}'_j, \mathbf{d}') \left[\prod_{m=1}^M p(\mathbf{x}'_{1:s,m,j} | \boldsymbol{\theta}'_j, \boldsymbol{\xi}_{1:s}) \right] p(\boldsymbol{\theta}'_j), \quad (4.2)$$

which leads to the MH acceptance probability

$$\alpha = \min \left(1, \frac{\prod_{j=1}^{J(t)} u(\{\mathbf{z}'_j, \mathbf{y}_{1:s}\}, \{\mathbf{d}', \boldsymbol{\xi}_{1:s}\}, \boldsymbol{\theta}'_j)}{\prod_{j=1}^{J(t)} u(\{\mathbf{z}_j^{(t-1)}, \mathbf{y}_{1:s}\}, \{\mathbf{d}^{(t-1)}, \boldsymbol{\xi}_{1:s}\}, \boldsymbol{\theta}_j^{(t-1)})} \times \right. \\ \left. \frac{\prod_{j=1}^{J(t)} \left[\frac{1}{M} \sum_{m=1}^M p_{\epsilon}(\mathbf{y}_{1:s} | \mathbf{x}'_{1:s,m,j}, \boldsymbol{\theta}'_j) \right]}{\prod_{j=1}^{J(t)} \left[\frac{1}{M} \sum_{m=1}^M p_{\epsilon}(\mathbf{y}_{1:s} | \mathbf{x}_{1:s,m,j}^{(t-1)}, \boldsymbol{\theta}_j^{(t-1)}) \right]} \frac{g(\mathbf{d}^{(t-1)} | \mathbf{d}')}{g(\mathbf{d}' | \mathbf{d}^{(t-1)})} \right).$$

Using an importance distribution similar to (4.2) results in weights of

$$w_n^{(t)} \propto \frac{\prod_{j=1}^{J(t)} u(\{\mathbf{z}_j^{(t)}, \mathbf{y}_{1:s}\}, \{\mathbf{d}^{(t)}, \boldsymbol{\xi}_{1:s}\}, \boldsymbol{\theta}_j^{(t)}) \left[\frac{1}{M} \sum_{m=1}^M p_{\epsilon}(\mathbf{y}_{1:s} | \mathbf{x}_{1:s,m,j}^{(t)}, \boldsymbol{\theta}_j^{(t)}) \right]}{g_{IS}(\mathbf{d}^{(t)} | \mathbf{d}^{(t-1)})}$$

for the sampling importance resampling sampler (cf. equation (3.3)) and

$$w_n^{(t)} \propto w_n^{(t-1)} \prod_{j=J(t-1)+1}^{J(t)} u(\{\mathbf{z}_j^{(t-1)}, \mathbf{y}_{1:s}\}, \{\mathbf{d}^{(t-1)}, \boldsymbol{\xi}_{1:s}\}, \boldsymbol{\theta}_j^{(t-1)}) \left[\frac{1}{M} \sum_{m=1}^M p_{\epsilon}(\mathbf{y}_{1:s} | \mathbf{x}_{1:s,m,j}^{(t-1)}, \boldsymbol{\theta}_j^{(t-1)}) \right]$$

for the resampling-Markov algorithm (cf. equation (3.4)). The subscripts n for the n^{th} particle are omitted to simplify notation.

If M is low, the acceptance probability will be close to 0 for most iterations so that the Markov chain is highly persistent. If an importance sampler is applied instead, a small M will lead to very imbalanced weights (many weights will be 0) and therefore to a low ESS. On the other hand, choosing a sufficiently high M may be computationally prohibitive, in particular if sampling the data is costly.

When using particle methods, rejection control techniques can be employed to alleviate the problem of imbalanced weights (see e.g. Liu 2001, Sisson et al. 2007, and Beaumont et al. 2009). Particles with low weights are discarded and resampled from the particle set at time $t - 1$ according to the weight distribution. Therefore, rejection control techniques may also require considerable additional computational resources if the size of the particle set is required to stay constant over time. Moreover, it may be very hard to select a proper value for ϵ or a proper schedule $\{\epsilon_t, t = 1, \dots, T\}$ (typically monotonically decreasing) and assess its efficiency for a complex target distribution such as $h_{LF, J(t), \epsilon_t}$. There are many other factors that crucially affect the efficiency of the sampler, for example the choice of M or the schedule for $J(t)$.

Estimating the posterior distribution separately

A second potential approach to solving the simulation-based design problem with prior observations and non-available likelihood functions is to divide the problem into smaller tasks.

In a first step, the posterior distribution

$$p(\boldsymbol{\theta} | \mathbf{y}_{1:s}, \boldsymbol{\xi}_{1:s}) \propto \prod_{i=1}^s p(\mathbf{y}_i | \boldsymbol{\theta}, \boldsymbol{\xi}_i) p(\boldsymbol{\theta})$$

may be approximated by any suitable method. One possible method is the sequential Monte Carlo algorithm dealing with general ABC problems developed in Del Moral et al. (2012). Their algorithm automatically adapts the schedule for ϵ_t . The complexity of this algorithm is $O(R)$, where R denotes the number of particles used for approximating $p(\boldsymbol{\theta} | \mathbf{y}_{1:s}, \boldsymbol{\xi}_{1:s})$. An alternative importance sampler for the same task which is based on population Monte Carlo ideas (Cappé et al. 2004) is proposed by Sisson et al. (2007) and Beaumont et al. (2009). The latter improve the method of Sisson et al. (2007) by taking care of the bias that is engendered by the originally proposed method. Both algorithms also make use of rejection control, whereas Del Moral et al. (2012) use a sufficiently high simulation number M to improve sampling quality. The algorithm of Beaumont et al. (2009) has a complexity of $O(R^2)$.

The final particle set delivered by any of these particle methods can be regarded as a discrete approximation to the true posterior distribution. The weights correspond to the probability mass function of the posterior approximation. One may then replace the exact posterior distribution

$p(\boldsymbol{\theta}|\mathbf{y}_{1:s}, \boldsymbol{\xi}_{1:s})$ with this approximate distribution $\hat{p}_{LF}(\boldsymbol{\theta}|\mathbf{y}_{1:s}, \boldsymbol{\xi}_{1:s})$, which is the marginal distribution of $\hat{p}_{LF}(\boldsymbol{\theta}, \mathbf{x}_{1:s,1:M}|\mathbf{y}_{1:s}, \boldsymbol{\xi}_{1:s})$, in the algorithms described in Sections 2.3 and 2.4. Sampling from $\hat{p}_{LF}(\boldsymbol{\theta}|\mathbf{y}_{1:s}, \boldsymbol{\xi}_{1:s})$ is easily accomplished, for example, by discrete sampling from the weighted particle set $\{\boldsymbol{\theta}_r^{(L)}, r = 1, \dots, R\}$. The particle sampler for estimating $p(\boldsymbol{\theta}|\mathbf{y}_{1:s}, \boldsymbol{\xi}_{1:s})$ comprises R particles and iterates over L time steps.

A way to obtain an approximate closed formula for the posterior distribution is the non-parametric conditional density estimation approach of Bonassi et al. (2011). They fit a mixture of multivariate normals to the joint distribution of the summary statistics and the parameters $(\mathbf{T}(\mathbf{x}), \boldsymbol{\theta})$ based on the samples $\{(\mathbf{T}(\mathbf{x}^{(l)}), \boldsymbol{\theta}^{(l)}), l = 1, \dots, L\}$ and condition on the summary statistics to get a normal mixture approximation of the posterior. A similar approach is taken by Fan et al. (2012) to obtain an analytic representation of the likelihood function. The likelihood function can then be combined with different priors, for example in order to conduct a sensitivity analysis. Thus, estimating the likelihood instead of the posterior offers greater flexibility. Fan et al. (2012) use the samples $\{(\mathbf{T}(\mathbf{x}^{(l)}), \boldsymbol{\theta}^{(l)}), l = 1, \dots, L\}$ to fit univariate mixtures of normals to the summary statistics. The normal mixtures depend on the parameters $\boldsymbol{\theta}$. In order to account for the dependence structure between the summary statistics and the parameters, they use a mixture of Gaussian copulas for the standardized summary statistics and the parameters. Their method works quite well even for higher-dimensional summary statistics.

5. Utility function

The choice of the utility function $u(\{\mathbf{z}, \mathbf{y}_{1:s}\}, \{\mathbf{d}, \boldsymbol{\xi}_{1:s}\}, \boldsymbol{\theta})$ is highly problem-specific. The experimenter has great freedom in selecting an appropriate utility function. However, the utility function has to be bounded and non-negative, otherwise h_J will not be a proper probability distribution.

It is possible to construct very customized utility functions as Müller et al. (2004) do for their network design examples (a rainfall network and an ozone monitoring network). Their utility functions incorporate information about the number of future observations that are correctly covered by a small confidence interval around the predicted response value for stations that are not in the network, or the number of future observations that are correctly estimated to be above a certain threshold value if interest is in predicting extreme events. Furthermore, they incorporate the sum of squared

distances of the future observations from the predicted response values for all stations that are not in the network, and include the variation of the response surface around its estimated value. Lastly, they also allow for the costs of the number of stations to enter into the utility function.

If the different designs can be interpreted as alternative models, it is even possible to perform some kind of model evaluation and selection by comparing the simulated \mathbf{z} to the past observations $\mathbf{y}_{1:s}$.

There is often a need to base the utility on parameters of the updated posterior distribution $p(\boldsymbol{\theta}|\{\mathbf{z}, \mathbf{y}_{1:s}\}, \{\mathbf{d}, \boldsymbol{\xi}_{1:s}\})$, like the variance of $\boldsymbol{\theta}$ or the Kullback-Leibler divergence. Sometimes it may even be of interest to include parameters of the updated posterior predictive distribution $\int p(\mathbf{z}_p|\boldsymbol{\theta}, \mathbf{d}_p)p(\boldsymbol{\theta}|\{\mathbf{z}, \mathbf{y}_{1:s}\}, \{\mathbf{d}, \boldsymbol{\xi}_{1:s}\})d\boldsymbol{\theta}$, like the estimated mean response at some points in the design space. These parameters cannot be used in the utility function unless there is an analytical expression for them which is very unusual for the kinds of problems where ABC methods are used. However, if an approximate sample from the posterior distribution $p(\boldsymbol{\theta}|\mathbf{y}_{1:s}, \boldsymbol{\xi}_{1:s})$ is available, for example by performing a particle sampler beforehand, the posterior distribution can be updated rather quickly.

Assume that a weighted set of particles $\{(\boldsymbol{\theta}_r^{(L)}, W_r^{(L)}), r = 1, \dots, R\}$ distributed according to the approximate distribution $\hat{p}_{LF}(\boldsymbol{\theta}|\mathbf{y}_{1:s}, \boldsymbol{\xi}_{1:s}) \approx p(\boldsymbol{\theta}|\mathbf{y}_{1:s}, \boldsymbol{\xi}_{1:s})$ is available. The updated posterior distribution is then

$$\hat{p}_{LF}(\boldsymbol{\theta}, \mathbf{x}_{1:M}|\{\mathbf{z}, \mathbf{y}_{1:s}\}, \{\mathbf{d}, \boldsymbol{\xi}_{1:s}\}) \propto \left[\frac{1}{M} \sum_{m=1}^M p_\epsilon(\mathbf{z}|\mathbf{x}_m, \boldsymbol{\theta}) \right] \left[\prod_{m=1}^M p(\mathbf{x}_m|\boldsymbol{\theta}, \mathbf{d}) \right] \hat{p}_{LF}(\boldsymbol{\theta}|\mathbf{y}_{1:s}, \boldsymbol{\xi}_{1:s}).$$

The importance distribution is constructed by sampling $\boldsymbol{\theta}_r^{(L+1)}$ from the approximate posterior and sampling $\{\mathbf{x}_m^{(L+1)}, m = 1, \dots, M\}$ from $p(\mathbf{x}_m|\boldsymbol{\theta}, \mathbf{d})$, that is

$$q_{\boldsymbol{\theta}, LF}(\boldsymbol{\theta}, \mathbf{x}_{1:M}|\{\mathbf{z}, \mathbf{y}_{1:s}\}, \{\mathbf{d}, \boldsymbol{\xi}_{1:s}\}) = \left[\prod_{m=1}^M p(\mathbf{x}_m|\boldsymbol{\theta}, \mathbf{d}) \right] \hat{p}_{LF}(\boldsymbol{\theta}|\mathbf{y}_{1:s}, \boldsymbol{\xi}_{1:s}),$$

so one arrives at the new importance weights for the updated approximate posterior:

$$w_r^{(L+1)} \propto \sum_{m=1}^M p_\epsilon(\mathbf{z}|\mathbf{x}_m, \boldsymbol{\theta}).$$

This whole procedure has to be repeated $J(t)$ times at each time step $t = 1, \dots, T$ if a simulated-annealing-type extension is employed. In addition, rejection control techniques may be applied to ensure that only particles with non-zero weights are sampled (see Liu 2001).

Given the particles of the updated approximate posterior, one can either construct summary statistics like the variance of $\theta|\{\mathbf{z}, \mathbf{y}_{1:s}\}, \{\mathbf{d}, \xi_{1:s}\}$ (cf. the Bayesian D-posterior precision utility used by Drovandi et al. 2013) or simulate from the updated posterior predictive distribution $\int p(\mathbf{z}_p|\theta, \mathbf{d}_p)p(\theta|\{\mathbf{z}, \mathbf{y}_{1:s}\}, \{\mathbf{d}, \xi_{1:s}\})d\theta$ by sampling $\theta_p^{(1:L)}$ from $\hat{p}_{LF}(\theta|\{\mathbf{z}, \mathbf{y}_{1:s}\}, \{\mathbf{d}, \xi_{1:s}\})$ and then sampling $\mathbf{z}_p^{(1:L)}$ from the likelihood. A simple parameter like the mean response at design point \mathbf{d}_p can then be calculated by averaging the draws of $\mathbf{z}_p^{(1:L)}$. A single forecast is created by setting $L = 1$. Solonen et al. (2012) follow a similar approach in order to estimate a response variance criterion for their utility function.

An alternative way to update the posterior distribution would be to approximate the likelihood function by a multivariate mixture of normals as in Fan et al. (2012). It is then straightforward to combine the analytic expressions for the likelihood of $\{\mathbf{z}, \mathbf{y}_{1:s}\}$ and for the prior distribution of the parameters to obtain an update of the posterior distribution, for example by creating a sample from the updated posterior by standard MCMC sampling. Calculating the updated posterior distribution, however, may be rather computer intensive, which diminishes the usefulness of this method if a new updated posterior has to be computed for every evaluation of $u(\cdot)$.

A very important standard utility criterion which depends on the updated posterior is the Kullback-Leibler divergence, also often called Shannon information:

$$\text{KLD}(\mathbf{z}, \mathbf{d}) = \int_{\theta \in \Theta} \log \left(\frac{p(\theta|\{\mathbf{z}, \mathbf{y}_{1:s}\}, \{\mathbf{d}, \xi_{1:s}\})}{p(\theta)} \right) p(\theta|\{\mathbf{z}, \mathbf{y}_{1:s}\}, \{\mathbf{d}, \xi_{1:s}\}) d\theta.$$

The KLD may also be computed as distance between $p(\theta|\{\mathbf{z}, \mathbf{y}_{1:s}\}, \{\mathbf{d}, \xi_{1:s}\})$ and $p(\theta|\mathbf{y}_{1:s}, \xi_{1:s})$.

Chaloner and Verdinelli (1995) point out that for a standard linear model the average Kullback-Leibler distance $\int_{\mathbf{z} \in \mathcal{Y}} \text{KLD}(\mathbf{z}, \mathbf{d})p(\mathbf{z}|\mathbf{d})d\mathbf{z}$ corresponds to the classical D-criterion. In the case of non-linear models, the average Kullback-Leibler distance still gives a reasonable approximation to the Bayesian D-criterion involving the Fisher information matrix if the posterior distribution of θ is approximately normally distributed. Moreover, the Kullback-Leibler distance is more robust than utilities which only depend on the variance of $\theta|\{\mathbf{z}, \mathbf{y}_{1:s}\}, \{\mathbf{d}, \xi_{1:s}\}$, especially if the posterior is multi-modal.

Computing the KLD makes it necessary to obtain $p(\theta|\{\mathbf{z}, \mathbf{y}_{1:s}\}, \{\mathbf{d}, \xi_{1:s}\})$ or at least an approximation to it. This can be achieved by the posterior updating approaches described above.

As an example, Kück et al. use Shannon information as utility function for their SMC sampler.

They refer to Sebastiani and Wynn (1997) who show that under some assumptions such as stationarity

$$\int_{\boldsymbol{\theta} \in \Theta} \int_{\mathbf{z} \in \mathcal{Y}} \int_{\boldsymbol{\theta}' \in \Theta} p(\boldsymbol{\theta}) p(\mathbf{z} | \boldsymbol{\theta}, \mathbf{d}) [p(\boldsymbol{\theta}' | \mathbf{z}, \mathbf{d}) \log p(\boldsymbol{\theta}' | \mathbf{z}, \mathbf{d})] d\boldsymbol{\theta}' d\mathbf{z} d\boldsymbol{\theta}$$

is equal to the negative entropy of the marginal with respect to \mathbf{z} ,

$$C - \int_{\mathbf{z} \in \mathcal{Y}} p(\mathbf{z} | \mathbf{d}) \log p(\mathbf{z} | \mathbf{d}) d\mathbf{z},$$

which helps to reduce the complexity of the SMC sampler of Kück et al. to some degree.

6. Example

6.1. Example setting

We apply the simulation-based design methodology developed in the previous sections to a standard linear regression example. In that case the likelihood function is of a well-known and simple form, so there is no need to invoke likelihood-free methods. The purpose of our example is merely to demonstrate various important aspects one has to consider when applying simulation-based design algorithms with likelihood-free extensions. For our example it is very easy to calculate the posterior distribution given previous observations, so it is possible to compare the likelihood-free approximations of the posterior distribution to the exact posterior distribution. Moreover, here the expected utility integral can also be computed analytically. This allows us to compare the results from the simulation-based optimal design algorithms to the exact results. Thus we are able to analyze the sensitivity of the result to the approximation quality and to what extent approximation quality depends on the settings of the ABC algorithms.

We assume that

$$\mathbf{z} | \boldsymbol{\theta}, \mathbf{d} \sim \mathcal{N}(\mathbf{D}\boldsymbol{\theta}, \sigma^2 \mathbf{I}_n).$$

That is, the expected value of the dependent variable is a linear combination of the parameter values $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^k$ and depends on the design through the design matrix $\mathbf{D} = (\mathbf{f}(d_1), \dots, \mathbf{f}(d_n))^T$, where $\mathbf{f}(\cdot)$ is a k -dimensional function of the design variables $d_i \in [-1, 1]$ and $\mathbf{d} = (d_1, \dots, d_n)$. The n observations are assumed to be normally distributed, independent, and homoscedastic with known variance σ^2 .

We assume that s previous observations $\mathbf{y} = (y_1, \dots, y_s)$ have been collected which follow the same distribution:

$$\mathbf{y}|\boldsymbol{\theta}, \boldsymbol{\xi} \sim \mathcal{N}(\mathbf{K}\boldsymbol{\theta}, \sigma^2 \mathbf{I}_s),$$

where $\mathbf{K} = (\mathbf{f}(\xi_1), \dots, \mathbf{f}(\xi_s))^T$ and $\boldsymbol{\xi} = (\xi_1, \dots, \xi_s)$.

Furthermore, the parameters $\boldsymbol{\theta}$ follow the prior normal distribution

$$\boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{\theta}_0, \sigma^2 \mathbf{R}^{-1}).$$

The posterior distribution of $\boldsymbol{\theta}$ given the previous observations can easily be computed. Introducing $\mathbf{L} = \mathbf{K}^T \mathbf{K}$, the posterior is given by

$$\boldsymbol{\theta}|\mathbf{y}, \boldsymbol{\xi} \sim \mathcal{N}((\mathbf{L} + \mathbf{R})^{-1}(\mathbf{K}^T \mathbf{y} + \mathbf{R}\boldsymbol{\theta}_0), \sigma^2(\mathbf{L} + \mathbf{R})^{-1}).$$

In the same way, the updated posterior distribution can be obtained as

$$\boldsymbol{\theta}|\{\mathbf{z}, \mathbf{y}\}, \{\mathbf{d}, \boldsymbol{\xi}\} \sim \mathcal{N}((\mathbf{M} + \mathbf{L} + \mathbf{R})^{-1}(\mathbf{D}^T \mathbf{z} + \mathbf{K}^T \mathbf{y} + \mathbf{R}\boldsymbol{\theta}_0), \sigma^2(\mathbf{M} + \mathbf{L} + \mathbf{R})^{-1}),$$

where $\mathbf{M} = \mathbf{D}^T \mathbf{D}$.

We take $u(\mathbf{z}, \mathbf{d}, \boldsymbol{\theta}) = \log\left(\frac{p(\boldsymbol{\theta}|\{\mathbf{z}, \mathbf{y}\}, \{\mathbf{d}, \boldsymbol{\xi}\})}{p(\boldsymbol{\theta})}\right)$ as our utility function, so that the expected utility for a specific design \mathbf{d} is the expected gain in Shannon information (see Chaloner and Verdinelli 1995):

$$\begin{aligned} U(\mathbf{d}) &= \int_{\mathbf{z} \in \mathbb{R}^n} \int_{\boldsymbol{\theta} \in \Theta} u(\mathbf{z}, \mathbf{d}, \boldsymbol{\theta}) p(\mathbf{z}, \boldsymbol{\theta}|\mathbf{y}, \{\mathbf{d}, \boldsymbol{\xi}\}) d\boldsymbol{\theta} d\mathbf{z} \\ &= \int_{\mathbf{z} \in \mathbb{R}^n} \int_{\boldsymbol{\theta} \in \Theta} \log\left(\frac{p(\boldsymbol{\theta}|\{\mathbf{z}, \mathbf{y}\}, \{\mathbf{d}, \boldsymbol{\xi}\})}{p(\boldsymbol{\theta})}\right) p(\mathbf{z}|\boldsymbol{\theta}, \mathbf{d}) p(\boldsymbol{\theta}|\mathbf{y}, \boldsymbol{\xi}) d\boldsymbol{\theta} d\mathbf{z}. \end{aligned}$$

Since $\int \int \log(p(\boldsymbol{\theta})) p(\mathbf{z}|\boldsymbol{\theta}, \mathbf{d}) p(\boldsymbol{\theta}|\mathbf{y}, \boldsymbol{\xi}) d\boldsymbol{\theta} d\mathbf{z} = \int \log(p(\boldsymbol{\theta})) p(\boldsymbol{\theta}|\mathbf{y}, \boldsymbol{\xi}) (\int p(\mathbf{z}|\boldsymbol{\theta}, \mathbf{d}) d\mathbf{z}) d\boldsymbol{\theta}$ does not depend on \mathbf{d} , for optimization it would be sufficient to compute

$$\begin{aligned} U^*(\mathbf{d}) &= \int_{\mathbf{z} \in \mathbb{R}^n} \int_{\boldsymbol{\theta} \in \Theta} u^*(\mathbf{z}, \boldsymbol{\theta}, \mathbf{d}) p(\mathbf{z}|\boldsymbol{\theta}, \mathbf{d}) p(\boldsymbol{\theta}|\mathbf{y}, \boldsymbol{\xi}) d\boldsymbol{\theta} d\mathbf{z} \\ &= \int_{\mathbf{z} \in \mathbb{R}^n} \int_{\boldsymbol{\theta} \in \Theta} \log(p(\boldsymbol{\theta}|\{\mathbf{z}, \mathbf{y}\}, \{\mathbf{d}, \boldsymbol{\xi}\})) p(\mathbf{z}|\boldsymbol{\theta}, \mathbf{d}) p(\boldsymbol{\theta}|\mathbf{y}, \boldsymbol{\xi}) d\boldsymbol{\theta} d\mathbf{z}. \end{aligned}$$

For our particular model, the integral can be computed analytically and is given as

$$U^*(\mathbf{d}) = -\frac{k}{2} \log(2\pi) - \frac{k}{2} + \frac{1}{2} \log \det(\sigma^{-2}(\mathbf{M} + \mathbf{L} + \mathbf{R})).$$

It has the same maximum as the criterion for D_B optimality, $\Psi(\mathbf{d}) = \det(\mathbf{M} + \mathbf{L} + \mathbf{R})$ (cf. Atkinson et al. 2007). Note that the D_B -optimal design does neither depend on σ^2 nor on the prior mean θ_0 nor on the previous observations \mathbf{y} .

We choose a setting for which the exact solution can be obtained easily, and thus a comparison of the results of our design algorithms is feasible.

The following setting is used: the predictor is a polynomial of order two in one factor, i.e.

$$\mathbf{D} = \begin{pmatrix} 1 & d_1 & d_1^2 \\ 1 & d_2 & d_2^2 \\ \vdots & \vdots & \vdots \\ 1 & d_n & d_n^2 \end{pmatrix} \quad \text{and} \quad \mathbf{K} = \begin{pmatrix} 1 & \xi_1 & \xi_1^2 \\ 1 & \xi_2 & \xi_2^2 \\ \vdots & \vdots & \vdots \\ 1 & \xi_s & \xi_s^2 \end{pmatrix}.$$

The continuous optimal design for this problem puts equal weights of $1/3$ on the three design points $-1, 0$, and 1 , see Atkinson et al. (2007). Likewise, if the number of trials of an exact design is divisible by three, then at the optimal design $1/3$ of the trials are set to $-1, 0$, and 1 , respectively. For our first example, we choose the prior information matrix \mathbf{R} in a way so that it represents prior information equivalent to one trial taken at the design point 0 , i.e. $\mathbf{R} = \mathbf{f}(0)\mathbf{f}^T(0) = (1, 0, 0)^T(1, 0, 0)$. A value of 10^{-5} is added to the diagonal elements to make \mathbf{R} invertible. Furthermore, we assume that one previous observation has been collected at the design point -1 , so that $\mathbf{L} = \mathbf{f}(-1)\mathbf{f}^T(-1) = (1, -1, 1)^T(1, -1, 1)$. If we have $n = 1$ (future) trial, it is optimal to set this trial to 1 .

6.2. MCMC sampler for augmented target distribution

We applied two different design algorithms to this example. The first one is an MCMC sampler where the target distribution for the expected utility function is augmented with the set of simulated previous observations $\mathbf{x}_{1:M,1:J}$. This is the first algorithm described in Section 4.2. We set the power of the expected utility function, J , to 1 and the number of repetitions of the simulated previous observations, M , also to 1 . The design dimension of this problem is one, so there is no need to use a higher power J or an increasing schedule for J . The mode of the sampled distribution can easily be read off from a histogram or a kernel density plot of the sample. As neighborhood kernel we use the uniform kernel:

$$p_\epsilon(y, x) \propto \mathbb{I}_{|y-x| < \epsilon}(x).$$

We use the uniform distribution on the interval $[-1, 1]$ as independence proposal distribution for $\mathbf{d} = d$. For our example this is a reasonable choice because the utility surface is rather flat. Furthermore, we set $\sigma^2 = 2$, $\boldsymbol{\theta}_0 = (0, 0, 0)^T$, and we assume that the previously collected observation at $\xi = -1$ is $y = 40$. Note that these parameters should have no effect on the outcome in our example.

We have run the algorithm for various values of ϵ ($\sigma, 2\sigma, 4\sigma, 8\sigma, 16\sigma$) and for various lengths of the Markov chain ($10^4, 10^8, 10^9$). Due to memory allocation constraints the output of the Markov chains of length 10^8 and 10^9 was thinned, keeping every 10^{th} and 100^{th} element of the chain, respectively.

The utility function $u(z, d, \boldsymbol{\theta})$ is not non-negative everywhere. If negative utilities occur, the simulation step is repeated until the sampled utility is positive. This modification distorts the output of the estimated utility surface, but we are only interested in regions of high expected utility anyway. If we use the utility function $u(z, d, \boldsymbol{\theta}) = \log(p(\boldsymbol{\theta}|\{z, y\}, \{d, \xi\})) - \log(p(\boldsymbol{\theta}))$, we do not observe many cases with negative utilities. One could also add a positive constant to the utility function if too many negative utilities are sampled.

The differences between the outputs for the various Markov chain lengths are hardly surprising. Especially for low ϵ , however, thinning does not reduce the integrated autocorrelation time (IAT)¹ proportionally. On the other hand, a Markov chain of length 10^4 is often too short to properly represent the expected utility surface because of the very low acceptance rates. We therefore have found the Markov chain sampler with 10^8 iterations to be more efficient than the sampler running for 10^9 iterations while still giving very accurate results. On a PC with an Intel Core i3 CPU (2.10 GHz) and 4 GB RAM the MCMC samplers with 10^8 iterations needed from 2.25 to 3 minutes to produce their sample.

Table 1 shows the acceptance rates and IATs for MCMC samplers with 10^8 iterations for different values of ϵ . A higher accuracy for the estimation of the posterior distribution is clearly associated with a higher IAT. One has to find a reasonable compromise between these two conflicting goals.

Figure 1 suggests that the choice of ϵ has no effect on the result. However, this is a special feature of our example and our choice of the utility function and not the case in general. In our example the

¹ The integrated autocorrelation time of a process is defined as $\text{IAT} = 1 + 2 \sum_{i=1}^{\infty} \rho_i$, where ρ_i denotes the autocorrelation of the process at lag i . It roughly says that one observation drawn from a hypothetical i.i.d. sampler with the same stationary distribution as the MCMC sampler is worth IAT observations drawn from the MCMC sampler. We estimate the IAT using the estimating procedure proposed by Geyer (1992).

Table 1: Acceptance rates and IATs for MCMC sampler on the augmented utility with 10^8 iterations for different values of ϵ .

ϵ	Accept. rate	IAT
1.41	0.0048	44.01
2.82	0.0092	22.75
5.64	0.0149	14.05
11.28	0.0163	12.93
22.56	0.0170	12.28

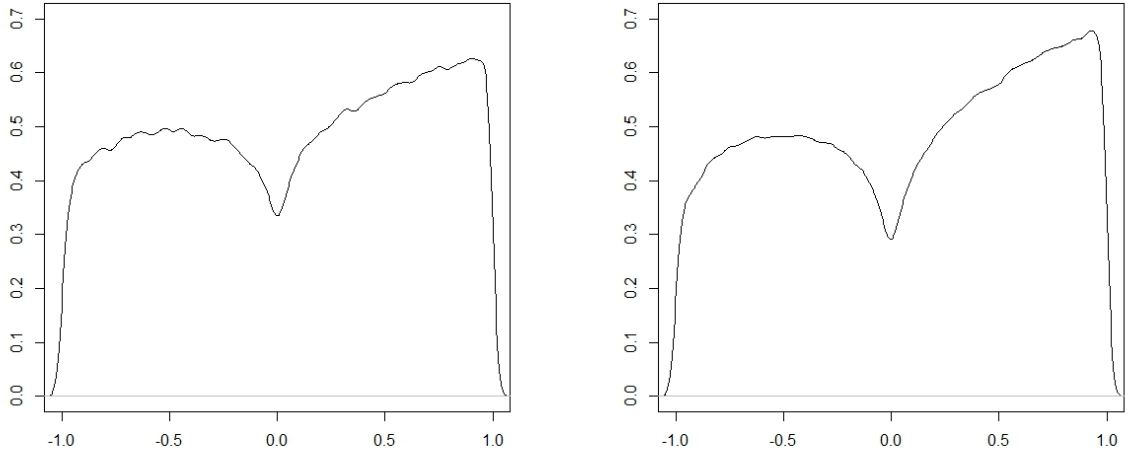


Figure 1: Kernel density estimates of marginal distribution of d for MCMC sampler on the augmented utility; 10^8 iterations; $\epsilon = 1.41$ (left) and $\epsilon = 22.56$ (right).

value of y does not matter for the optimal design, and hence it is irrelevant whether the simulated observations are close to the actual observations or not. Further, the utility is a function of the updated posterior. As the posterior is known explicitly, the sampled values are just plugged in to compute the utility function. If the posterior density function is not known, one of the approaches outlined in

Section 5 has to be pursued.

We have also found that using a higher power J proves problematic if this algorithm is used. In our example, the prior variance of θ is very high due to the lack of prior information, so the prior distribution $p(\theta)$ is very dissimilar to the posterior distribution $p(\theta|\mathbf{y}, \xi)$. Therefore, almost all simulated observations $\mathbf{x}_{m,j}$ fall outside some reasonable neighborhood of \mathbf{y} so that the rejection rates are very high. However, for the acceptance probability to be higher than zero, we have to have $\sum_{m=1}^M p_\epsilon(\mathbf{y}, \mathbf{x}_{m,j}) > 0 \quad \forall j \in \{1, \dots, J\}$. Even if we take a high number M , this is very unlikely to happen for each $j = 1, \dots, J$ if J is sufficiently high. Thus, this algorithm is not suitable for higher-dimensional design problems where it is essential to have $J \gg 1$.

6.3. Two-stage MCMC sampler

The considerations mentioned above lead us to pursue a two-stage approach, where we first estimate the posterior distribution $p(\theta|\mathbf{y}, \xi)$ by the population Monte Carlo scheme suggested by Beaumont et al. (2009), and then use this approximate posterior in a standard MCMC simulation-based design algorithm as proposed by Müller (1999).

The population Monte Carlo scheme for approximate Bayesian computation works by creating a set of N particles for which standard likelihood-free rejection sampling is applied. All particles have equal weights after the first step. Then in each consecutive step particles are created by sampling a particle from the previous step according to the weights, propagating this particle using a normal distribution with the mean set to the particle value at the previous step and the variance set to twice the empirical variance of the sample from the previous step. If the newly created particle is rejected, then a new particle is sampled from the particle set of the previous step and propagated. This is repeated until the particle gets accepted. This procedure implements a form of rejection control: in the final sample there are only particles with non-zero weights.

In each step the target distribution changes. The idea is to start with a target distribution with a rather high ϵ so that the first sample is obtained quickly. In the subsequent steps ϵ is slowly getting smaller until the approximation of the posterior distribution is sufficiently accurate. The algorithm is of order $O(N^2)$ because for computing the weights one has to iterate over the whole particle set obtained at the previous step. Other algorithms like the one developed by Del Moral et al. (2012)

only have complexity $O(N)$, but the population Monte Carlo sampler is rather reliable, unbiased, and easy to implement.

For the same setting as for the previous sampler we have computed the posterior distribution $p(\theta|y = 40, \xi = -1)$ using the population Monte Carlo method of Beaumont et al. (2009). The descending sequence of values for ϵ we used was $(16\sigma, 8\sigma, 4\sigma, 2\sigma, \sigma)$. The algorithm was run for particle sets of size 5000, 10000, and 20000. We know the exact posterior distribution, so we can compare the moments of the exact distribution to the moments of the weighted particle sample. The comparison of the mean, variance and covariance parameters shows that all different sample sizes deliver results that come very close to the exact posterior distribution, the deviations from the true values being less than 1 for the true mean parameters $(0.002, -19.998, 19.998)$, for example. The deviations from the true variance and especially covariance parameters are more notable, but it makes no difference how large the particle set is. There is not much gain in having a particle set of size 20000, which took 13.71 minutes to compute, compared to a particle set of size 10000, which was computed in 3.41 minutes (the complexity is $O(N^2)$).

Since we have already obtained a weighted sample of the posterior distribution of size 20000, we use this sample as an approximation to the posterior distribution for the second stage. In the second stage we perform an ordinary MCMC simulation-based design algorithm as described by Müller (1999). We vary the power of the expected utility function ($J = 50, 100, 200$) and the length of the Markov chain (10^5 and 10^4). We do not employ an increasing schedule for J but have it set to a fixed value. The independence uniform proposal distribution for d ensures that we still do not miss regions of high utility even if J is quite high. We also performed the analyses for the case where we use the posterior sample of size 5000, but the results are not much different except for a slightly more uneven expected utility surface.

The length of the Markov chain has a rather small effect on the result, even though for high J the acceptance rates are very low. In order to make sure that our sample is informative enough, however, we prefer a sample of size 10^5 , which took the computer 41 seconds to obtain.

As would be expected, varying J has the most pronounced effects on the outcome. Thus, Figure 2 displays the kernel density estimates of the marginal sample of d for $J = 50, 100$, and 200 when the Markov chain has length 10^5 . The corresponding acceptance rates and IATs are given in Table 2.

The two-stage approach can easily be extended to designs with more than one dimension. The

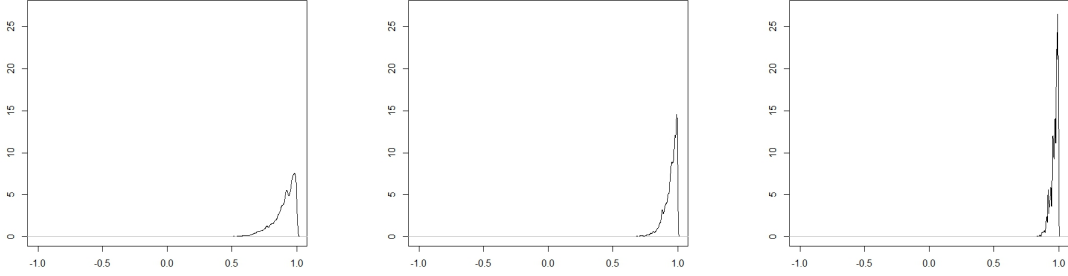


Figure 2: Kernel density estimates of marginal distribution of d for a two-stage MCMC sampler; one-dimensional design; sample length 10^5 ; $J = 50$ (left), $J = 100$ (center), and $J = 200$ (right).

Table 2: Acceptance rates and IATs of a two-stage MCMC sampler for a one-dimensional design with sample length 10^5 using different values of J .

J	Accept. rate	IAT
50	0.0628	38.96
100	0.0260	108.13
200	0.0078	524.14

setting is altered in the following way to create a two-dimensional design problem:

$\mathbf{R} = \mathbf{f}(-1)\mathbf{f}(-1)^T + \mathbf{f}(0)\mathbf{f}(0)^T$ and $\boldsymbol{\xi} = (-1, 1)^T$ so that the optimal design is either $\mathbf{d} = (0, 1)^T$ or $\mathbf{d} = (1, 0)^T$ (two maxima). In the multidimensional case, the uniform ABC kernel function is defined as

$$p_\epsilon(\mathbf{y}, \mathbf{x}) \propto \mathbb{I}_{\|\mathbf{y} - \mathbf{x}\|/\sqrt{n} < \epsilon}(\mathbf{x}).$$

For the values of ϵ we choose the same sequence as in the one-dimensional case. We obtain posterior samples of size 20000 (13.39 min.), 10000 (3.35 min.), and 5000 (0.85 min.) for the posterior $p(\boldsymbol{\theta}|\mathbf{y} = (-1, 160)^T, \boldsymbol{\xi} = (-1, 1)^T)$. The differences between the three samples are only marginal, all of them being very good approximations to the true posterior distribution. We use the sample of size 20000 for the second stage.

Figure 3 shows the outputs of the samplers for $J = 50$, $J = 100$, and $J = 200$ and for the

Markov chains of length 10^4 and 10^5 . The longest runtime for the second stage is 1.19 minutes (sample length 10^5 and $J = 200$), which is perfectly acceptable. One can see that for $J = 200$ we get very good results. The sampled design points are very close to the optimal values for almost all iterations of the sampler.

Table 3: Acceptance rates and IATs of a two-stage MCMC sampler for a two-dimensional design using different values of J .

J	Markov chain length 10^4			Markov chain length 10^5		
	Accept. rate	IAT d_1	IAT d_2	Accept. rate	IAT d_1	IAT d_2
50	0.1510	30.45	28.64	0.1411	42.29	43.79
100	0.0352	88.55	149.71	0.0240	1109.95	1134.46
200	0.0090	427.40	485.51	0.0026	2543.14	2457.598

Acknowledgements

This work was partially supported by AANR-2011-IS01-001-01 DESIRE and FWF I 833-N18.

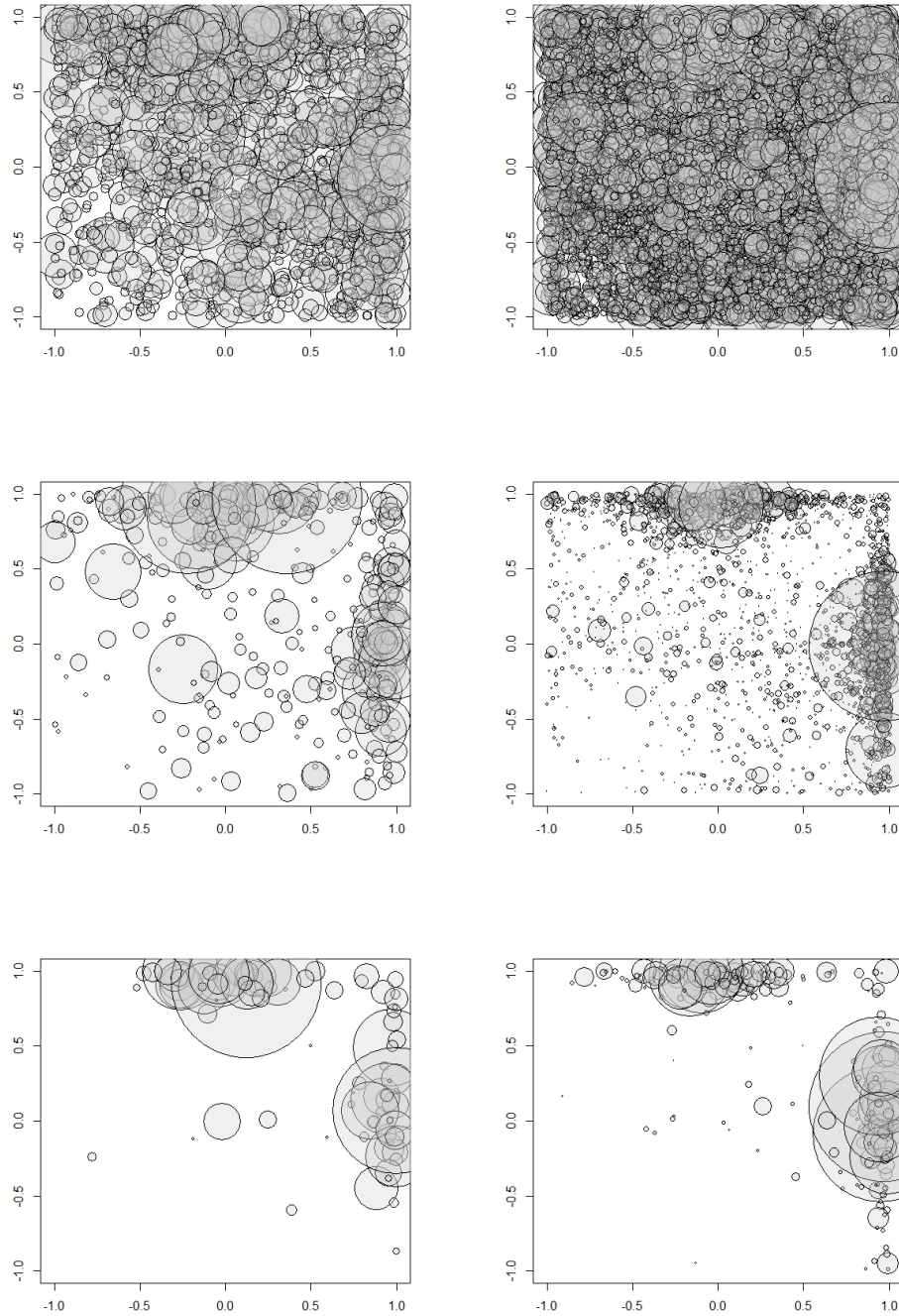


Figure 3: Bubble plot of samples of marginal distribution of \mathbf{d} (x-axis: d_1 ; y-axis: d_2) for a two-stage MCMC sampler; two-dimensional design; sample lengths 10^4 (left) and 10^5 (right); $J = 50$ (top), 100 (center), and 200 (bottom); circle areas proportional to the sample frequency of points (relative to the point with the highest frequency in the respective sample) .

Bibliography

- B. Amzal, F. Y. Bois, E. Parent, and C. P. Robert. Bayesian-optimal design via interacting particle systems. *Journal of the American Statistical Association*, 101(474):773–785, 2006. doi: 10.1198/016214505000001159.
- A. C. Atkinson, A. N. Donev, and R. D. Tobias. *Optimum Experimental Designs, with SAS*. Oxford University Press, New York, 2007.
- M. A. Beaumont, W. Zhang, and D. J. Balding. Approximate Bayesian computation in population genetics. *Genetics*, 162(4):2025–2035, 2002. URL <http://www.genetics.org/content/162/4/2025.full.pdf+html>.
- M. A. Beaumont, J.-M. Cornuet, J.-M. Marin, and C. P. Robert. Adaptive approximate Bayesian computation. *Biometrika*, 96:983–990, 2009. doi: 10.1093/biomet/asp052. URL <http://dx.doi.org/10.1093/biomet/asp052>.
- F. V. Bonassi, L. You, and M. West. Bayesian learning from marginal data in bionetwork models. *Statistical Applications in Genetics and Molecular Biology*, 10, 2011. doi: 10.2202/1544-6115.1684. URL <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3215428/>.
- P. Bortot, S. G. Coles, and S. A. Sisson. Inference for stereological extremes. *Journal of the American Statistical Association*, 102(477):84–92, 2007. doi: 10.1198/016214506000000988.
- O. Cappé, A. Guillin, J.-M. Marin, and C. P. Robert. Population Monte Carlo. *Journal Of Computational and Graphical Statistics*, 13(4):907–929, 2004. doi: 10.1198/106186004X12803. URL <http://dx.doi.org/10.1198/106186004X12803>.

- K. Chaloner and I. Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10(3):273–304, 1995. doi: 10.1214/ss/1177009939. URL <http://dx.doi.org/10.1214/ss/1177009939>.
- P. Del Moral, A. Doucet, and A. Jasra. Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society, Series B*, 68:411–436, 2006. doi: 10.1111/j.1467-9868.2006.00553.x. URL <http://dx.doi.org/10.1111/j.1467-9868.2006.00553.x>.
- P. Del Moral, A. Doucet, and A. Jasra. An adaptive sequential Monte Carlo method for approximate Bayesian computation. *Statistics and Computing*, 22:1009–1020, 2012. doi: 10.1007/s11222-011-9271-y. URL <http://dx.doi.org/10.1007/s11222-011-9271-y>.
- C. C. Drovandi and A. N. Pettitt. Likelihood-free Bayesian estimation of multivariate quantile distributions. *Computational Statistics & Data Analysis*, 55(9):2541–2556, 2011. doi: 10.1016/j.csda.2011.03.019. URL <http://dx.doi.org/10.1016/j.csda.2011.03.019>.
- C. C. Drovandi and A. N. Pettitt. Bayesian experimental design for models with intractable likelihoods. Technical report, Queensland University of Technology, Brisbane, Australia, 2012. URL <http://eprints.qut.edu.au/53924/>.
- C. C. Drovandi, J. M. McGree, and A. N. Pettitt. Sequential Monte Carlo for Bayesian sequentially designed experiments for discrete data. *Computational Statistics & Data Analysis*, 57:320–335, 2013. doi: doi:10.1016/j.csda.2012.05.014. URL <http://dx.doi.org/10.1016/j.csda.2012.05.014>.
- R. J. Erhardt and R. L. Smith. Approximate Bayesian computing for spatial extremes. *Computational Statistics & Data Analysis*, 56(6):1468–1481, 2012. doi: 10.1016/j.csda.2011.12.003.
- Y. Fan, D. J. Nott, and S. A. Sisson. Approximate Bayesian computation via regression density estimation, 2012. URL <http://arxiv.org/abs/1212.1479>.
- C. J. Geyer. Practical Markov chain Monte Carlo. *Statistical Science*, 7:473–483, 1992.
- M. Hainy, W. G. Müller, and H. Wagner. Simulation-based D_B -optimal designs: conception and implementation issues. In A. Komárek and S. Nagy, editors, *Proceedings of the 27th International Workshop on Statistical Modelling*, pages 537–542, 2012.

- M. Hainy, W. G. Müller, and H. P. Wynn. Approximate Bayesian computation design (ABCD), an introduction. In D. Ucinsky, A. C. Atkinson, and M. Patan, editors, *mODa 10 - Advances in Model-Oriented Design and Analysis*, Contributions to Statistics, pages 135–143. Springer International Publishing, 2013. doi: 10.1007/978-3-319-00218-7_16. URL http://dx.doi.org/10.1007/978-3-319-00218-7_16.
- X. Huan and Y. M. Marzouk. Simulation-based optimal Bayesian experimental design for nonlinear systems. Technical report, Massachusetts Institute of Technology, Cambridge, 2011. URL <http://arxiv.org/abs/1108.4146>.
- F. Jabot and J. Chave. Inferring the parameters of the neutral theory of biodiversity using phylogenetic information, and implications for tropical forests. *Ecology Letters*, 12(3):239–248, 2009. doi: 10.1111/j.1461-0248.2008.01280.x.
- E. Jacquier, M. Johannes, and N. G. Polson. Maximum expected utility via MCMC. 2011. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.118.393>.
- S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.
- H. Kück, N. de Freitas, and A. Doucet. SMC samplers for Bayesian optimal nonlinear design. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.72.6193>.
- J. S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer, New York, 2001.
- P. Marjoram, J. Molitor, V. Plagnol, and S. Tavaré. Markov chain Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences of the United States of America*, 100(26):15324–15328, 2003. doi: 10.1073/pnas.0306899100. URL <http://dx.doi.org/10.1073/pnas.0306899100>.
- P. Müller. Simulation based optimal design. In J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith, editors, *Bayesian Statistics 6*, pages 459–474. Oxford University Press, New York, 1999. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.131.3004>.

- P. Müller, B. Sansó, and M. De Iorio. Optimal Bayesian design by inhomogeneous Markov chain simulation. *Journal of the American Statistical Association*, 99(467):788–798, 2004. doi: 10.1198/016214504000001123.
- C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer, New York, 2 edition, 2004.
- P. Sebastiani and H. P. Wynn. Bayesian experimental design and Shannon information. In *Proceedings of the Section on Bayesian Statistical Science*, pages 176–181, 1997.
- S. A. Sisson and Y. Fan. Likelihood-free Markov chain Monte Carlo. In S. P. Brooks, A. Gelman, G. Jones, and X.-L. Meng, editors, *Handbook of Markov Chain Monte Carlo*, Handbooks of Modern Statistical Methods, pages 319–341. Chapman and Hall/CRC Press, Boca Raton, 2011.
- S. A. Sisson, Y. Fan, and M. M. Tanaka. Sequential Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences of the United States of America*, 104(6):1760–1765, 2007. doi: 10.1073/pnas.0607208104. URL <http://www.pnas.org/cgi/doi/10.1073/pnas.0607208104>.
- A. Solonen, H. Haario, and M. Laine. Simulation-based optimal design using a response variance criterion. *Journal Of Computational and Graphical Statistics*, 21(1):234–252, 2012. doi: 10.1198/jcgs.2011.10070.
- J. C. Spall. Implementation of the simultaneous perturbation algorithm for stochastic optimization. *IEEE Transactions on Aerospace and Electronic Systems*, 34(3):817–823, 1998a. doi: 10.1109/7.705889.
- J. C. Spall. An overview of the simultaneous perturbation method for efficient optimization. *Johns Hopkins APL Technical Digest*, 19(4):482–492, 1998b. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.19.3014>.
- L. Tierney. Markov chains for exploring posterior distributions. *The Annals of Statistics*, 22(4):1701–1728, 1994. doi: 10.1214/aos/1176325750. URL <http://dx.doi.org/10.1214/aos/1176325750>.
- P. J. M. van Laarhoven and E. H. L. Aarts. *Simulated Annealing: Theory and Applications*. Mathematics and Its Applications. Kluwer, Dordrecht, 1992.

- A. A. Zhigljavsky and A. Žilinskas. *Stochastic Global Optimization*, volume 9 of *Springer Optimization and Its Applications*. Springer, Berlin, 2008.