INSTITUT NATIONAL DE LA STATISTIQUE ET DES ETUDES ECONOMIQUES
Série des Documents de Travail du CREST
(Centre de Recherche en Economie et Statistique)

n° 2010-24

Adaptive Monte Carlo on Multivariate Binary Sampling Spaces

N. CHOPIN[1]
C. SCHÄFER[2]

[1] CREST and ENSAE. Email : nicolas.chopin@ensae.fr
[2] CREST and Université Paris-Dauphine. Email : christian.schafer@ensae.fr

Adaptive Monte Carlo on multivariate binary sampling spaces

Christian Schäfer*, Nicolas Chopin†

A Monte Carlo algorithm is said to be adaptive if it can adjust automatically its current proposal distribution, using past simulations. The choice of the parametric family that defines the set of proposal distributions is critical for a good performance. We treat the problem of constructing such parametric families for adaptive sampling on multivariate binary spaces.

A practical motivation for this problem is variable selection in a linear regression context, where we need to either find the best model, with respect to some criterion, or to sample from a Bayesian posterior distribution on the model space. In terms of adaptive algorithms, we focus on the Cross-Entropy (CE) method for optimisation, and the Sequential Monte Carlo (SMC) methods for sampling.

Raw versions of both SMC and CE algorithms are easily implemented using binary vectors with independent components. However, for high-dimensional model choice problems, these straightforward proposals do not yields satisfactory results. The key to advanced adaptive algorithms are binary parametric families which take at least the linear dependencies between components into account.

We review suitable multivariate binary models and make them work in the context of SMC and CE. Extensive computational studies on real life data with a hundred covariates seem to prove the necessity of more advanced binary families, to make adaptive Monte Carlo procedures efficient. Besides, our numerical results encourage the use of SMC and CE methods as alternatives to techniques based on Markov chain exploration.

Keywords: Adaptive Monte Carlo; Multivariate binary data; Sequential Monte Carlo; Cross-Entropy method; Linear regression; Variable selection.

1 Introduction

1.1 Adaptive Monte Carlo techniques

A Monte Carlo algorithm is said to be adaptive when it has the ability to adjust, sequentially and automatically, its sampling distribution to the problem at hand. Important classes of adaptive Monte Carlo algorithms include: Adaptive Importance Sampling (e.g. Cappé et al., 2008); Adaptive Markov chain Monte Carlo (e.g. Andrieu and Thoms, 2008); Sequential

*CREST and Université Paris Dauphine, christian.schafer@ensae.fr
†CREST and ENSAE, nicolas.chopin@ensae.fr

Monte Carlo (Del Moral et al., 2006); the Cross-Entropy method (Rubinstein and Kroese, 2004), among others.

Specifically, an adaptive algorithm relies on a parametric family of sampling distributions which should have the following three properties: (a) the parametric family is sufficiently rich, so as to guarantee a reasonable performance for the chosen algorithm when fully adapted; (b) we can quickly sample from each member of the parametric family; (c) we can calibrate the parameters of the family using past simulation. For problems in continuous sampling spaces, the most typical example is the multivariate normal distribution, which clearly fulfils (b) and (c), and complies with (a) in many practical problems.

1.2 Adaptive Monte Carlo on binary spaces

The objective of this work is to review and suggest parametric families for adaptive Monte Carlo applications in a binary sampling space $\boldsymbol{\Gamma} = \{0,1\}^d$, where d is too large to allow for exhaustive enumeration of $\boldsymbol{\Gamma}$. The discrete problem is more difficult than its continuous analogue, since there is no multivariate binary family which we can easily parameterise by first and second order moments like the multivariate normal. We shall see in an upcoming review of binary models that hardly any binary model complies with all (a), (b) and (c) for multivariate binary sampling problems in high dimensions.

Our motivating application is variable selection in a Gaussian linear regression model. In this context, a binary vector $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_d)$ encodes whether each of d possible covariates should be included or not in the regression model. In a Bayesian framework, and for a proper choice of prior for the regression coefficient, one can compute the marginal posterior distribution $\pi(\boldsymbol{\gamma}) = \pi(\boldsymbol{\gamma} \,|\, \boldsymbol{y})$.

Depending on the context, one may wish to either sample from $\pi(\boldsymbol{\gamma})$, so as to approximate quantities such as the marginal probability of inclusion of each variable, or to find the mode of $\pi(\boldsymbol{\gamma})$, that is the model with highest posterior probability. In a Frequentist framework, one wishes to discover the optimal $\boldsymbol{\gamma}$ with respect to some standard criterion such as BIC. In short, one is interested in either sampling from a probability mass function, or maximising the function, both defined on a set of binary vectors.

1.3 Global versus local methods

In terms of classes of adaptive algorithms, we focus on global methods, namely sequential Monte Carlo for the sampling problem, and the Cross-Entropy method for the optimisation problem. The reason is two-fold.

Firstly, there is growing evidence that global methods, which track a population of 'particles', initially well spread over the sampling space $\boldsymbol{\Gamma}$, are often more robust than local methods based on MCMC, as the latter are more prone to get trapped in the neighbourhood of local modes. We illustrate this point in our simulations.

Secondly, global methods have the property to be easily parallelisable. Parallel implementations of Monte Carlo algorithms have gained a tremendous interest in the very recent years (Lee et al., 2009; Suchard et al., 2010), due to the increasing availability of multi-core (central or graphical) processing units in standard computers.

Anyhow, we expect that the parametric families we review in this paper should also be useful in the context of other classes of adaptive Monte Carlo algorithms, such as Adaptive Markov Chain Monte Carlo.

1.4 Plan, notations

The paper is organised as follows. In Section 2, we briefly review the basics of our motivating application, namely variable selection in linear regression models. In Section 3, we discuss several approaches for constructing parametric families on multivariate binary spaces, in light of the aforementioned criteria. This is the core of our work.

In Section 4, we explain how to incorporate these parametric families in a Sequential Monte Carlo algorithm. We compare our approach to the standard Markov chain Monte Carlo on the basis of a real-life data variable selection model with 105 covariates. In the following Section 5, we do the same for the Cross-Entropy optimisation algorithm and compare the results to standard Simulated Annealing. Section 6 concludes and gives a brief outlook.

Our generic notation for, respectively, scalars, vectors, and matrices is: x, $\boldsymbol{x}$ (bold face), and $\mathbf{A}$ (bold capital). For a vector $\boldsymbol{x}$, the sub-vector indexed by $I \subset \mathbb{N}$ is denoted by $\boldsymbol{x}_I$. For an index set $I = \{i, \dots, j\}$, we write $\boldsymbol{x}_{i:j}$ instead. For a matrix $\mathbf{A}$, the determinant is $|\mathbf{A}|$, the trace is $\operatorname{tr}[\mathbf{A}]$, the operator $\operatorname{diag}[\cdot]$ transforms either vectors into diagonal matrices or vice versa.

2 Variable selection: A binary sampling problem

In this section, we briefly introduce the details of our motivating application. The complexity of variable selection problems in practice often outgrows the increase in computational power, such that adopting adaptive Monte Carlo methods to binary spaces is a relevant topic.

2.1 Variable selection in linear regression models

The standard linear normal model postulates that the relationship between the observed explained variable $\boldsymbol{y} \in \mathbb{R}^m$ and the observations $\mathbf{Z} = [\boldsymbol{z}_1, \dots, \boldsymbol{z}_d] \in \mathbb{R}^{m,d}$ is given by

$$\boldsymbol{y} \,|\, \boldsymbol{\beta}, \boldsymbol{\gamma}, \sigma^2, \mathbf{Z} \sim \mathcal{N}\left(\mathbf{Z} \operatorname{diag}[\boldsymbol{\gamma}]\, \boldsymbol{\beta}, \sigma^2 \mathbf{I}\right), \tag{1}$$

where the first column $\mathbf{Z}_{\cdot,1}$ is assumed to be a constant. The parameter $\boldsymbol{\gamma} \in \boldsymbol{\Gamma} = \{0,1\}^d$ determines which covariates are included in or dropped from the linear regression model. Hence, in total, we can construct 2^d different linear normal models from the data.

We assign a prior distribution $\pi(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\gamma} \,|\, \mathbf{Z})$ to the parameters, treating them as random variables. From the posterior distribution $\pi(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\gamma} \,|\, \boldsymbol{y}, \mathbf{Z}) \propto \pi(\boldsymbol{y} \,|\, \boldsymbol{\beta}, \sigma^2, \boldsymbol{\gamma}, \mathbf{Z})\, \pi(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\gamma} \,|\, \mathbf{Z})$ we may compute the posterior probability of each model

$$\pi(\boldsymbol{\gamma} \,|\, \boldsymbol{y}, \mathbf{Z}) = \int \pi(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\gamma} \,|\, \boldsymbol{y}, \mathbf{Z})\, d(\boldsymbol{\beta}, \sigma^2) \tag{2}$$

by integrating out the parameters $\boldsymbol{\beta}$ and σ^2.

2.2 Hierarchical Baysian model

In a purely Bayesian context, we obtain an explicit solution of the integral in (2), by decomposing the full posterior and choosing conjugate hierarchical priors (George and McCulloch, 1997); that is a normal $\pi(\boldsymbol{\beta} \,|\, \sigma, \boldsymbol{\gamma}, \mathbf{Z})$ and an inverse-gamma $\pi(\sigma^2 \,|\, \boldsymbol{\gamma}, \mathbf{Z})$. We refer to (George and McCulloch, 1997), and the citations therein, for more details on a meaningful choice of the prior parameters.

For our purpose, we let $\pi(\boldsymbol{\beta} \,|\, \sigma, \boldsymbol{\gamma}, \mathbf{Z}) = \mathcal{N}\left(\mathbf{0}, \sigma^2 v^2 \mathrm{diag}\,[\boldsymbol{\gamma}]\right)$, $\pi(\sigma^2 \,|\, \boldsymbol{\gamma}, \mathbf{Z}) = \mathcal{I}\Gamma(w, w)$, and set a uniform $\pi(\boldsymbol{\gamma} \,|\, \mathbf{Z}) \equiv 2^{-d}$. Then, let $\boldsymbol{b}_{\boldsymbol{\gamma}} = \mathbf{Z}_{\boldsymbol{\gamma}}^\intercal \boldsymbol{y}$ and $\mathbf{C}_{\boldsymbol{\gamma}} \mathbf{C}_{\boldsymbol{\gamma}}^\intercal = \mathbf{Z}_{\boldsymbol{\gamma}}^\intercal \mathbf{Z}_{\boldsymbol{\gamma}}$ a Cholesky decomposition, where $\mathbf{Z}_{\boldsymbol{\gamma}}$ is $\mathbf{Z}\,\mathrm{diag}\,[\boldsymbol{\gamma}]$ without the zero columns, and $s_{\boldsymbol{\gamma}}^2 = \boldsymbol{y}^\intercal \boldsymbol{y} - (\mathbf{C}_{\boldsymbol{\gamma}}^{-1} \boldsymbol{b}_{\boldsymbol{\gamma}})^\intercal (\mathbf{C}_{\boldsymbol{\gamma}}^{-1} \boldsymbol{b}_{\boldsymbol{\gamma}})$. The log-posterior probability is given by

$$\begin{aligned} \log \pi(\boldsymbol{\gamma} \,|\, \boldsymbol{y}, \mathbf{Z}) &= c - \frac{1}{2}\left[\log(\left|\mathbf{C}_{\boldsymbol{\gamma}} \mathbf{C}_{\boldsymbol{\gamma}}^\intercal + v^{-1} \mathrm{diag}\,[\boldsymbol{\gamma}]\right|) - |\boldsymbol{\gamma}| \log(v) - (m + w) \log(w + s_{\boldsymbol{\gamma}}^2)\right] \\ &\approx c - \log\left(\textstyle\prod_i^{|\boldsymbol{\gamma}|} c_{ii}^{[\boldsymbol{\gamma}]}\right) - \frac{|\boldsymbol{\gamma}|}{2} \log(v) - m \log\left(s_{\boldsymbol{\gamma}}\right). \end{aligned} \tag{3}$$

In our numerical examples in Section 4.5, we let $v \approx 10^2$, and use the slight simplification (3), which removes the computational burden of evaluating the determinant. For the second parameter, we take $w \approx 10^{-1}$, which makes it almost insignificant when the sample size m is large.

2.3 Baysian Information Criterion

In a Frequentist framework, one instead tries to choose a model which minimises some criterion. A popular criterion is BIC (Schwarz, 1978, Bayesian Information Criterion), which basically is a second degree Laplace approximation of (2):

$$\log \pi(\boldsymbol{\gamma} \,|\, \boldsymbol{y}, \mathbf{Z}) \approx \mathrm{BIC} = c - \frac{|\boldsymbol{\gamma}|}{2} \log(m) - m \log(s_{\boldsymbol{\gamma}}), \tag{4}$$

where $s_{\boldsymbol{\gamma}}$ is the maximum likelihood estimator of σ^2 for the model $\boldsymbol{\gamma}$. Note that this expression is rather similar to the simplified conjugate Bayesian formula derived in (3).

2.4 Data

For our numerical experiments in Sections 4 and 5, we use the Boston Housing dataset, originally treated by Harrison and Rubinfeld (1978), and build a linear regression model for the mean value of owner-occupied homes. We augment the 13 covariates by adding a constant column and then crossing all variables, thus building a model with 105 possible covariates. We use the hierarchical Bayesian approach, with prior distributions as explained in the above Section 2.2, to construct a posterior distribution $\pi(\boldsymbol{\gamma}) = \pi(\boldsymbol{\gamma} \,|\, \boldsymbol{y}, \mathbf{Z})$ on the set of possible models $\boldsymbol{\Gamma} = \{0, 1\}^{105}$.

We choose this particular problem to run our numerical experiments, because $\pi(\boldsymbol{\gamma})$ results to be a rather complex, multi-modal posterior distribution with considerable correlation between the components of $\boldsymbol{\gamma} \sim \pi(\boldsymbol{\gamma})$. In other words, the Boston Housing dataset yields a challenging integration and optimisation problem, while the problem size 105 still allows for efficient non-parallelised processing on a standard personal computer.

3 Binary distributions

In this section, we review models $q(\boldsymbol{\gamma} \,|\, \theta)$ for multivariate binary data, which can serve as parametric families in adaptive Monte Carlo algorithms on binary spaces $\boldsymbol{\Gamma} = \{0, 1\}^d$, as described in the introductory section 1. We discuss how to incorporate these binary models into our example applications, that is Sequential Monte Carlo for integration and Cross-Entropy for optimisation, at the end of sections 4 and 5, respectively.

3.1 Preliminaries

We look for binary models $q(\boldsymbol{\gamma} \,|\, \theta)$ which allow to generate independent samples $\boldsymbol{x} \sim q(\boldsymbol{x} \,|\, \theta)$ and to estimate the parameter θ given a sample $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) \sim \pi(\boldsymbol{\gamma})$. We start with the simplest case: For $\theta = \boldsymbol{p} \in (0,1)^d$ we define the *independent* binary model as

$$q(\boldsymbol{\gamma} \,|\, \boldsymbol{p}) \stackrel{\text{def.}}{=} \prod_{i \in D} b(\gamma_i \,|\, p_i), \quad b(\gamma \,|\, p) \stackrel{\text{def.}}{=} p^{\gamma}(1-p)^{1-\gamma}. \tag{5}$$

where $D = \{1, \ldots, d\}$. Sampling and parameter estimation is fast and easy for this model, but it does not generate suitable proposals in complicated, high-dimensional applications of adaptive Monte Carlo. We shall therefore review some richer binary models and compare their advantages and drawbacks.

3.1.1 Remarks on binary data

In general, multivariate binary data is characterised by either 2^d probabilities or 2^d cross-moments

$$m_I \stackrel{\text{def.}}{=} \mathbb{E}_{\pi}\left[\textstyle\prod_{i \in I} \gamma_i\right] = \sum_{\boldsymbol{\gamma} \in \boldsymbol{\Gamma}} \pi(\boldsymbol{\gamma}) \prod_{i \in I} \gamma_i = \sum_{\boldsymbol{\gamma} \in \{\boldsymbol{x} \in \boldsymbol{\Gamma}, \boldsymbol{x}_I = \mathbf{1}\}} \pi(\boldsymbol{\gamma}) = \mathbb{P}\left(\boldsymbol{\gamma}_I = \mathbf{1}\right), \tag{6}$$

where $I \subseteq D$ is a set of indices. The only constraints on multivariate binary data are

$$\max\left\{\textstyle\sum_{i \in I} m_i - |I| + 1, 0\right\} \leq m_I \leq \min\left\{m_K \,, K \subseteq I\right\}, \tag{7}$$

where the upper bound is the monotonicity of the measure, and the lower bound follows from

$$|I| - 1 = \textstyle\sum_{\boldsymbol{\gamma} \in \boldsymbol{\Gamma}} (|I| - 1)\pi(\boldsymbol{\gamma}) \geq \sum_{\boldsymbol{\gamma} \in \boldsymbol{\Gamma}} \left(\sum_{k \in I} \gamma_i - \prod_{i \in I} \gamma_i\right) \pi(\boldsymbol{\gamma}) = \sum_{i \in I} m_i - m_I. \tag{8}$$

In fact, m_I is a $|I|$-dimensional copula with respect to the moments m_i for $i \in I$, see Nelsen (2006, p.45), and (7) correspond to the Fréchet-Hoeffding bounds.

3.1.2 Model classification

Most of the literature on multivariate binary data stems from binary response models, multiway contingency tables and multivariate interaction theory. Cox (1972) gives an overview of binary models, and Streitberg (1990) develops an additive decomposition analysis, which includes the Bahadur (1961) expansion for binary data as a special case.

Following these authors, we classify the models according to their main structural characteristics. There are linear representation of $\pi(\boldsymbol{\gamma})$ and $\log \pi(\boldsymbol{\gamma})$ from which we derive *additive* and *multiplicative* models $q(\boldsymbol{\gamma} \,|\, \theta)$, respectively. We refer to $q(\boldsymbol{\gamma} \,|\, \theta)$ as a *latent variable* model if $\boldsymbol{\gamma} = g(\boldsymbol{v})$ for a latent random vector $\boldsymbol{v} \sim p(\boldsymbol{v} \,|\, \theta)$.

3.1.3 Model properties

Before we embark on the discussion of binary models, we specify more precisely the characteristics called for in the introductory section 1.1, that is (a) the richness of the model and (b) the possibility to sample from it. We do not further redescribe (c), since we only discuss models for which parameter estimation is somehow feasible.

(a) We consider models with at most $d(d+1)/2$ parameters which can take at least linear dependencies into account, meaning we should find θ such that

$$\mathbb{E}_{q\,|\,\theta}\left[\gamma_i\gamma_j\right] \approx \mathbb{E}_{\pi}\left[\gamma_i\gamma_j\right], \quad i,j \in D. \tag{9}$$

Unfortunately, most binary models $q(\boldsymbol{\gamma}\,|\,\theta)$ do not, for arbitrary binary data, admit a feasible θ such that (9) holds.

(b) We want to generate independent samples from $q(\boldsymbol{\gamma}\,|\,\theta)$. Unless we use a latent variable model, sampling is done via a decomposition into the conditionals

$$q(\boldsymbol{\gamma}\,|\,\theta) = q(\boldsymbol{\gamma}) = q(\boldsymbol{\gamma}_1)\prod_{i=2}^{d} q(\boldsymbol{\gamma}_{1:i})/q(\boldsymbol{\gamma}_{1:i-1}) = q(\boldsymbol{\gamma}_1)\prod_{i=2}^{d} q(\boldsymbol{\gamma}_i|\boldsymbol{\gamma}_{1:i-1}), \tag{10}$$

which requires explicit or recursive formulas for the marginal distributions $q(\boldsymbol{\gamma}_{1:k}\,|\,\theta)$. From such a decomposition, we can sample a random vector componentwise, conditioning on the part we already generated.

3.2 Additive models

For any binary probability mass function $\pi(\boldsymbol{\gamma})$, there are coefficients a_I for $I \subseteq D$ such that we can write it as $\sum_{I\subseteq D} a_I \prod_{i\in I}\gamma_i$, where the empty product is defined as 1. It seems natural to build a $d(d+1)/2$ parameter model by just truncating higher degree interactions terms.

As Streitberg (1999) points out, the main problem of any additive interaction model is the fact that a truncated model might not define a probability distribution because it is not non-negative. For the *quadratic* binary model of the kind

$$q(\boldsymbol{\gamma}\,|\,\mathbf{A}) \overset{\text{def.}}{=} c + \boldsymbol{\gamma}^{\intercal}\mathbf{A}\boldsymbol{\gamma}, \tag{11}$$

we have explicit and recursive formulas to compute the marginal probabilities. Further, we can determine a matrix $\mathbf{A}$ to match the second order moments of a sample by just solving a linear system of dimension $d(d+1)/2+1$.

However, in our simulations, the fitted $\mathbf{A}$ is indeed hardly ever positive definite, which renders the model rather useless, since it produces negative conditional probabilities when sampling from (10). As other authors interested in sampling binary vectors (Park et al., 1996; Emrich and Piedmonte, 1991) remark, additive representations like the Bahadur (1961) expansion are beautiful but, unfortunately, of rather limited practical value.

3.3 Multiplicative models

For any binary probability mass function $\pi(\boldsymbol{\gamma})$, there are coefficients a_I for $I \subseteq D$ such that we can write it as $\log\pi(\boldsymbol{\gamma}) = \sum_{I\subseteq D} a_I \prod_{i\in I}\gamma_i$, where the empty product is defined as 1. Hence, we can treat any binary distribution as a complete log-linear model. We easily identify the independent model described in (5) as the special case where $a_\emptyset = \log[\,\mathbb{P}\,(\boldsymbol{\gamma}=\mathbf{0})]$, $a_i = \log[\,\mathbb{P}\,(\gamma_i=1)\,/\mathbb{P}\,(\gamma_i=0)]$, and higher terms are zero.

3.3.1 Quadratic exponential binary model

We truncate the series to obtain a $d(d+1)/2$ parameter model, which is a binary analogue of the normal distribution. We refer to

$$q(\boldsymbol{\gamma}\,|\,\mathbf{A}) \overset{\text{def.}}{\propto} \exp(\boldsymbol{\gamma}^{\intercal}\mathbf{A}\boldsymbol{\gamma}). \tag{12}$$

as the *quadratic exponential* binary model. From log-linear models, albeit they define proper distributions, it is not possible to sample via (10), since their marginal distributions are not easy to compute.

Cox and Wermuth (1994) remedy this drawback by giving a recursion for approximate marginal distributions that are of the same form (12), again omitting higher order interaction terms. However, note that we propagate the sampling error when consecutively sampling from approximate conditional distributions, which is hazardous in high-dimensional problems.

3.3.2 Logistic binary model

Instead of fitting a log-linear model like (12) and computing approximations to the conditional distribution, we might rather fit separate regression models for each conditional distribution $\pi(\gamma_i \,|\, \boldsymbol{\gamma}_{1:i-1})$. We consider a log-linear model for the odds ratio $\mathbb{P}\left(\gamma_i = 1\right) / \mathbb{P}\left(\gamma_i = 0\right)$ of each component γ_i, conditional on the components $\boldsymbol{\gamma}_{1:i-1}$, which gives us a *logistic* binary model

$$q(\boldsymbol{\gamma} \,|\, \boldsymbol{\beta}) \stackrel{\text{def.}}{=} \prod_{i=1}^{d} b\left(\gamma_i \,|\, p(\boldsymbol{\gamma}_{0:i-1}^\intercal \boldsymbol{\beta}_i)\right), \quad p(\boldsymbol{\gamma}_{0:i-1}^\intercal \boldsymbol{\beta}_i) \stackrel{\text{def.}}{=} \left[1 + \exp(-\boldsymbol{\gamma}_{0:i-1}^\intercal \boldsymbol{\beta}_i)\right]^{-1}, \tag{13}$$

where $\boldsymbol{\beta} = \{\boldsymbol{\beta}_i \in \mathbb{R}^i,\ i \in D\}$ is a set of regression coefficients and $b(\gamma_i \,|\, p) = p^\gamma (1-p)^{1-\gamma}$ is defined in (5). The first component $\gamma_0 = 1$ is a constant added to the binary vector to enhance the logistic regressions.

From the logistic binary model, we can easily sample via (10). One might object that, given the data, there are $d!$ different logistic binary models and we arbitrarily pick one, while there should be a parameterisation which is optimal in a sense of nearness to the data. In high-dimensional applications, however, we did not observe that permutations of the order had an impact on our results.

3.3.3 Parameter estimation

The major drawback of all multiplicative models is the fact that there are no closed-form likelihood-maximisers and parameter estimation requires iterative, numerical fitting procedures. In the following, we give a brief review of the numerical methods for maximising the likelihood function $\ell(\boldsymbol{\beta})$ and point to some inherent pitfalls.

The natural way to solve the first order condition $\partial \ell / \partial \boldsymbol{\beta} = \mathbf{0}$ is a Newton-Raphson iteration

$$-\frac{\partial^2 \ell(\boldsymbol{\beta}_r)}{\partial \boldsymbol{\beta} \boldsymbol{\beta}^\intercal} (\boldsymbol{\beta}_{r+1} - \boldsymbol{\beta}_r) = \frac{\partial \ell(\boldsymbol{\beta}_r)}{\partial \boldsymbol{\beta}}, \quad r > 0, \tag{14}$$

starting at some $\boldsymbol{\beta}_0$. Green (1984) suggests to approximate the observed information matrix by its expected value conditional on $\boldsymbol{\beta}$, or Fisher information,

$$\mathcal{I}(\boldsymbol{\beta}) = \mathbb{E}\left[-\frac{\partial^2 \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta} \boldsymbol{\beta}^\intercal} \,\middle|\, \boldsymbol{\beta}\right] = -\left(\frac{\partial\, \boldsymbol{p}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}\right)^\intercal \mathbf{D} \left(\frac{\partial\, \boldsymbol{p}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}\right), \tag{15}$$

where $\mathbf{D}$ results to be a diagonal matrix. This simplification allows to compute $\boldsymbol{\beta}_{r+1}$ as the solution of *iteratively reweighted least squares* (IRLS) regressions. Naturally, other updating formulas for quasi-Newton iterations are also known to work well.

3.3.4 Complete separation

When maximising the standard likelihood function, we sometimes observe that the Newton iteration does not converge, due to data with *complete* or *quasi-complete* separation in the sample points. Albert and Anderson (1984) describe the separation problem, which leads to monotone likelihood functions with infinite maximisers. There are several ways to handle this issue:

(a) We just halt the algorithm after a fixed number of iterations and ignore the lack of convergence. However, such proceeding might cause numerical problems or lead to rather spurious approximations of the dependence structure.

(b) We remove the ith component of $\boldsymbol{\gamma}$ from the logistic binary model $q(\boldsymbol{\gamma} \,|\, \theta)$, if the parameter $\boldsymbol{\beta}_i$ fails to converge, and draw γ_i independently of $\boldsymbol{\gamma}_{-i}$. We also stick to this fallback option if the marginal probability $\pi_i(\gamma_i)$ is close to either boundary of $(0, 1)$. Such components are very likely to suffer from separation, and, anyhow, dependencies are negligibly low considering the bounds (7).

(c) We put a Jeffrey's prior on $\boldsymbol{\beta}$ as proposed by Firth (1993) to ensure that $\ell(\boldsymbol{\beta})$ is not monotonic. However, for the resulting penalized log-likelihood

$$\ell^*(\boldsymbol{\beta}) = \ell(\boldsymbol{\beta}) + \frac{1}{2} \operatorname{tr} \left[\mathcal{I}^{-1}(\boldsymbol{\beta}) \frac{\partial \mathcal{I}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \right] \tag{16}$$

computing the second derivatives is more involved, which calls for approaches like IRLS.

3.4 Latent variable models

Let $p(\boldsymbol{v} \,|\, \theta)$ be a distribution on $\mathcal{X}$ and $g : \mathcal{X} \to \boldsymbol{\Gamma}$ a mapping into the binary state space. We can sample from the *latent variable* model

$$q(\boldsymbol{\gamma} \,|\, \theta) = \int\limits_{g^{-1}(\boldsymbol{\gamma})} p(\boldsymbol{v} \,|\, \theta) \, d\boldsymbol{v} \tag{17}$$

by letting $\boldsymbol{\gamma} = g(\boldsymbol{v})$ for $\boldsymbol{v} \sim p(\boldsymbol{v} \,|\, \theta)$, but evaluating the probability mass function $q(\boldsymbol{\gamma} \,|\, \theta)$ might not be feasible. Ultimately, we just take a suitable parametric dependence structure and change the marginals to a binary vector. Hence, we could also discuss this approach in terms of copula methods, but, as Mikosch (2006) remarks in a critical paper, there is no scientific reason to insist on uniform marginals.

3.4.1 Normal binary model

Joe (1996) studies families with $d(d-1)/2$ bivariate dependence parameters and concludes that all non-normal families seem to either have a very limited dependence structure or unfavourable properties. Hence, the multivariate normal distribution appears to be not only the natural, but pretty much the only option for the latent distribution $p(\boldsymbol{v} \,|\, \theta)$.

Consequently, this choice has been discussed repeatedly in the literature (Emrich and Piedmonte, 1991; Leisch et al., 1998; Cox and Wermuth, 2002) with varying degrees of elaboration. For a vector $\boldsymbol{\mu} \in \mathbb{R}^d$ and a correlation matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{d,d}$, we define the *normal* binary distribution as

$$q(\boldsymbol{\gamma} \,|\, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \stackrel{\text{def.}}{=} \mathbb{P}\left(\mathbb{1}_{(0,\infty)}(v_i) = \gamma_i, \; i \in D \right), \quad \boldsymbol{v} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{18}$$

We can thus express the first and second order marginal probabilities of $\boldsymbol{\gamma} \sim q(\boldsymbol{\gamma} \,|\, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ through $\Phi_1(\mu_i) = \mathbb{P}(\gamma_i = 1)$ and $\Phi_2(\mu_i, \mu_j; \sigma_{ij}) = \mathbb{P}(\gamma_i = 1, \gamma_j = 1)$, where $\Phi_1(\cdot)$ denotes the cumulative distribution function (cdf) of the univariate and $\Phi_2(\,\cdot\,|\,\sigma_{ij})$ the cdf of the bivariate normal distribution with zero mean, unit variance and correlation σ_{ij}.

3.4.2 Parameter estimation

To construct a proposal distribution, we choose the parameter $\theta = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$ such that the marginal probabilities fit the sample moments $s_{ij} \stackrel{\text{def.}}{=} n^{-1} \sum_{k=1}^{n} x_{k,i} x_{k,j}$, that is

$$\Phi_1(\mu_i) = s_{ii}, \quad \Phi_2(\mu_i, \mu_j; \sigma_{ij}) = s_{ij}, \quad i \neq j,\ i, j \in D. \tag{19}$$

We easily solve the first part of (19) by setting $\mu_i = \Phi^{-1}(s_{ii})$ for all $i \in D$. The main task is the efficient computation of a feasible correlation matrix. We suggest the following Newton-Raphson method to solve $\Phi(\mu_i, \mu_j; \sigma_{ij}) - s_{ij} = 0$. Recall the standard result (see e.g. Johnson et al. (2002, p.255))

$$\frac{\partial \Phi_2(y_1, y_2; \sigma)}{\partial \sigma^2} = \phi_2(y_1, y_2; \sigma), \tag{20}$$

where $\phi_2(\cdot; \sigma)$ denotes the density of the bivariate normal distribution, which yields a straightforward Newton iteration

$$\alpha_{r+1} = \alpha_r - \frac{\Phi_2(\mu_i, \mu_j; \alpha_r) - s_{ij}}{\phi_2(\mu_i, \mu_j; \alpha_r)}, \quad r > 0, \quad 1 \leq i, j \leq d. \tag{21}$$

starting at some $\alpha_0 \in (-1, 1)$. We can efficiently evaluate the bivariate normal probabilities $\Phi_2(\mu_i, \mu_j; \alpha)$ using series approximations as suggested by Drezner and Wesolowsky (1990) or Divgi (1979). These approximations are critical when α_r comes very close to either boundary of $(-1, 1)$ and the Newton iteration might fail. However, equation (20) shows that $\Phi_2(y_1, y_2; \sigma)$ is monotonic in σ, and we can switch to bisectional search if necessary.

3.4.3 Infeasible parameters

A rather discouraging shortcoming of the normal model is the fact that the locally fitted correlation matrix $\boldsymbol{\Sigma}$ might not be positive definite for $d \geq 3$. This is due to the fact that an elliptical distribution like the normal can only attain the bounds (7) for $d < 3$ but not for higher dimensions.

There seem to be few suggestions on this topic in the literature. We present two ideas to obtain an approximate, but feasible parameter:

(a) We can replace $\boldsymbol{\Sigma}$ by $\boldsymbol{\Sigma}^* = (\boldsymbol{\Sigma} + |\lambda|\,\mathbf{I})/(1 + |\lambda|)$, where λ is the smallest eigenvalue of the dependency matrix $\boldsymbol{\Sigma}$. This approach evenly lowers the local correlations to a feasible level and is easy to implement on standard software. Alas, we make a considerable effort to estimate $d(d-1)/2$ dependency parameters, and in the end we might not obtain a lot more than an independent model.

(b) We can compute the correlation matrix $\boldsymbol{\Sigma}^*$ which minimizes the distance $\|\boldsymbol{\Sigma}^* - \boldsymbol{\Sigma}\|_F$, where $\|\mathbf{A}\|_F = \sqrt{\operatorname{tr}[\mathbf{A}\mathbf{A}^\intercal]}$. In other words, we construct the projection of $\boldsymbol{\Sigma}$ into the set of correlation matrices. Higham (2002) proposes an *alternating projections* algorithm to solve nearest-correlation matrix problems. Yet, if $\boldsymbol{\Sigma}$ is rather far from the set of correlation matrices, computing the projection is expensive and, in our experience, leads to troublesome distortions in the correlation structure.

3.4.4 Archimedean copula models

Genest and Neslehova (2007) discuss the potentials and pitfalls of applying copula theory, which is well developed for bivariate, continuous random variables, to multivariate discrete distribution. Yet, there have been earlier attempts to sample binary vectors via copulae: Lee (1993) describes how to construct an Archimedian copula, more precisely the *Frank* family, (see e.g. Nelsen (2006, p.119)), for sampling multivariate binary data.

Unfortunately, most results in copula theory do not easily extend to high dimensions. Indeed, we need to solve a non-linear equation for each component when generating a random vector from the Frank copula, and Lee acknowledges that this is only applicable for $d \leq 3$. For low-dimensional problems, however, we can just enumerate the solution space $\mathbf{\Gamma}$ and draw from an alias table (Walker, 1977), which somewhat renders the copula approach an interesting exercise without much practical value.

3.4.5 Multivariate reduction models

Several approaches to generating multivariate binary data are based on a representation of the components $\boldsymbol{\gamma}_i$ as functions of sums of independent variables, for all $i \in D$. These techniques are limited to certain patterns of non-negative correlation, and do, therefore, not yield suitable proposal distributions. We mention them for the sake of completeness.

Park et al. (1996) propose to let $\gamma_i = \delta_0(y_i)$ with $\boldsymbol{y} = \mathbf{A}\boldsymbol{z}$, where $\boldsymbol{z} \sim \mathcal{P}(\boldsymbol{z} \,|\, \boldsymbol{\lambda})$ is a vector of independent Poisson variables and $\mathbf{A}$ is a binary matrix. They describe a greedy algorithm that tries to find a feasible $\mathbf{A}$ and $\boldsymbol{\lambda}$. In the same spirit, Lunn and Davies (1998) propose to model the components $\boldsymbol{\gamma}_i$ as functions of independent Bernoulli draws, which is, although generalised by Oman and Zucker (2001), an even more limited approach. Yet, apparently, both methods work quite well on clustered data or auto-correlation structures.

3.5 Summary: Binary models in practice

Ultimately, the models suitable for adaptive Monte Carlo applications on high-dimensional binary sampling spaces are only the independent, the logistic and the normal model. In this section, we give some final remarks on these three models.

3.5.1 Review of the normal model

We argue that the logistic model dominates the normal model in terms of advantageous properties. For both the logistic model $q(\boldsymbol{\gamma} \,|\, \boldsymbol{\beta})$ and the latent normal model $q(\boldsymbol{\gamma} \,|\, \boldsymbol{\mu}, \mathbf{\Sigma})$ we have to iteratively estimate $d(d-1)/2$ dependence parameters. Hence, the models are equivalent concerning their complexity, albeit the latent normal model has two major shortcomings:

Firstly, the dependence structure of the model is more limited, and we seldom obtain a feasible parameter $\mathbf{\Sigma}$ when modelling high-dimensional binary data. Secondly, the probability mass function defined in (18) is an integral expression we cannot efficiently evaluate for $d \geq 3$, which makes the normal model impractical in importance sampling contexts. More precisely, we can employ the normal model in Cross-Entropy optimisation, but not in Sequential Monte Carlo algorithms. Eventually, we neglect the normal model in favour of the logistic model.

3.5.2 Modelling power versus estimation speed

We observe that models which incorporate arbitrary linear dependencies in multivariate binary data require numerically expensive fitting procedures in the parameter estimation process. On the other hand, we can easily parameterise the independent model by the sample mean. Plainly spoken, there is an enormous trade-off between the increase in modelling power and the parameter estimation speed. Why bother with linear dependencies at all?

We respond to this question with a toy example: Figure 1 shows is the posterior of a Bayesian variable selection problem, see section 2.2, where we have two variables plus two noisy copies. A parsimonious linear regression model should either include the original variable or its copy, but not both.

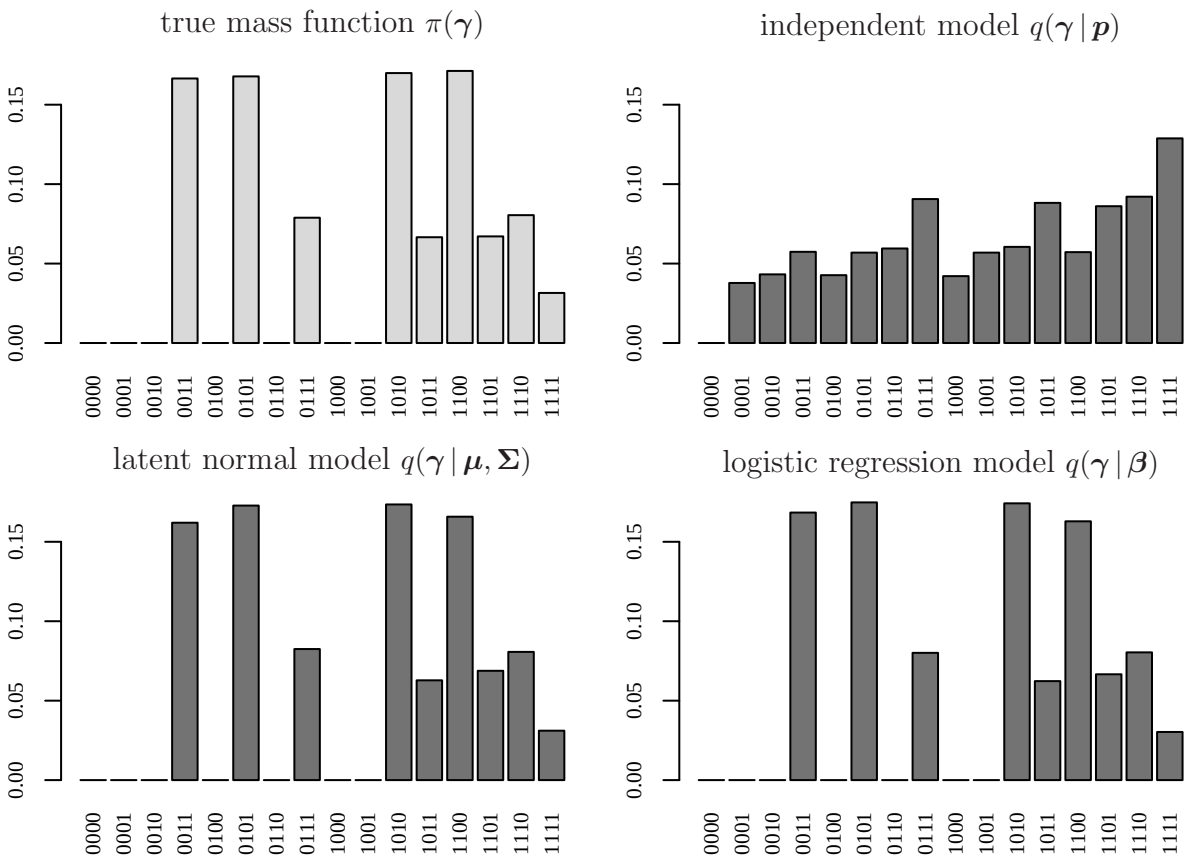


Figure 1: A toy example from variable selection.

4 Sequential Monte Carlo

In this section, we consider the problem of approximating integrals with respect to a probability mass function $\pi(\boldsymbol{\gamma})$ defined on $\boldsymbol{\Gamma} = \{0,1\}^d$. As motivated in the introduction, we focus on Sequential Monte Carlo (Del Moral et al., 2006). This general class of algorithms alternate importance sampling steps, resampling steps and Markov chain Monte Carlo steps, so as to approximate recursively a sequence of distributions, using a set of weighted 'particles' to represent the current distribution.

4.1 Sequence of distributions

The first ingredient of Sequential Monte Carlo is a sequence of distributions $(\pi_t)_{t=0}^{\tau}$, which ends up at the distribution of interest $\pi_\tau = \pi$. The intermediary distributions π_t are purely instrumental: the idea is to depart from a distribution π_0 with broad support, and which is easy to sample, and then to progress smoothly towards the distribution of interest π.

In our context, a natural strategy is the following geometric bridge (Gelman and Meng, 1998; Neal, 2001; Del Moral et al., 2006):

$$\pi_t(\boldsymbol{\gamma}) \overset{\text{def.}}{\propto} \pi_0(\boldsymbol{\gamma})^{1-\alpha_t}\pi(\boldsymbol{\gamma})^{\alpha_t}, \quad \alpha_t = t/\tau. \tag{22}$$

Of course, other choices of bridge distributions are possible, but (22) is most convenient, because we can easily compute $\log \pi_t(\boldsymbol{\gamma})$. Theoretically, we can depart from any distribution $\pi_0(\boldsymbol{\gamma})$ that we can sample from, but, in practice, the uniform distribution, $\pi_0(\boldsymbol{\gamma}) \equiv 2^{-d}$ turns out to yield the most reliable results.

4.2 Generic algorithm

Algorithm 1 describes a generic SMC algorithm. If we just do Step 1, the algorithm is equivalent to sequential importance sampling, where we sample the particles from π_0 at the first iteration, and just reweight the particles recursively, using the incremental weight function $u_t = \pi_t/\pi_{t-1}$.

This is inefficient if $\pi_\tau = \pi$ is far away from π_0, since we obtain a sample with extremely uneven weights. We remedy the weight degeneracy by introducing Step 2, a resample-move step (Gilks and Berzuini, 2001), where (a) particles are resampled according to their weights, and (b) resampled particles are replaced by iterates of a Markov kernel $\kappa_t(\boldsymbol{x}, \boldsymbol{y}) = p_t(\boldsymbol{y} \,|\, \boldsymbol{x})$ with invariant distribution π_t.

For the resampling step, several recipes exist, e.g. multinomial resampling (Gordon et al., 1993), residual resampling (Liu and Chen, 1998), systematic resampling (Kitagawa, 1998; Carpenter et al., 1999). We apply the latest in our simulations. The judicious choice of the kernel is the main concern of this paper and will be discussed separately in section 4.4.

4.3 Switching criterions

We still need to determine when to switch between the reweighting and the resample-move steps. We go to Step 2 when the weight degeneracy, measured by an efficient sample size criterion, see Kong et al. (1994),

$$(\text{ess})_t \overset{\text{def.}}{=} \|\boldsymbol{w}_t\|_1^2 / (n \cdot \|\boldsymbol{w}_t\|_2^2) < \eta, \quad \eta \in (1/n, 1), \tag{23}$$

exceeds a certain threshold η. In our simulations, we choose η to be about 2/3. We go back to Step 1, as soon as moving the particle system according to the transition kernel $\kappa_t(\boldsymbol{x}, \boldsymbol{y})$ does not increase the particle diversity, that is the number of distinct particles,

$$(\text{pd})_t \overset{\text{def.}}{=} \left|\{\boldsymbol{x}_k^{[t]},\, i \in N\}\right|, \quad N = \{1, \ldots, n\} \tag{24}$$

any longer. In our simulations, we return to Step 1 as soon as $|(\text{pd})_t - (\text{pd})_{t-1}| \leq 10^{-2}$. Note that (24) is a quality criterion for a particle system which has no analogue in continuous sampling spaces.

Algorithm 1 A generic SMC algorithm

0. Sample $\boldsymbol{x}_k^{[0]} \overset{\text{iid.}}{\sim} \pi_0$ and set $w_k^{[0]} = 1$ for all $k \in N = \{1, \ldots, n\}$. Let $t = 0$ and $s = 0$.

1. Until a degeneracy criterion like (23) is fulfilled or $t = \tau$, we update weights

$$w_k^{[t]} = w_k^{[t]} u_t(\boldsymbol{x}_k^{[s]}), \quad u_t \overset{\text{def.}}{=} \pi_t / \pi_{t-1}.$$

2.a We resample the particles, that is we construct a sample $\hat{\boldsymbol{x}}_1^{[s]}, \ldots, \hat{\boldsymbol{x}}_n^{[s]}$ which consists of $r_k^{[t]}$ replicates of the particle $\boldsymbol{x}_k^{[s]}$, for all $k \in N$, where $r_k^{[t]}$ is a nonnegative integer-valued random variable such that

$$\mathbb{E}\left[r_k^{[t]}\right] = n \cdot w_k^{[t]} / \left\|\boldsymbol{w}\right\|_1 .$$

After resampling, we set $w_k^{[t]} = 1$ for all $k \in N$.

2.b As long as we can increase a quality criterion like (24), we move the particles

$$\hat{\boldsymbol{x}}_k^{[s+1]} \sim \kappa_t\left(\hat{\boldsymbol{x}}_k^{[s]}, \boldsymbol{y}\right).$$

Otherwise, we set $\boldsymbol{x}_k^{[s]} = \hat{\boldsymbol{x}}_k^{[s]}$ for all $k \in N$ and go back to Step 1.

4.4 Choice of the Markov kernel

Here, we address the central problem of how to choose the transition kernels κ_t. We first introduce the adaptive, independent Metropolis-Hastings kernel and then argue why other, non-adaptive kernels known from Markov chain Monte Carlo do not work well.

4.4.1 Adaptive, independent Metropolis-Hastings kernels

We take κ_t as an *independent* Metropolis-Hastings kernel (e.g. Robert and Casella, 2004, chap. 7) with invariant distribution π_t,

$$\kappa_t(\boldsymbol{\xi}, \boldsymbol{\gamma}) \overset{\text{def.}}{=} \varrho_q^{[t]}(\boldsymbol{\xi}, \boldsymbol{\gamma}) + \delta_{\boldsymbol{\xi}}(\boldsymbol{\gamma}) \left[1 - \sum_{\boldsymbol{\gamma} \in \boldsymbol{\Gamma}} \varrho_q^{[t]}(\boldsymbol{\xi}, \boldsymbol{\gamma})\right], \; \varrho_q^{[t]}(\boldsymbol{\xi}, \boldsymbol{\gamma}) \overset{\text{def.}}{=} \left[1 \wedge \frac{\pi_t(\boldsymbol{\gamma}) q_t(\boldsymbol{\xi})}{\pi_t(\boldsymbol{\xi}) q_t(\boldsymbol{\gamma})}\right] q_t(\boldsymbol{\gamma}), \quad (25)$$

where q_t can be any distribution, as long as its support includes the support of π_t. We refer to the acceptance rate $\varrho_q^{[t]}(\boldsymbol{\xi}, \boldsymbol{\gamma})$ minus the probability $q_t(\boldsymbol{\xi})$ as the *mutation rate*, namely the probability that a draw from $\kappa_t(\boldsymbol{\xi}, \cdot)$ is not $\boldsymbol{\xi}$. The mutation rate is higher, the closer we choose q_t to π_t, and this is where the concept of adaptive Monte Carlo comes into play: we employ a suitable parametric family $q_t(\boldsymbol{\gamma}) = q(\boldsymbol{\gamma} \,|\, \theta_t)$, see section 3, where the parameter θ_t is fit to the weighted sample $(\boldsymbol{x}_s, \boldsymbol{w}_t)$.

The importance of criterion (a) we called for in the introductory section 1.1 is now evident: if q_t just poorly approximates π_t, the kernel κ_t is very unlikely to accept proposals from q_t, such that the particle diversity (24) is hard to augment and step 2.b of Algorithm 1 takes extremely long. Therefore, q_t should at least capture the correlation observed in the particle system; recall the toy example at the end of section 3 (Figure 1).

The comment made on modelling power and estimation speed in 3.5.2 translates into the context of Sequential Monte Carlo: there is a trade-off between high mutation rates and rapid

parameter estimation. In concrete terms, we can either quickly parameterise an independent model and suffer from low mutation rates, or fit a computationally intensive logistic model and enjoy higher mutation rates. In practice, we start with an independent model, since π_0 is an easy distribution, and switch to the logistic model as soon as the mutation rates drop considerably.

4.4.2 Non-adaptive alternative kernels

Do we need the adaptive Metropolis-Hastings kernel at all? We could just take κ_t as some kernel developed for Markov chain Monte Carlo on multivariate binary sampling spaces. For instance, a *symmetric* Metropolis-Hastings kernel is defined by

$$\kappa_t(\boldsymbol{\xi}, \boldsymbol{\gamma}) \stackrel{\text{def.}}{=} \varrho_k^{[t]}(\boldsymbol{\xi}, \boldsymbol{\gamma}) + \delta_{\boldsymbol{\xi}}(\boldsymbol{\gamma}) \left[1 - \sum_{\boldsymbol{\gamma} \in \boldsymbol{\Gamma}} \varrho_k^{[t]}(\boldsymbol{\xi}, \boldsymbol{\gamma}) \right], \; \varrho_k^{[t]}(\boldsymbol{\xi}, \boldsymbol{\gamma}) \stackrel{\text{def.}}{=} \left[1 \wedge \frac{\pi_t(\boldsymbol{\gamma})}{\pi_t(\boldsymbol{\xi})} \right] q(\boldsymbol{\gamma} \,|\, k)\, \omega(k), \quad (26)$$

where $q(\boldsymbol{\gamma} \,|\, k) = \delta_k(\|\boldsymbol{\xi} - \boldsymbol{\gamma}\|_1)\, k!(d-k)!/d!$ is the uniform distribution on the subset of vectors that differ by k components from $\boldsymbol{\xi}$, and $\omega(k)$ is an arbitrary distribution on D.

If we change one component at a time, letting $\omega(k) = \delta_1(k)$ as do Madigan et al. (1995), we rather often accept the proposals, but we are unlikely to augment the particle diversity, since we are inclined to just move forth and back between neighbouring modes.

In case we propose larger jumps in the state space, using, say, a truncated geometric distribution $\omega(k) \propto \mathbb{1}_D(k)\,(1-p)^{k-1}\,p$ with $p = 4/d$, the mutation rate almost vanishes in high dimensions, since we blindly propose arbitrary points in the sampling space.

We do not review the Gibbs kernel (George and McCulloch, 1993), since the symmetric Metropolis-Hastings kernel dominates it in terms of mutation rates, as discussed by George and McCulloch (1997). Finally, our numerical experiments confirm that indeed non-adaptive kernels do not, or very slowly, augment the particle diversity in a high-dimensional sampling space and are thus impractical for step 2.b of Algorithm 1.

4.5 Numerical example

In this section, we give some remarks on how to efficiently implement Algorithm 1 and compare the Sequential Monte Carlo approach to classic Markov chain Monte Carlo.

4.5.1 Remarks on the implementation

In Section 3.5.2 we comment the trade-off between modelling power and estimation speed. Here, we remark some basic ideas to reduce the computational burden of calibrating the proposal distributions.

(a) It is vital to work with an independent model as long as we can achieve reasonable mutation rates and switch to the logistic model only if necessary.

(b) Since calibrating the logistic model is computationally expensive, we reduce its dimension as far as reasonable. If the probability $\mathbb{P}(\gamma_i = 1)$ is close to either bound of the unit interval $(0, 1)$, we can justify to neglect interactions of γ_i with other components. For $\varepsilon > 0$, we define the set

$$R_\epsilon^{[t]} \stackrel{\text{def.}}{=} \{i \in D \,|\, s_i^{[t]} \in (\varepsilon, 1-\varepsilon)\}, \quad s_i^{[t]} \stackrel{\text{def.}}{=} n^{-1} \textstyle\sum_{k=1}^n x_{k,i}^{[t]} \quad (27)$$

For $R = R_\epsilon^{[t]}$, we draw $\boldsymbol{\gamma}_R \sim q(\boldsymbol{\gamma}_R \,|\, \boldsymbol{\beta}_R)$ from a suitable logistic model, while we generate the remaining components independently via $\gamma_i \sim b(\gamma \,|\, s_i^{[t]})$ for $i \in D \setminus R$.

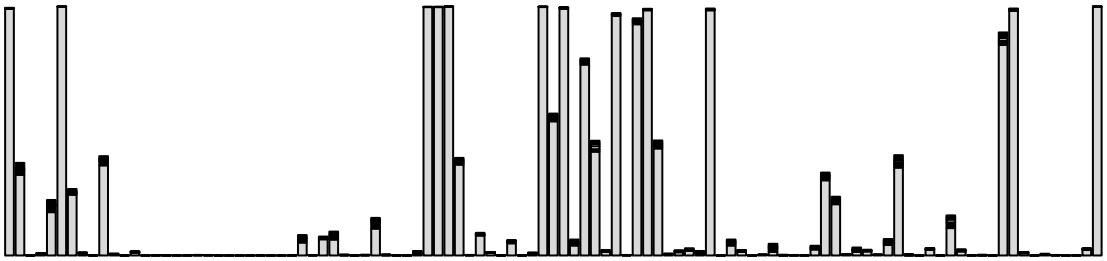
(c) Recall that $(\pi_t)_{t=0}^{\tau}$ is a smoothly evolving sequence of distributions, such that the adapted family $(q_t)_{t=0}^{\tau} = (q(\cdot \,|\, \theta_t))_{t=0}^{\tau}$ is characterised by a smooth parameter sequence $(\theta_t)_{t=0}^{\tau}$. Hence, it significantly improves the iterative parameter estimation, see Section 3.3.3, if we choose θ_t as starting value for the estimation of θ_{t+1}. Indeed, towards the end of Algorithm 1, we can fit the next logistic model $q(\boldsymbol{\gamma} \,|\, \boldsymbol{\beta}_{t+1})$ in less than four iterations on average, starting at $\boldsymbol{\beta}_0^{[t+1]} = \boldsymbol{\beta}_t$, while it takes about 13 iterations on average when starting at $\boldsymbol{\beta}_0^{[t+1]} = \mathbf{0}$.

4.5.2 Comparison to Markov chain Monte Carlo

Based on the variable selection problem described in Section 2.4, we compare Algorithm 1 to a classic Markov chain Monte Carlo approach driven by a symmetric Metropolis Hastings kernel. In Figure 2, we plot, for 200 runs, the estimates of the expected value $\overline{\boldsymbol{\gamma}} = \mathbb{E}_\pi[\boldsymbol{\gamma}]$ as grey bars. The white boxes on top contain 80% of the estimates, while the black boxes contain the 20% outliers. The horizontal bar in the white box indicates the median.

We allow the MCMC algorithm to perform more than twice as many evaluations of the posterior density compared to the SMC algorithm, in order to counterbalance the extra computational time we need to calibrate of the parametric family $q(\boldsymbol{\gamma} \,|\, \theta)$ in SMC. Regarding Figure 2, we observe that the variation of the SMC estimates is clearly smaller than the variation of the MCMC estimates.

Sequential Monte Carlo with $2.0 \cdot 10^4$ particles requiring about $1.1 \cdot 10^6$ evaluations of π.



Markov chain Monte Carlo with $2.5 \cdot 10^6$ evaluations of π.

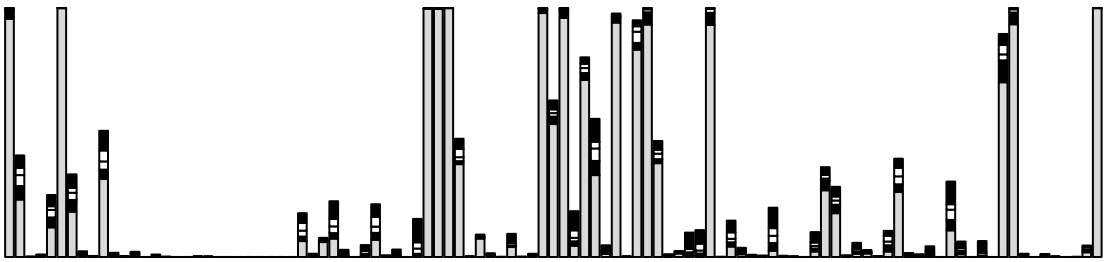


Figure 2: 200 runs of SMC and MCMC for an estimate of $\overline{\boldsymbol{\gamma}} = \mathbb{E}_\pi[\boldsymbol{\gamma}]$. For each component $\overline{\gamma}_i$, the white boxes contain 80% of the results, the bar indicates the median.

5 Cross-Entropy Optimisation

In this section, we consider the problem of finding the maximum of a function $\pi(\boldsymbol{\gamma})$ defined on $\boldsymbol{\gamma} \in \boldsymbol{\Gamma} = \{0,1\}^d$. There are plenty of stochastic search algorithms for discrete optimisation problems, but we focus on the Cross-Entropy (CE) method proposed by Rubinstein (1999) and further developed in Rubinstein and Kroese (2004, chap. 4), since it clearly is an adaptive Monte Carlo search technique and a global approach.

5.1 Sequence of distributions

Similar to the SMC algorithm discussed in Section 4, the main ingredient of CE optimisation is a sequence of distributions $(q_t)_{t=0}^{\tau}$ which ends up at the delta function $q_\tau = \delta_{\boldsymbol{\gamma}^*}$, where all mass is concentrated in a maximum $\boldsymbol{\gamma}^* = \operatorname{argmax}_{\boldsymbol{\gamma}} \pi(\boldsymbol{\gamma})$. We construct such a sequence letting $q_t(\boldsymbol{\gamma}) = q(\boldsymbol{\gamma} \,|\, \theta_t)$, for some suitable parametric family $q(\boldsymbol{\gamma} \,|\, \theta)$, which allows for the special cases of the uniform distribution $q(\boldsymbol{\gamma} \,|\, \theta_0) \equiv 2^{-d}$ and the delta distribution $q(\boldsymbol{\gamma} \,|\, \theta_\tau) = \delta_{\boldsymbol{\gamma}^*}$.

Unlike the SMC algorithm, we do not know the intermediary distributions and the final time τ at the beginning of the CE algorithm. Note that, analogously to the remark made for SMC, we can depart from any distribution $q(\boldsymbol{\gamma} \,|\, \theta_0)$ which includes $\boldsymbol{\gamma}^*$ in its support, but the uniform distribution seems to yield the most reliable results.

5.2 Generic algorithm

Algorithm 2 describes a generic CE algorithm. The primal idea of the CE method is to sample from $q(\boldsymbol{\gamma} \,|\, \theta_t)$ and estimate the parameter θ_{t+1} from past simulations, such that $q(\boldsymbol{\gamma} \,|\, \theta_{t+1})$ gets closer to $\delta_{\boldsymbol{\gamma}^*}(\boldsymbol{\gamma})$ in terms of some distance, originally the cross-entropy which is an information theoretic term for the Kullback-Leibler divergence.

More specifically, we sample $\boldsymbol{x}_k^{[t]} \sim q(\boldsymbol{\gamma} \,|\, \theta_t)$, independently for all $k \in N = \{1, \dots, n\}$, and order them according to the target function, such that

$$\pi(\boldsymbol{x}_{h_1}^{[t]}) \geq \cdots \geq \pi(\boldsymbol{x}_{h_n}^{[t]}), \quad h : N \to N. \tag{28}$$

We refer to the $\hat{n}$ draws with the highest scores, that is $\boldsymbol{x}_{h_1}^{[t]}, \dots, \boldsymbol{x}_{h_{\hat{n}}}^{[t]}$, as the elite sample, where $\hat{n} = \lceil \varrho \cdot n \rceil$ for some fraction $\varrho \approx 0.02$. Next, we choose the parameter θ_{t+1} that minimizes the divergence of the elite sample from the parametric family $q(\boldsymbol{\gamma} \,|\, \theta)$.

As Rubinstein and Kroese (2004, p. 45) remark, θ_{t+1} is just the maximum likelihood estimator based on the elite sample. Even though for some models discussed in section 3, we propose other fitting criteria than maximum-likelihood, e.g. matching the moments, the CE algorithm still remains valid.

5.3 Termination criterion

Rubinstein (1999) proposes to terminate the algorithm once the lowest value in the elite sample does not increase for $s > 0$ steps, that is

$$\pi(\boldsymbol{x}_{h_{\hat{n}}}^{[\tau-s]}) = \cdots = \pi(\boldsymbol{x}_{h_{\hat{n}}}^{[\tau]}). \tag{29}$$

For optimisation on binary spaces, we propose to rather monitor the number of components with marginal probabilites away from the bounds of $(0,1)$. For $\epsilon > 0$, we define the sets

$$L_\epsilon^{[t]} \stackrel{\text{def.}}{=} \{i \in D \,|\, \hat{s}_i^{[t]} < \epsilon\}, \quad U_\epsilon^{[t]} \stackrel{\text{def.}}{=} \{i \in D \,|\, \hat{s}_i^{[t]} > 1 - \epsilon\} \tag{30}$$

where $\hat{s}_i^{[t]} \stackrel{\text{def.}}{=} \hat{n}^{-1} \sum_{k=1}^{\hat{n}} x_{h_k,i}^{[t]}$ is the elite sample mean. Most probably, Algorithm 2 will converge to a maximiser $\boldsymbol{\gamma}^*$ in the subset

$$\boldsymbol{\Gamma}_\epsilon^{[t]} \stackrel{\text{def.}}{=} \{\boldsymbol{\gamma} \in \boldsymbol{\Gamma} \,|\, \boldsymbol{\gamma}_{L_\epsilon^{[t]}} = \mathbf{0}, \boldsymbol{\gamma}_{U_\epsilon^{[t]}} = \mathbf{1}\} \subseteq \boldsymbol{\Gamma}. \tag{31}$$

We stop the CE algorithm and solve the remaining problem via exhaustive search in $\boldsymbol{\Gamma}_\epsilon^{[t]}$, as soon as the number of strongly random components is sufficiently small, that is

$$(\text{src})_t \stackrel{\text{def.}}{=} \left|R_\epsilon^{[t]}\right| \leq u, \quad R_\epsilon^{[t]} \stackrel{\text{def.}}{=} D \setminus (L_\epsilon^{[t]} \cup U_\epsilon^{[t]}), \quad u \in \mathbb{N} \tag{32}$$

In our simulations, we choose $u = 12$. Note that for $u \leq \log(sn)/\log 2$, the reduction criterion (32) is more efficient than the convergence criterion (29).

Algorithm 2 A generic CE algorithm

0. Let $n^* \stackrel{\text{def.}}{=} \lceil \varrho \cdot n \rceil$ and $t = 0$.

1. We sample $\boldsymbol{x}_k^{[t]} \overset{\text{iid.}}{\sim} q(\boldsymbol{\gamma} \,|\, \theta_t)$ for all $k \in N = \{1, \ldots, n\}$.

2. We order $\boldsymbol{x}_{h_1}^{[t]}, \ldots, \boldsymbol{x}_{h_n}^{[t]}$ with respect to π, such that $\pi(\boldsymbol{x}_{h_1}^{[t]}) \geq, \ldots, \geq \pi(\boldsymbol{x}_{h_n}^{[t]})$.

3. We minimise the divergence of the elite sample from the parametric family q,

$$\theta_{t+1} = \underset{\theta}{\text{argmin}}\ \text{d}\left(\hat{n}^{-1} \sum_{k=1}^{\hat{n}} \delta_{\boldsymbol{x}_{h_k}} \,\middle|\, q(\cdot \,|\, \theta)\right).$$

4. If the reduction criterion (32) is fulfilled, we stop and run an exhaustive search on the subset $\boldsymbol{\Gamma}_\epsilon^{[t]}$. If the convergence criterion (29) is fullfilled, we stop and return $\boldsymbol{x}_{h_1}$. Otherwise, we return to **1**.

5.4 Numerical example

In this section, we give some remarks on how to efficiently implement Algorithm 2 and compare the Cross- Entropy approach to classic Simulated Annealing.

5.4.1 Remarks on the implementation

Here, we discuss the judicious choice of the number of particles n and comment on how to efficiently mix the independent and the logistic model.

We need to choose n large enough to ensure that the much smaller number $\hat{n}$ of elite samples still permits an accurate estimation of the parameter θ. For the independent model, few samples suffice to estimate the expected value $\boldsymbol{p}$. For the logistic regression model, however, we should have at least $\hat{n} > 10 \cdot d$, or the badly fit parameters $\boldsymbol{\beta}$ might guide the stochastic search in the wrong direction.

On the other hand, evaluating, in each step, $n > 10 \cdot d/\varrho$ times the target function $\pi(\boldsymbol{\gamma})$ considerably levels down the performance. We compromise using the mixture

$$q(\boldsymbol{\gamma} \,|\, \boldsymbol{p}_{\varrho_1}, \boldsymbol{\beta}_{\varrho_2}) \stackrel{\text{def.}}{=} \lambda q(\boldsymbol{\gamma} \,|\, \boldsymbol{p}_{\varrho_1}) + (1 - \lambda) q(\boldsymbol{\gamma} \,|\, \boldsymbol{\beta}_{\varrho_2}) \tag{33}$$

of an independent model with $\varrho_1 \approx 0.02$ and a logistic regression model $\varrho_2 \approx 0.15$. Typically, we choose λ to be $1/4$. Hence, we use a larger elite sample to fit the logistic model and mix it with an independent model to ensure the innovation and convergence of the CE algorithm. In our experiments, such a mixture significantly beats the pure independent model $q(\boldsymbol{\gamma} \,|\, \boldsymbol{p}_{\varrho_1})$ and performs a little better than a latent normal model $q(\boldsymbol{\gamma} \,|\, \boldsymbol{\mu}_{\varrho_1}, \boldsymbol{\Sigma}_{\varrho_2})$.

Besides, the remarks on model size reduction made in Section 4.5.1 also hold true for the CE algorithm. We monitor $R = R_{\varepsilon}^{[t]}$ as defined in (32) and draw $\boldsymbol{\gamma}_R \sim q(\boldsymbol{\gamma}_R \,|\, \boldsymbol{\beta}_R)$ from a suitable logistic model, while we generate the remaining components independently. In other words, (32) tells us to stop when the dimension of the logistic model is less than u.

5.4.2 Comparison to Simulated annealing

Based on the variable selection problem described in Section 2.4, we compare Algorithm 2 to a basic Simulated Annealing approach with a linear cooling schedule. In Figure 3, we ran each algorithm 200 times and plotted the histogram of the 10 highest detected modes; the outliers are collected in the rightmost bucket. We cannot guarantee that $\boldsymbol{\gamma}_1 = \operatorname{argmax}_{\boldsymbol{\gamma}} \pi(\boldsymbol{\gamma})$, but we may compare the relative performance of the CE algorithm versus SA.

Similar to Section 4.5, we allow the SA algorithm to perform twice as many evaluations of the posterior density compared to the CE method, in order to counterbalance the extra computational time we need to calibrate of the parametric family $q(\boldsymbol{\gamma} \,|\, \boldsymbol{p}_{\varrho_1}, \boldsymbol{\beta}_{\varrho_2})$.

We do not claim that, as Figure 3 might suggest, CE optimisation generally works better than SA. In principle, comparing these stochastic search algorithms is a delicate task, since the performance depends on a judicious choice of the numerous parameters $\lambda, \varrho_1, \varrho_2, \varepsilon$ and u for CE and on the cooling schedule for SA. However, we also ran this evaluation on several other problems and observed that, even if CE never converges to the highest mode found by SA, the variation of the CE results is significantly lower compared to SA. Hence, without giving detailed evidence, we believe that CE is more robust than SA.

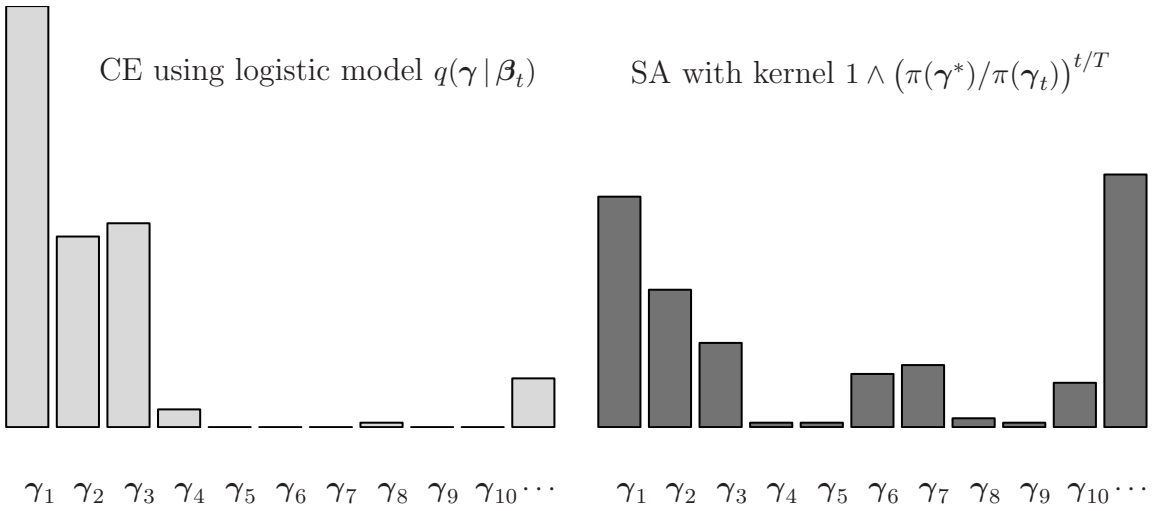


Figure 3: Histogram of 200 runs of Cross-Entropy optimisation and Simulated Annealing, where $\boldsymbol{\gamma}_1 = \boldsymbol{\gamma}^*$ and $\pi(\boldsymbol{\gamma}_1) \geq \cdots \geq \pi(\boldsymbol{\gamma}_{10})$. The dots $\cdots$ denote further outliers.

6 Outlook

In this paper, we discussed and successfully implemented two examples of Adaptive Monte Carlo algorithms on binary sampling spaces. Yet, there are several related topics we continue to work on.

6.1 Extensive comparison of global versus local methods

We did not explicitly argue that global methods like SMC and CE generally yield better results than Markov chain methods like MCMC and SA. However, there certainly is evidence which should be reinforced by broader numerical analysis. It would enrich the discussion to also test Adaptive MCMC schemes, as proposed by Nott and Kohn (2005), against our SMC approach. We plan to publish a collection of test datasets known to yield challenging posterior densities and will release the PYTHON code we used for our computations and evaluations.

6.2 Double parallelisation

In the introductory section we mentioned the fact that global methods are easy to parallelise. In our algorithms, we can even parallelise *twice*: we could evaluate the posterior $\pi(\boldsymbol{\gamma})$ in parallel for all particles and estimate the parameters $\boldsymbol{\beta}$ for the logistic model in parallel for all dimensions. Although technically demanding, an implementation of our algorithms using parallel computing on graphic cards would allow to process problems of magnitude 10^5 within hours, which would require weeks to be reliably solved by Markov chain methods.

Acknowledgements

N. Chopin is supported by the ANR grant ANR-008-BLAN-0218 "BigMC" of the French Ministry of research.

References

Albert, A. and Anderson, J. A. (1984). On the existence of maximum likelihood estimates in logistic regression models. *Biometrika*, (72):1–10.

Andrieu, C. and Thoms, J. (2008). A tutorial on adaptive MCMC. 18(4):343–373.

Bahadur, R. (1961). A representation of the joint distribution of responses to n dichotomous items. In Solomon, H., editor, *Studies in Item Analysis and Prediction*, pages pp. 158–68. Stanford University Press.

Cappé, O., Douc, R., Guillin, A., Marin, J., and Robert, C. (2008). Adaptive importance sampling in general mixture classes. *Statistics and Computing*, 18(4):447–459.

Carpenter, J., Clifford, P., and Fearnhead, P. (1999). Improved particle filter for nonlinear problems. *IEE Proc. Radar, Sonar Navigation*, 146(1):2–7.

Cox, D. (1972). The analysis of multivariate binary data. *Applied Statistics*, pages 113–120.

Cox, D. and Wermuth, N. (1994). A note on the quadratic exponential binary distribution. *Biometrika*, 81(2):403–408.

Cox, D. and Wermuth, N. (2002). On some models for multivariate binary variables parallel in complexity with the multivariate Gaussian distribution. *Biometrika*, 89(2):462.

Del Moral, P., Doucet, A., and Jasra, A. (2006). Sequential monte carlo samplers. *Journal of the Royal Statistical Society: Series B(Statistical Methodology)*, 68(3):411–436.

Divgi (1979). Computation of univariate and bivariate normal probability functions. *Annals of Statistics*, (7):903–910.

Drezner, Z. and Wesolowsky, G. O. (1990). On the computation of the bivariate normal integral. *Journal of Statistical Computation and Simulation*, (35):101–107.

Emrich, L. and Piedmonte, M. (1991). A method for generating high-dimensional multivariate binary variates. *The American Statistician*, 45(4):302–304.

Firth, D. (1993). Bias reduction of maximum likelihood estimates. *Biometrika*, (80):27–38.

Gelman, A. and Meng, X. (1998). Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statistical Science*, 13(2):163–185.

Genest, C. and Neslehova, J. (2007). A primer on copulas for count data. *Astin Bulletin*, 37(2):475.

George, E. and McCulloch, R. (1993). Variable selection via Gibbs sampling. *Journal of the American Statistical Association*, pages 881–889.

George, E. I. and McCulloch, R. E. (1997). Approaches for Bayesian variable selection. *Statistica Sinica*, (7):339–373.

Gilks, W. and Berzuini, C. (2001). Following a moving target — Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(1):127–146.

Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proc. F, Comm., Radar, Signal Proc.*, 140(2):107–113.

Green, P. J. (1984). Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. *Journal of the Royal Statistical Society*, (46):149–192.

Harrison, D. and Rubinfeld, D. L. (1978). Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, 5(1):81–102.

Higham, N. J. (2002). Computing the nearest correlation matrix — a problem from finance. *IMA Journal of Numerical Analysis*, (22):329–343.

Joe, H. (1996). Families of m-variate distributions with given margins and m (m-1)/2 bivariate dependence parameters. *Lecture Notes-Monograph Series*, 28:120–141.

Johnson, N., Kotz, S., and Balakrishnan, N. (2002). *Continuous Multivariate Distributions, volume 1, Models and Applications*. New York: John Wiley & Sons,.

Kitagawa, G. (1998). A self-organizing state-space model. *Journal of the American Statistical Association*, 93(443):1203–1215.

Kong, A., Liu, J. S., and Wong, W. H. (1994). Sequential imputation and Bayesian missing data problems. *Journal of the American statistical association*, 89:278–288.

Lee, A. (1993). Generating Random Binary Deviates Having Fixed Marginal Distributions and Specified Degrees of Association. *The American Statistician*, 47(3).

Lee, A., Yau, C., Giles, M., Doucet, A., and Holmes, C. (2009). On the utility of graphics cards to perform massively parallel simulation of advanced Monte Carlo methods. *Arxiv preprint arXiv:0905.2441 v3*.

Leisch, F., Weingessel, A., and Hornik, K. (1998). On the generation of correlated artificial binary data. *preparation*.

Liu, J. and Chen, R. (1998). Sequential Monte Carlo methods for dynamic systems. *Journal of the American statistical association*, 93(443):1032–1044.

Lunn, A. and Davies, S. (1998). A note on generating correlated binary variables. *Biometrika*, 85(2):487–490.

Madigan, D., York, J., and Allard, D. (1995). Bayesian graphical models for discrete data. *International Statistical Review/Revue Internationale de Statistique*, 63(2):215–232.

Mikosch, T. (2006). Copulas: Tales and facts. *Extremes*, 9(1):3–20.

Neal, R. (2001). Annealed importance sampling. *Statistics and Computing*, 11(2):125–139.

Nelsen, R. (2006). *An introduction to copulas*. Springer Verlag.

Nott, D. and Kohn, R. (2005). Adaptive sampling for Bayesian variable selection. *Biometrika*, 92(4):747.

Oman, S. and Zucker, D. (2001). Modelling and generating correlated binary variables. *Biometrika*, 88(1):287.

Park, C., Park, T., and Shin, D. (1996). A Simple Method for Generating Correlated Binary Variates. *The American Statistician*, 50(4).

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

Rubinstein, R. Y. (1999). The cross-entropy method for combinatorial and continuous optimization. *Methodology and Computing in Applied Probability*, pages 127–190.

Rubinstein, R. Y. and Kroese, D. P. (2004). *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation, and Machine Learning*. Springer-Verlag.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, (6):461–464.

Streitberg, B. (1990). Lancaster interactions revisited. *Annals of Statistics*, 18(4):1878–1885.

Streitberg, B. (1999). Exploring interactions in high-dimensional tables: a bootstrap alternative to log-linear models. *Annals of Statistics*, pages 405–413.

Suchard, M., Holmes, C., and West, M. (2010). Some of the What?, Why?, How?, Who? and Where? of Graphics Processing Unit Computing for Bayesian Analysis. In Bernardo, J. M. et al., O. U. P., editor, *Bayesian Statistics 9*.

Walker, A. (1977). An efficient method for generating discrete random variables with general distributions. *ACM Transactions on Mathematical Software*, 3(3):256.