Stratified Splitting for Efficient Monte Carlo Integration

Radislav Vaisman, Robert Salomone, and Dirk P. Kroese

School of Mathematics and Physics
The University of Queensland, Brisbane, Australia

E-mail: r.vaisman@uq.edu.au,
robert.salomone@uqconnect.edu.au,
kroese@maths.uq.edu.au

Summary.
The efficient evaluation of high-dimensional integrals is of importance in both theoretical and practical fields of science, such as Bayesian inference, statistical physics, and machine learning. However, due to the curse of dimensionality, deterministic numerical methods are inefficient in high-dimensional settings. Consequentially, for many practical problems one must resort to Monte Carlo estimation. In this paper, we introduce a novel Sequential Monte Carlo technique called Stratified Splitting which enjoys a number of desirable properties not found in existing methods. Specifically, the method provides unbiased estimates and can handle various integrand types including indicator functions, which are used in rare-event probability estimation problems. Moreover, this algorithm achieves a rigorous efficiency guarantee in terms of the required sample size. The results of our numerical experiments suggest that the Stratified Splitting method is capable of delivering accurate results for a wide variety of integration problems.

Keywords: Monte Carlo integration; Multilevel splitting; Markov chain Monte Carlo; Algorithmic efficiency; Sequential Monte Carlo; Resample-move; Nested sampling; Power posteriors

1. Introduction

We consider the evaluation of expectations and integrals of the form

$$\mathbb{E}_f\left[\varphi(\mathbf{X})\right] = \sum_{\mathscr{X}} \varphi(\mathbf{x}) f(\mathbf{x}) \quad \text{or} \quad \mathbb{E}_f\left[\varphi(\mathbf{X})\right] = \int_{\mathscr{X}} \varphi(\mathbf{x}) f(\mathbf{x})\, \mathrm{d}\mathbf{x},$$

where $\mathbf{X} \sim f$ is a random variable taking values in a set $\mathscr{X} \subseteq \mathbb{R}^d$, f is a probability density function (pdf) with respect to the Lebesgue or counting measure, and $\varphi : \mathscr{X} \to \mathbb{R}$ is a real-valued function.

The evaluation of such high-dimensional integrals is of critical importance in many scientific areas, including statistical inference (Gelman et al., 2003; Lee, 2004), rare-event estimation (Asmussen and Glynn, 2007), machine learning (Russell and Norvig, 2009; Koller and Friedman, 2009), and cryptography (McGrayne, 2011). An important application is the calculation of the normalizing constant of a probability distribution, such as the marginal likelihood (model evidence) in Bayesian statistics (Hooper, 2013). However, often obtaining even a reasonably accurate estimate of $\mathbb{E}_f\left[\varphi(\mathbf{X})\right]$ can be hard (Robert and Casella, 2004).

In this paper, we propose a novel Sequential Monte Carlo (SMC) approach for reliable and fast estimation of high-dimensional integrals. Our method extends the Generalized Splitting (GS) algorithm of Botev and Kroese (2012), to allow the estimation of quite general integrals. In addition, our algorithm is specifically designed to perform efficient sampling in regions of $\mathscr{X}$ where f takes small values and φ takes large values. In particular, we present a way of implementing stratification for variance reduction in the absence of knowing the strata probabilities. A major benefit of the proposed Stratified Splitting algorithm (SSA) is that it provides an unbiased estimator of $\mathbb{E}_f\left[\varphi(\mathbf{X})\right]$, and that it can be analyzed in a non-asymptotic setting. In particular, the SSA provides a bound on the sample size required to achieve a predefined error.

Due to its importance, the high-dimensional integration problem has been considered extensively in the past. Computation methods that use Fubini's theorem (Friedman, 1980) and quadrature rules or extrapolations (Forsythe et al., 1977), suffer from the curse of dimensionality, with the number of required function evaluations growing exponentially with the dimension. In order to address this problem, many methods have been proposed. Examples include Bayesian quadrature, sparse grids, and various Monte Carlo, quasi-Monte Carlo, and Markov Chain Monte

Carlo (MCMC) algorithms (O'Hagan, 1991; Morokoff and Caflisch, 1995; Newman and Barkema, 1999; Heiss and Winschel, 2008; Kroese et al., 2011). Many of these procedures are based on the SMC approach (Gilks and Berzuini, 2001; Chen et al., 2005; Del Moral et al., 2006; Friel and Pettitt, 2008; Andrieu et al., 2010), and provide consistent estimators that possess asymptotic normality. However, one might be interested in the actual number of required samples to achieve a predefined error bound. Our method, which also belongs to the SMC framework, is capable of addressing this issue.

Among alternative methods, we distinguish the Nested Sampling (NS) algorithm of Skilling (2006), the Annealed Importance Sampling (AIS) method of Neal (2001), and the Power posterior approach of Friel and Pettitt (2008), for their practical performance and high popularity (Murray et al., 2005; Feroz and Skilling, 2013; Andrieu et al., 2010). As always, due to the varied approaches of different methods and nuances of different problems, no individual method can be deemed universally better. For example, despite good practical performance and convergence in probability to the true integral value, the NS algorithm is not unbiased and in fact, to ensure its consistency, both sample size and ratio of sampling iterations to sample population size should be infinite for certain classes of integrands (Evans, 2007). Moreover, consistency of estimates obtained with Nested Sampling when Markov Chain Monte Carlo (MCMC) is used for sampling remains an open problem (Chopin and Robert, 2010).

Similar to other well-known SMC methods, the SSA falls into a multi-level estimation framework, which will be detailed in Section 2. As with classical stratified sampling (see, e.g., Rubinstein and Kroese (2017), Chapter 5), the SSA defines a partition of the state space into strata, and uses the law of total probability to deliver an estimator of the value of the integral. To do so, one needs to obtain a sample population from each strata and know the exact probability of each such strata. Under the classical stratified sampling framework, it is assumed that the former is easy to achieve and the latter is known in advance. However, such favorable scenarios are rarely seen in practice. In particular, obtaining samples from within a stratum and estimating the associated probability that a sample will be within this stratum is hard in general (Jerrum et al., 1986). To resolve this issue, the SSA incorporates a multi-level splitting mechanism (Kahn and Harris, 1951; Botev and Kroese, 2012; Rubinstein et al., 2013) and uses an appropriate MCMC method to sample from

conditional densities associated with a particular stratum.

The rest of the paper is organized as follows. In Section 2 we introduce the SSA, explain its correspondence to a generic multi-level sampling framework, and prove that the SSA delivers an unbiased estimator of the expectation of interest. In Section 3, we provide a rigorous analysis of the approximation error of the proposed method. In Section 4, we introduce a difficult estimation problem called the weighted component model, for which the SSA provides the best possible efficiency result one can hope to achieve. Namely, we show that the SSA can obtain an arbitrary level of precision by using a sample size (and computation time) that is polynomial in the corresponding problem size. In Section 5, we report our numerical findings on various test cases that typify classes of problems for which the SSA is of practical interest. Finally, in Section 6 we summarize the results and discuss possible directions for future research.

2. Stratified splitting algorithm

2.1. Generic multilevel splitting framework

We begin by considering a very generic multilevel splitting framework, similar to (Gilks and Berzuini, 2001). Let $\mathbf{X} \sim f$ be a random variable taking values in a set $\mathscr{X}$, and consider a decreasing sequence of sets $\mathscr{X} = \mathscr{X}_0 \supseteq \cdots \supseteq \mathscr{X}_n = \emptyset$. Define $\mathscr{Z}_t = \mathscr{X}_{t-1} \setminus \mathscr{X}_t$, for $t = 1, \ldots, n$, and note that $\mathscr{X}_{t-1} = \bigcup_{i=t}^{n} \mathscr{Z}_i$, and that $\{\mathscr{Z}_t\}$ yields a partition of $\mathscr{X}$; that is

$$\mathscr{X} = \bigcup_{t=1}^{n} \mathscr{Z}_t, \quad \mathscr{Z}_{t_1} \cap \mathscr{Z}_{t_2} = \emptyset \quad \text{for} \quad 1 \leq t_1 < t_2 \leq n. \tag{1}$$

Then, we can define a sequence of conditional pdfs

$$f_t(\mathbf{x}) = f(\mathbf{x} \mid \mathbf{x} \in \mathscr{X}_{t-1}) = \frac{f(\mathbf{x}) \mathbb{1}\{\mathbf{x} \in \mathscr{X}_{t-1}\}}{\mathbb{P}_f(\mathbf{X} \in \mathscr{X}_{t-1})} \quad \text{for} \quad t = 1, \ldots, n, \tag{2}$$

where $\mathbb{1}$ denotes the indicator function. Also, define

$$g_t(\mathbf{x}) = f\left(\mathbf{x} \mid \mathbf{x} \in \mathscr{Z}_t\right) = \frac{f(\mathbf{x}) \mathbb{1}\{\mathbf{x} \in \mathscr{Z}_t\}}{\mathbb{P}_f(\mathbf{X} \in \mathscr{Z}_t)} \quad \text{for} \quad t = 1, \ldots, n. \tag{3}$$

Our main objective is to sample from the pdfs f_t and g_t in (2) and (3), respectively. To do so, we first formulate a generic multilevel splitting framework, given in Algorithm 1.

Algorithm 1: Generic multilevel splitting framework

input : $\mathscr{X}_0, \ldots, \mathscr{X}_n$ and $\{f_t, g_t\}_{1 \le t \le n}$.
output: Samples from f_t and g_t for $1 \le t \le n$.
Create a multi-set $\mathcal{X}_1$ of samples from f_1.
for $t = 1$ **to** n **do**
  Set $\mathcal{Z}_t \leftarrow \mathcal{X}_t \cap \mathscr{X}_t$.
  Set $\mathcal{Y}_t \leftarrow \mathcal{X}_t \setminus \mathcal{Z}_t$.
  if $t < n$ **then**
    Create a multi-set $\mathcal{X}_{t+1}$ of samples (particles) from f_{t+1}, (possibly) using elements of the $\mathcal{Y}_t$ set. This step is called the splitting or the rejuvenation step.
return multi-sets $\{\mathcal{X}_t\}_{1 \le t \le n}$, and $\{\mathcal{Z}_t\}_{1 \le t \le n}$.

Note that the samples in $\{\mathcal{X}_t\}_{1 \le t \le n}$ and $\{\mathcal{Z}_t\}_{1 \le t \le n}$ are distributed according to f_t and g_t, respectively, and these samples can be used to handle several tasks. In particular, the $\{\mathcal{X}_t\}_{1 \le t \le n}$ sets allow one to handle the general non-linear Bayesian filtering problem (Gilks and Berzuini, 2001; Gordon et al., 1993; Del Moral et al., 2006). Moreover, by tracking the cardinalities of the sets $\{\mathcal{X}_t\}_{1 \le t \le n}$ and $\{\mathcal{Z}_t\}_{1 \le t \le n}$, one is able to tackle hard rare-event probability estimation problems, such as delivering estimates of $\mathbb{P}_f(\mathbf{X} \in \mathscr{Z}_n)$ (Botev and Kroese, 2012; Kroese et al., 2011; Rubinstein et al., 2013). Finally, it was recently shown by Vaisman et al. (2016) that Algorithm 1 can be used as a powerful variance minimization technique for any general SMC procedure. In light of the above, we propose taking further advantage of the sets $\{\mathcal{X}_t\}_{1 \le t \le n}$ and $\{\mathcal{Z}_t\}_{1 \le t \le n}$, to obtain an estimation method suitable for general integration problems.

2.2. The SSA set-up

Following the above multilevel splitting framework, it is convenient to construct the sequence of sets $\{\mathscr{X}_t\}_{0 \le t \le n}$ by using a performance function $S : \mathscr{X} \to \mathbb{R}$, such that $\{\mathscr{X}_t\}_{0 \le t \le n}$ can be written as *super* level-sets of S for chosen levels $\gamma_0, \ldots, \gamma_n$, where γ_0 and γ_n are equal to $\inf_{\mathbf{x} \in \mathscr{X}} S(\mathbf{x})$ and $\sup_{\mathbf{x} \in \mathscr{X}} S(\mathbf{x})$, respectively. In particular, $\mathscr{X}_t = \{\mathbf{x} \in \mathscr{X} : S(\mathbf{x}) \ge \gamma_t\}$

for $t = 0, \ldots, n$. The partition $\{\mathscr{Z}_t\}_{1 \le t \le n}$, and the densities $\{f_t\}_{1 \le t \le n}$ and $\{g_t\}_{1 \le t \le n}$, are defined as before via (1), (2), and (3), respectively. Similarly, one can define a sequence of *sub* level-sets of S; in this paper we use the latter for some cases and whenever appropriate.

Letting $z_t \stackrel{\text{def}}{=} \mathbb{E}_f\left[\varphi\left(\mathbf{X}\right) \mid \mathbf{X} \in \mathscr{Z}_t\right] \mathbb{P}_f\left(\mathbf{X} \in \mathscr{Z}_t\right)$ for $t = 1, \ldots, n$, and combining (1) with the law of total probability, we arrive at

$$z \stackrel{\text{def}}{=} \mathbb{E}_f\left[\varphi\left(\mathbf{X}\right)\right] = \sum_{t=1}^{n} \mathbb{E}_f\left[\varphi\left(\mathbf{X}\right) \mid \mathbf{X} \in \mathscr{Z}_t\right] \mathbb{P}_f\left(\mathbf{X} \in \mathscr{Z}_t\right) = \sum_{t=1}^{n} z_t. \quad (4)$$

The SSA proceeds with the construction of estimators $\widehat{Z}_t$ for z_t for $t = 1, \ldots, n$ and, as soon as these are available, we can use (4) to deliver the SSA estimator for z, namely $\widehat{Z} = \sum_{t=1}^{n} \widehat{Z}_t$.

For $1 \le t \le n$, let $\varphi_t \stackrel{\text{def}}{=} \mathbb{E}_f\left[\varphi\left(\mathbf{X}\right) \mid \mathbf{X} \in \mathscr{Z}_t\right]$, $p_t \stackrel{\text{def}}{=} \mathbb{P}_f\left(\mathbf{X} \in \mathscr{Z}_t\right)$, and let $\widehat{\Phi}_t$ and $\widehat{P}_t$ be estimators of φ_t and p_t, respectively. We define $\widehat{Z}_t = \widehat{\Phi}_t \, \widehat{P}_t$, and recall that, under the multilevel splitting framework, we obtain the sets $\{\mathcal{X}_t\}_{1 \le t \le n}$, and $\{\mathcal{Z}_t\}_{1 \le t \le n}$. These sets are sufficient to obtain unbiased estimators $\{\widehat{\Phi}_t\}_{1 \le t \le n}$ and $\{\widehat{P}_t\}_{1 \le t \le n}$, in the following way.

(a) We define $\widehat{\Phi}_t$ to be the (unbiased) Crude Monte Carlo (CMC) estimator of φ_t, that is,

$$\widehat{\Phi}_t = \frac{1}{|\mathcal{Z}_t|} \sum_{\mathbf{Z} \in \mathcal{Z}_t} \varphi(\mathbf{Z}) \quad \text{for all } \; t = 1, \ldots, n.$$

(b) The estimator $\widehat{P}_t$ is defined similar to the one used in the Generalized Splitting (GS) algorithm of Botev and Kroese (2012). In particular, the GS product estimator is defined as follows. Define the level entrance probabilities $r_0 \stackrel{\text{def}}{=} 1$, $r_t \stackrel{\text{def}}{=} \mathbb{P}_f\left(\mathbf{X} \in \mathscr{X}_t \mid \mathbf{X} \in \mathscr{X}_{t-1}\right)$ for $t = 1, \ldots, n$, and note that $\mathbb{P}_f\left(\mathbf{X} \in \mathscr{X}_t\right) = \prod_{i=0}^{t} r_i$. Then, for $t = 1, \ldots, n$, it holds that

$$\begin{aligned} p_t &= \mathbb{P}_f\left(\mathbf{X} \in \mathscr{Z}_t\right) = \mathbb{P}_f\left(\mathbf{X} \in \mathscr{X}_{t-1}\right) - \mathbb{P}_f\left(\mathbf{X} \in \mathscr{X}_t\right) \\ &= \prod_{i=0}^{t-1} r_i - \prod_{i=0}^{t} r_i = \left(1 - r_t\right) \prod_{i=0}^{t-1} r_i. \end{aligned}$$

This suggests the estimator $\widehat{P}_t = (1 - \widehat{R}_t) \prod_{i=0}^{t-1} \widehat{R}_i$, for p_t, where $\widehat{R}_0 \stackrel{\text{def}}{=} 1$, and $\widehat{R}_t = \frac{|\mathcal{Y}_t|}{|\mathcal{X}_t|} = \frac{|\mathcal{X}_t \setminus \mathcal{Z}_t|}{|\mathcal{X}_t|}$, for all $t = 1, \ldots, n$.

In practice, obtaining the $\{\mathcal{X}_t\}_{1 \le t \le n}$ and $\{\mathcal{Z}_t\}_{1 \le t \le n}$ sets requires the implementation of a sampling procedure from the conditional pdfs in (2) and (3). However, for many real-life applications, designing such a procedure can be extremely challenging. Nevertheless, we can use the $\mathcal{Y}_t = \mathcal{X}_t \setminus \mathcal{Z}_t$ set from iteration t, to sample $\mathcal{X}_{t+1}$ from f_{t+1} for each $t = 1, \ldots, n-1$, via MCMC. In particular, the particles from the $\mathcal{Y}_t$ set can be "split", in order to construct the desired set $\mathcal{X}_{t+1}$ for the next iteration, using a Markov transition kernel $\kappa_{t+1}(\cdot \mid \cdot)$ whose stationary pdf is f_{t+1}, for each $t = 1, \ldots, n-1$. Algorithm 2 summarizes the general procedure for the SSA.

Algorithm 2: The SSA for estimating $z = \mathbb{E}_f[\varphi(\mathbf{X})]$

input : A set $\mathscr{X}$, a pdf f, the functions $\varphi : \mathscr{X} \to \mathbb{R}$ and $S : \mathscr{X} \to \mathbb{R}$, a sequence of levels $\gamma_0, \ldots, \gamma_n$, and the sample size $N \in \mathbb{N}$.
output: $\widehat{Z}$ — an estimator of $z = \mathbb{E}_f[\varphi(\mathbf{X})]$.
Set $\widehat{R}_0 \leftarrow 1$, $\mathcal{X}_1 \leftarrow \emptyset$, and $f_1 \leftarrow f$.
for $i = 1$ **to** N **do**
  draw $\mathbf{X} \sim f_1(\mathbf{x})$ and add $\mathbf{X}$ to $\mathcal{X}_1$.
for $t = 1$ **to** n **do**
  Set $\mathcal{Z}_t \leftarrow \{\mathbf{X} \in \mathcal{X}_t : \mathbf{X} \in \mathscr{Z}_t\}$ and $\mathcal{Y}_t \leftarrow \mathcal{X}_t \setminus \mathcal{Z}_t$.
  Set $\widehat{\Phi}_t \leftarrow \frac{1}{|\mathcal{Z}_t|} \sum_{\mathbf{X} \in \mathcal{Z}_t} \varphi(\mathbf{X})$.
  Set $\widehat{R}_t \leftarrow \frac{|\mathcal{Y}_t|}{N}$, and $\widehat{P}_t \leftarrow \left(1 - \widehat{R}_t\right) \prod_{j=0}^{t-1} \widehat{R}_j$.
  Set $\widehat{Z}_t \leftarrow \widehat{\Phi}_t \widehat{P}_t$.
  if $t < n$ **then**
    `/* Performing splitting to obtain` $\mathcal{X}_{t+1}$. `*/`
    Set $\mathcal{X}_{t+1} \leftarrow \emptyset$ and draw $K_i \sim \mathsf{Bernoulli}(0.5)$, for $i = 1, \ldots, |\mathcal{Y}_t|$, such that $\sum_{i=1}^{|\mathcal{Y}_t|} K_i = N \bmod |\mathcal{Y}_t|$.
    for $\mathbf{Y} \in \mathcal{Y}_t$ **do**
      Set $M_i \leftarrow \left\lfloor \frac{N}{|\mathcal{Y}_t|} \right\rfloor + K_i$ and $\mathbf{X}_{i,0} \leftarrow \mathbf{Y}$.
      for $j = 1$ **to** M_i **do**
        Draw $\mathbf{X}_{i,j} \sim \kappa_{t+1}^{\tau}(\cdot \mid \mathbf{X}_{i,j-1})$ (where $\kappa_{t+1}^{\tau}(\cdot \mid \cdot)$ is a τ-step transition kernel using $\kappa_{t+1}(\cdot \mid \cdot)$), and add $\mathbf{X}_{i,j}$ to $\mathcal{X}_{t+1}$.
return $\widehat{Z} = \sum_{t=1}^{n} \widehat{Z}_t$.

REMARK 2.1 (THE SPLITTING STEP). The particular splitting step described in Algorithm 2 is a popular choice (Botev and Kroese, 2012), especially for hard problems with unknown convergence behavior of the corresponding Markov chain. However, one can also apply different splitting strategies. For example, we can choose a single element from $\mathcal{Y}_t$, and use it to obtain all samples in the $\mathcal{X}_{t+1}$ set. Note that, in this case, $\mathcal{X}_{t+1}$ contains dependent samples. On the other hand, we might be interested to have independent samples in the $\mathcal{X}_{t+1}$ set. To do so, one will generally perform additional runs of the SSA, and take a single element (from each SSA run) from $\mathcal{Y}_t$ to produce a corresponding sample in the $\mathcal{X}_{t+1}$ set. Such a strategy is clearly more expensive computationally. Under this setting, a single SSA run will require a computational effort that is proportional to that of Algorithm 2, squared. However, such an approach is beneficial for an analysis of the SSA's convergence. See also Remark A.1.

THEOREM 2.1 (UNBIASED ESTIMATOR). *Algorithm 2 outputs an unbiased estimator; that is, it holds that* $\mathbb{E}\left[\widehat{Z}\right] = \mathbb{E}_f\left[\varphi\left(\mathbf{X}\right)\right] = z$.

PROOF. See Appendix A.

An immediate consequence of Theorem 2.1 is that the SSA introduces an advantage over conventional SMC algorithms, which provide only consistent estimators.

We next proceed with a clarification for a few remaining practical issues regarding the SSA.

Determining the SSA levels. It is often difficult to make an educated guess how to set the values of the level thresholds. However, the SSA requires the values of $\{\gamma_t\}_{1\leq t\leq n}$ to be known in advance, in order to ensure that the estimator is unbiased. To resolve this issue, we perform a single pilot run of Algorithm 2 using a so-called rarity parameter $0 < \rho < 1$. In particular, given samples from an $\mathcal{X}_t$ set, we take the $\rho\,|\mathcal{X}_t|$ performance quantile as the value of the corresponding level γ_t, and form the next level set. Such a pilot run, helps to establish a set of threshold values adapted to the specific problem. After the completion of the pilot run we simply continue with a regular execution of Algorithm 2 using the level threshold values observed in the pilot run.

Controlling the SSA error. A common practice when working with a Monte Carlo algorithm that outputs an unbiased estimator, is to run it

for R independent replications to obtain $\widehat{Z}^{(1)}, \ldots, \widehat{Z}^{(R)}$, and report the average value. Thus, for a final estimator, we take

$$\widehat{Z} = R^{-1} \sum_{j=1}^{R} \widehat{Z}^{(j)}.$$

To measure the quality of the SSA output, we use the estimator's relative error (RE), which is defined by

$$\mathrm{RE} = \sqrt{\mathrm{Var}\left(\widehat{Z}\right)} \Big/ \mathbb{E}\left[\widehat{Z}\right] \sqrt{R}.$$

As the variance and expectation of the estimator are not known explicitly, we report an estimate of the relative error by estimating both terms from the result of the R runs.

In the following section, we establish efficiency results for our estimator by conducting an analysis common in the field of randomized algorithms.

3. Efficiency of the SSA

In this section, we present an analysis of the SSA under a set of very general assumptions. We start with a definition a randomized algorithm's efficiency.

Definition 3.1 (Mitzenmacher and Upfal (2005)). *A randomized algorithm gives an (ε, δ)-approximation for the value z if the output $\widehat{Z}$ of the algorithm satisfies*

$$\mathbb{P}\left(z(1-\varepsilon) \leq \widehat{Z} \leq z(1+\varepsilon)\right) \geq 1 - \delta.$$

With the above definition in mind, we now aim to specify the sufficient conditions for the SSA to provide an (ε, δ)-approximation to z. The proof closely follows a technique that is used for the analysis of approximate counting algorithms. For an extensive overview, we refer to (Mitzenmacher and Upfal, 2005, Chapter 10). A key component in our analysis is to construct a Markov chain $\left\{X_t^{(m)},\, m \geq 0\right\}$ with stationary pdf f_t

(defined in (2)), for all $1 \leq t \leq n$, and to consider the speed of convergence of the distribution of $X_t^{(m)}$ as m increases. Let μ_t be the probability distribution corresponding to f_t, so

$$\mu_t(A) = \int_A f_t(u)\lambda(\mathrm{d}u),$$

for all Borel sets A, where λ is some base measure, such as the Lebesgue or counting measure. To proceed, we have

$$\kappa_t^\tau(A \mid \mathbf{x}) = \mathbb{P}\left(X_t^{(\tau)} \in A \mid X_t^{(0)} = \mathbf{x}\right),$$

for the τ-step transition law of the Markov chain. Consider the *total variation distance* between $\kappa_t^\tau(\cdot \mid \mathbf{x})$ and μ_t, defined as:

$$\|\kappa_t^\tau(\cdot \mid \mathbf{x}) - \mu_t\|_{\mathrm{TV}} = \sup_A |\kappa_t^\tau(A \mid \mathbf{x}) - \mu_t(A)|.$$

An essential ingredient of our analysis is the so-called mixing time (see Roberts et al. (2004) and Levin et al. (2009) for an extensive overview), which is defined as $\tau_{\mathrm{mix}}(\varepsilon, \mathbf{x}) = \min\{\tau : \|\kappa_t^\tau(\cdot \mid \mathbf{x}) - \mu_t\|_{\mathrm{TV}} \leq \varepsilon\}$. Let $\widehat{\mu}_t = \kappa_t^\tau(\cdot \mid \mathbf{x})$ be the SSA sampling distribution at steps $1 \leq t \leq n$, where for simplicity, we suppress $\mathbf{x}$ in the notation of $\widehat{\mu}_t$.

Finally, similar to μ_t and $\widehat{\mu}_t$, let ν_t be the probability distribution corresponding to the pdf g_t (defined in (3)), and let $\widehat{\nu}_t$ be the SSA sampling distribution, for all $1 \leq t \leq n$. Theorem 3.1 details the main efficiency result for the SSA.

THEOREM 3.1 (EFFICIENCY OF THE SSA). *Let φ be a strictly positive real-valued function, $a_t = \min_{\mathbf{x}\in\mathscr{X}_t}\{\varphi(\mathbf{x})\}$, $b_t = \max_{\mathbf{x}\in\mathscr{X}_t}\{\varphi(\mathbf{x})\}$, and $\underline{r}_t = \min\{r_t, 1 - r_t\}$ for $1 \leq t \leq n$. Then, the SSA gives an (ε, δ)-approximation to $z = \mathbb{E}_f[\varphi(\mathbf{X})]$, provided that for all $1 \leq t \leq n$, the following holds.*

(a) The samples in the $\mathcal{X}_t$ set are independent and are distributed according to $\widehat{\mu}_t$, such that (for every $\mathbf{x}$)

$$\|\widehat{\mu}_t - \mu_t\|_{\mathrm{TV}} \leq \frac{\varepsilon\,\underline{r}_t}{32n} \quad \textit{and} \quad |\mathcal{X}_t| \geq \frac{3072\,n^2\ln(4n^2/\delta)}{\varepsilon^2\underline{r}_t^2}.$$

(b) The samples in the $\mathcal{Z}_t$ set are independent and are distributed according to $\widehat{\nu}_t$, such that (for every $\mathbf{x}$)

$$\|\widehat{\nu}_t - \nu_t\|_{\mathrm{TV}} \leq \frac{\varepsilon\, a_t}{16(b_t - a_t)} \quad \textit{and} \quad |\mathcal{Z}_t| \geq \frac{128(b_t - a_t)^2 \ln(4n/\delta)}{\varepsilon^2 a_t^2}.$$

Proof. See Appendix A.

In some cases, the distributions of the states in $\mathscr{X}_t$ and $\mathscr{Z}_t$ generated by Markov chain defined by the kernel κ_t^{τ}, approach the target distributions μ_t and ν_t very fast. This occurs for example when there exists a polynomial in n (denoted by $\mathcal{P}(n)$), such that the mixing time (Levin et al., 2009) is bounded by $\mathcal{O}(\mathcal{P}(n))$, $(b_t - a_t)^2/a^2 = \mathcal{O}(\mathcal{P}(n))$, and $\underline{r}_t = \mathcal{O}(1/\mathcal{P}(n))$ for all $1 \leq t \leq n$. In this case, the SSA becomes a *fully polynomial randomized approximation scheme* (FPRAS) (Mitzenmacher and Upfal, 2005). In particular, the SSA results in a desired (ε, δ)-approximation to $z = \mathbb{E}_f[\varphi(\mathbf{X})]$ with running time bounded by a polynomial in n, ε^{-1}, and $\ln(\delta^{-1})$. Finally, it is important to note that an FPRAS algorithm for such problems is essentially the best result one can hope to achieve (Jerrum and Sinclair, 1996).

We next continue with a non-trivial example for which the SSA provides an FPRAS.

4. FPRAS for the weighted component model

We consider a system of k components. Each component i generates a specific amount of benefit, which is given by a positive real number w_i, $i = 1, \ldots, k$. In addition, each component can be operational or not.

Let $\mathbf{w} = (w_1, \ldots, w_k)^\top$ be the column vector of component weights (benefits), and $\mathbf{x} = (x_1, \ldots, x_k)^\top$ be a binary column vector, for which x_i indicates the ith component's operational status for $1 \leq i \leq k$. That is, if the component i is operational $x_i = 1$, and $x_i = 0$ if it is not. Under this setting, we define the system performance as

$$S(\mathbf{w}, \mathbf{x}) = \sum_{i=1}^{k} w_i\, x_i = \mathbf{w}^\top \mathbf{x}.$$

We further assume that all elements are independent from each other, and that each element is operational with probability 1/2 at any given

time. For the above system definition, we might be interested in the following questions.

(a) *Conditional expectation estimation.* Given a minimal threshold performance $\gamma \leq \sum_{i=1}^{k} w_i$, what is the expected system performance? That is to say, we are interested in the calculation of

$$\mathbb{E}\left[S(\mathbf{w}, \mathbf{X}) \mid S(\mathbf{w}, \mathbf{X}) \leq \gamma\right], \tag{5}$$

where $\mathbf{X}$ is a k-dimensional binary vector generated uniformly at random from the $\{0,1\}^k$ set. This setting appears (in a more general form), in a portfolio credit risk analysis (Glasserman and Li, 2005), and will be discussed in Section 5.

(b) *Tail probability estimation (Asmussen and Glynn, 2007).* Given the minimal threshold performance γ, what is the probability that the overall system performance is smaller than γ? In other words, we are interested in calculating

$$\mathbb{P}(S(\mathbf{X}) \leq \gamma) = \mathbb{E}\left[\mathbb{1}\{S(\mathbf{X}) \leq \gamma\}\right]. \tag{6}$$

The above problems are both difficult, since a uniform generation of $\mathbf{X} \in \{0,1\}^k$, such that $\mathbf{w}^\top \mathbf{X} \leq \gamma$, corresponds to the *knapsack* problem, which belongs to #P complexity class (Valiant, 1979; Morris and Sinclair, 2004).

In this section, we show how one can construct an FPRAS for both problems under the mild condition that the difference between the minimal and the maximal weight in the $\mathbf{w}$ vector is not large. This section's main result is summarized next.

PROPOSITION 4.1. *Given a weighted component model with k weights, $\mathbf{w} = (w_1, \ldots, w_k)$ and a threshold γ, let $\underline{w} = \min\{\mathbf{w}\}$, $\overline{w} = \max\{\mathbf{w}\}$. Then, provided that $\overline{w} = \mathcal{O}\left(\mathcal{P}(k)\right)\underline{w}$, there exists an FPRAS for the estimation of both* (5) *and* (6).

Prior to stating the proof of Proposition 4.1, define

$$\mathscr{X}_b = \left\{ \mathbf{x} \in \{0,1\}^k \ : \ \sum_{i=1}^{k} w_i \, x_i \leq b \right\} \quad \text{for } b \in \mathbb{R}, \tag{7}$$

and let μ_b be the uniform distribution on the $\mathscr{X}_b$ set. Morris and Sinclair (2004) introduce an MCMC algorithm that is capable of sampling from the $\mathscr{X}_b$ set almost uniformly at random. In particular, this algorithm can sample $\mathbf{X} \sim \widehat{\mu}_b$, such that $\|\widehat{\mu}_b - \mu_b\|_{\mathrm{TV}} \leq \varepsilon$. Moreover, the authors show that their Markov chain mixes rapidly, and in particular, that its mixing time is polynomial in k and is given by $\tau_{\mathrm{mix}}(\varepsilon) = \mathcal{O}\left(k^{9/2+\varepsilon}\right)$. Consequentially, the sampling from $\widehat{\mu}_b$ can be performed in $\mathcal{O}(\mathcal{P}(k))$ time for any $\varepsilon > 0$.

We next proceed with the proof of Proposition 4.1 which is divided into two parts. The first for the conditional expectation estimation and the second for the tail probability evaluation.

Proof (Proposition 4.1: FPRAS for (5)). With the powerful result of Morris and Sinclair (2004) in hand, one can achieve a straightforward development of an FPRAS for the conditional expectation estimation problem. The proof follows immediately from Lemma A.3.

In particular all we need to do in order to achieve an (ε, δ) approximation to (5), is to generate

$$m = \frac{(\overline{w} - \underline{w})^2 \ln(2/\delta)}{2(\varepsilon/4)^2 \underline{w}^2}$$

samples from $\widehat{\mu}_\gamma$, such that

$$\|\widehat{\mu}_\gamma - \mu_\gamma\|_{\mathrm{TV}} \leq \frac{\varepsilon\, \underline{w}}{4(\overline{w} - \underline{w})}.$$

Recall that the mixing time is polynomial in k, and note that the number of samples m is also polynomial in k, thus the proof is complete, since

$$m = \frac{(\overline{w} - \underline{w})^2 \ln(2/\delta)}{(\varepsilon/4)^2\, \underline{w}^2} \underbrace{=}_{\overline{w}=\mathcal{O}(\mathcal{P}(k))\underline{w}} \mathcal{O}(\mathcal{P}(k)) \frac{\ln(2/\delta)}{\varepsilon^2}. \qquad \square$$

Of course, the development of an FPRAS for the above case did not use the SSA, as an appropriate choice of transition kernel was sufficient for the result. The tail probability estimation problem, however, is more involved, and can be solved via the use of the SSA.

Proof (Proposition 4.1: FPRAS for (6)). In order to put this problem into the SSA setting and achieve an FPRAS, a careful definition

of the corresponding level sets is essential. In particular, the number of levels should be polynomial in k, and the level entrance probabilities $\{r_t\}_{1\leq t\leq n}$, should not be too small. Fix

$$n = \left\lfloor \frac{\left(\sum_{i=1}^{k} w_i\right) - \gamma}{\underline{w}} \right\rfloor,$$

to be the number of levels, and set $\gamma_t = \gamma + (n-t)\,\underline{w}$ for $t = 0, \ldots, n$. For general γ it holds that

$$\begin{aligned}
&\mathbb{E}\left[\mathbb{1}\{S(\mathbf{X}) \leq \gamma\}\right] = \mathbb{E}\left[\mathbb{1}\{S(\mathbf{X}) \leq \gamma + n\,\underline{w}\} \mid S(\mathbf{X}) \leq \gamma + n\,\underline{w}\right] \mathbb{P}\left(S(\mathbf{X}) \leq \gamma + n\,\underline{w}\right) \\
&+ \mathbb{E}\left[\mathbb{1}\{S(\mathbf{X}) \leq \gamma\} \mid \gamma + n\,\underline{w} < S(\mathbf{X}) \leq \sum_{i=1}^{k} w_i\right] \mathbb{P}\left(\gamma + n\,\underline{w} < S(\mathbf{X}) \leq \sum_{i=1}^{k} w_i\right) \\
&= \underbrace{\mathbb{E}\left[\mathbb{1}\{S(\mathbf{X}) \leq \gamma\} \mid S(\mathbf{X}) \leq \gamma + n\,\underline{w}\right]}_{(*)} \frac{2^k - 1}{2^k} + \left(\sum_{i=1}^{k} w_i\right) \frac{1}{2^k},
\end{aligned}$$

where the last equality follows from the fact that there is only one vector $\mathbf{x} = (1, 1, \ldots, 1)$ for which $\gamma + n\,\underline{w} < S(\mathbf{x}) \leq \sum_{i=1}^{k} w_i$. That is, it is sufficient to develop an efficient approximation to $(*)$ only, since the rest are constants.

We continue by defining the sets $\mathscr{X} = \mathscr{X}_{\gamma_0} \supseteq \cdots \supseteq \mathscr{X}_{\gamma_n}$ via (7), and by noting that for this particular problem, our aim is to find $\mathbb{P}(\mathbf{X} \in \mathscr{X}_{\gamma_n})$, so the SSA estimator simplifies into (see Section 2.2 (b)),

$$\widehat{Z} = \prod_{t=0}^{n} \widehat{R}_t.$$

In order to show that the SSA provides an FPRAS, we will need to justify only condition (a) of Theorem 3.1, which is sufficient in our case because we are dealing with an indicator integrand. Recall that the formal requirement is

$$\|\widehat{\mu}_t - \mu_t\|_{\mathrm{TV}} \leq \varepsilon\, \underline{r}_t / 32n, \quad \text{and} \quad |\mathcal{X}_t| \geq 3072\, n^2 \ln(4n^2/\delta)/\varepsilon^2 \underline{r}_t^2,$$

where μ_t is the uniform distribution on $\mathscr{X}_{\gamma_t}$ for $t = 0, \ldots, n$, and each sample in $\mathcal{X}_t$ is distributed according to $\widehat{\mu}_t$. Finally, the FPRAS result is established by noting that the following holds.

(a) From Lemma A.6, we have that $\underline{r}_t \geq \frac{1}{k+1}$ for $1 \leq t \leq n$.

(b) The sampling from $\widehat{\mu}_t$ can be performed in polynomial (in k) time (Morris and Sinclair, 2004).

(c) The number of levels n (and thus the required sample size $\{|\mathcal{X}_t|\}_{1\leq t\leq n}$), is polynomial in k since

$$\left\lfloor \frac{\left(\sum_{i=1}^{k} w_i\right) - \tau}{\underline{w}} \right\rfloor \leq \frac{k\,\overline{w}}{\underline{w}} \underbrace{=}_{\overline{w}=\mathcal{O}(\mathcal{P}(k))\underline{w}} \mathcal{O}(\mathcal{P}(k)). \qquad \square$$

Unfortunately, for many problems, an analytical result such as the one obtained in this section is not always possible to achieve. The aim of the following numerical section is to demonstrate that the SSA is capable of handling hard problems in the absence of theoretical performance.

5. Numerical experiments

5.1. Portfolio credit risk

We consider a portfolio credit risk setting (Glasserman and Li, 2005). Given a portfolio of k assets, the portfolio loss L is the random variable

$$L = \sum_{i=1}^{k} l_i\, X_i, \tag{8}$$

where l_i is the risk of asset $i \in \{1, \ldots, k\}$, and X_i is an indicator random variable that models the default of asset i. Under this setting (and similar to Section 4), one is generally interested in the following.

(a) *Conditional Value at Risk (CVaR).* Given a threshold (value at risk) v, calculate the conditional value at risk $c = \mathbb{E}[L \mid L \geq v]$.

(b) *Tail probability estimation.* Given the value at risk, calculate the tail probability $\mathbb{P}(L \geq v) = \mathbb{E}\left[\mathbb{1}\{L \geq v\}\right]$.

The SSA can be applied to both problems as follows. For tail probability estimation, we simply set $\varphi(\mathbf{x}) = \mathbb{1}\{\sum_{i=1}^{k} l_i\, x_i \geq v\}$. For conditional expectation, we set $\varphi(\mathbf{x}) = \sum_{i=1}^{k} l_i\, x_i$.

Note that the tail probability estimation (for which the integrand is the indicator function), is a special case of a general integration. Recall that the GS algorithm of Botev and Kroese (2012) works on indicator integrands, and thus GS is a special case of the SSA. Consequently, in this section we will investigate the more interesting (and more general) scenario of estimating an expectation conditional on a rare event.

As our working example, we consider a credit risk in a Normal Copula model and, in particular, a 21 factor model from Glasserman and Li (2005) with $1,000$ obligors.

The SSA setting is similar to the weighted component model from Section 4. We define a k-dimensional binary vector $\mathbf{x} = (x_1, \ldots, x_k)$, for which x_i stands for the ith asset default ($x_i = 1$ for default, and 0 otherwise). We take the performance function $S(\mathbf{x})$ to be the loss function (8). Then, the level sets are defined naturally by $\mathscr{X}_t = \{\mathbf{x} : S(\mathbf{x}) \geq \gamma_t\}$, (see also Section 2.2). In our experiment, we set $\gamma_0 = 0$ and $\gamma_n = 1 + \sum_{i=1}^{k} l_i$. In order to determine the remaining levels $\gamma_1, \ldots, \gamma_{n-1}$, we execute a pilot run of Algorithm 2 with $N = 1,000$ and $\rho = 0.1$. As an MCMC sampler, we use a Hit-and-Run algorithm (Kroese et al., 2011, Chapter 10, Algorithm 10.10), taking a new sample after 50 transitions.

It is important to note that despite the existence of several algorithms for estimating c, the SSA has an interesting feature, that (to the best of our knowledge) is not present in other methods. Namely, one is able to obtain an estimator for several CVaRs via a *single* SSA run. To see this, consider the estimation of $c_1, \ldots, c_s$ for $s \geq 1$. Suppose that $v_1 \leq \cdots \leq v_s$ and note that it will be sufficient to add these values to the $\{\gamma_t\}$ (as additional levels), and retain s copies of $\widehat{P}_t$ and $\widehat{Z}_t$. In particular, during the SSA execution, we will need to closely follow the γ levels, and as soon as we encounter a certain v_j for $1 \leq j \leq s$, we will start to update the corresponding values of $\widehat{P}_t^{(j)}$ and $\widehat{Z}_t^{(j)}$, in order to allow the corresponding estimation of $c_j = \mathbb{E}[L \mid L \geq v_j]$. Despite that such a procedure introduces a dependence between the obtained estimators, they still remain unbiased.

To test the above setting, we perform the experiment with a view to estimate $\{c_j\}_{1 \leq j \leq 13}$ using the following values at risk:

$$\{10000,\ 14000,\ 18000,\ 22000,\ 24000,\ 28000,\ 30000,\ 34000,\ 38000,\ 40000,\ 44000,\ 48000,\ 50000\}. \tag{9}$$

The execution of the SSA pilot run with the addition of the desired VaRs (levels) from (9) (marked in bold), yields the following level values of $(\gamma_0, \ldots, \gamma_{21})$:

$$(0,\ 788.3,\ 9616.7,\ \mathbf{10000},\ \mathbf{14000},\ \mathbf{18000},\ \mathbf{22000},\ \mathbf{24000},\ \mathbf{28000},\ \mathbf{30000},\ \mathbf{34000},\ \mathbf{38000},\ \mathbf{40000},\ \mathbf{44000},\ 47557.6,\ \mathbf{48000},\ 49347.8,\ \mathbf{50000},\ 50320.6,\ 50477.4,\ 50500,\ \infty).$$

Table 1 summarizes the results obtained by executing 1 pilot and 5 regular independent runs of the SSA. For each run, we use the parameter set that was specified for the pilot run ($N = 1,000$ and burn-in of 50). The overall execution time (for all these $R = 1+5 = 6$ independent runs) is 454 seconds. The SSA is very accurate. In particular, we obtain an RE of less than 1% for each $\widehat{c}$ while employing a very modest effort.

v	$\widehat{c}$	RE	v	$\widehat{c}$	RE
10000	1.68×10^4	0.67 %	34000	3.80×10^4	0.05 %
14000	2.09×10^4	0.51 %	38000	4.11×10^4	0.05 %
18000	2.46×10^4	0.21 %	40000	4.26×10^4	0.08 %
22000	2.82×10^4	0.19 %	44000	4.56×10^4	0.08 %
24000	2.99×10^4	0.23 %	48000	4.86×10^4	0.02 %
28000	3.32×10^4	0.21 %	50000	5.01×10^4	0.02 %
30000	3.48×10^4	0.14 %			

Table 1: The SSA results for the Normal copula credit risk model with 21 factors and 1,000 obligors.

The obtained result is especially appealing, since the corresponding estimation problem falls into rare-event setting (Glasserman, 2004). That is, a CMC estimator will not be applicable in this case.

5.2. Linear regression – non-nested models

Table 2 summarizes a dataset from Willams (1959), where observations from 42 specimens of radiata pine are considered. In particular, this data describes the maximum compression strength y_i, the density x_i, and the resin-adjusted density z_i.

Similar to (Friel and Pettitt, 2008; Han and Carlin, 2001; Chib, 1995; Bartolucci and Scaccia, 2004), we compare the following two models

$$\begin{aligned} \mathrm{M}_1: &\quad y_i = \alpha + \beta(x_i - \overline{x}) + \varepsilon_i, \quad \varepsilon_i \sim \mathsf{N}(0, \sigma^2), \\ \mathrm{M}_2: &\quad y_i = \gamma + \delta(z_i - \overline{z}) + \eta_i, \quad \eta_i \sim \mathsf{N}(0, \tau^2). \end{aligned}$$

i	y_i	x_i	z_i	i	y_i	x_i	z_i	i	y_i	x_i	z_i
1	3040	29.2	25.4	15	2250	27.5	23.8	29	1670	22.1	21.3
2	2470	24.7	22.2	16	2650	25.6	25.3	30	3310	29.2	28.5
3	3610	32.3	32.2	17	4970	34.5	34.2	31	3450	30.1	29.2
4	3480	31.3	31.0	18	2620	26.2	25.7	32	3600	31.4	31.4
5	3810	31.5	30.9	19	2900	26.7	26.4	33	2850	26.7	25.9
6	2330	24.5	23.9	20.0	1670	21.1	20.0	34	1590	22.1	21.4
7	1800	19.9	19.2	21	2540	24.1	23.9	35	3770	30.3	29.8
8	3110	27.3	27.2	22	3840	30.7	30.7	36	3850	32.0	30.6
9	3670	32.3	29.0	23	3800	32.7	32.6	37	2480	23.2	22.6
10	2310	24.0	23.9	24	4600	32.6	32.5	38	3570	30.3	30.3
11	4360	33.8	33.2	25	1900	22.1	20.8	39	2620	29.9	23.8
12	1880	21.5	21.0	26	2530	25.3	23.1	40	1890	20.8	18.4
13	3670	32.2	29.0	27	2920	30.8	29.8	41	3030	33.2	29.4
14	1740	22.5	22.0	28	4990	38.9	38.1	42	3030	28.2	28.2

Table 2: The radiata pine data-set taken from Willams (1959). The explanatory variables are the density x_i, and the resin-adjusted density z_i. The response variable y_i stands for the maximum compression strength parallel to the grain.

We wish to perform a Bayesian model comparison of M_1 and M_2 via the SSA. Similar to the above references, we choose a prior

$$\mathsf{Norm}\left(\begin{bmatrix}3000\\185\end{bmatrix}, \begin{bmatrix}10^6 & 0\\0 & 10^4\end{bmatrix}\right),$$

for both $(\alpha, \beta)^\top$ and $(\gamma, \delta)^\top$, and an inverse-gamma prior with hyperparameters $\alpha = 3$ and $\beta = (2 \cdot 300^2)^{-1}$ for both ε_i and η_i. By performing numerical integration, it was found by O'Hagan (1995), that the Bayes factor B_{21} under this setting is equal to

$$\mathrm{B}_{21} = \frac{\mathbb{P}(D \mid \mathrm{M}_2)}{\mathbb{P}(D \mid \mathrm{M}_1)} = \frac{\int \mathbb{P}\left(\gamma, \delta, \tau^2 \mid \mathrm{M}_2\right) \mathbb{P}(D \mid \gamma, \delta, \tau^2, \mathrm{M}_2)\, \mathrm{d}\gamma\, \mathrm{d}\delta\, \mathrm{d}\tau^2}{\int \mathbb{P}\left(\alpha, \beta, \sigma^2 \mid \mathrm{M}_1\right) \mathbb{P}(D \mid \alpha, \beta, \sigma^2, \mathrm{M}_1)\, \mathrm{d}\alpha\, \mathrm{d}\beta\, \mathrm{d}\sigma^2} = 4862.$$

Next, we run the SSA to obtain two estimators $\widehat{Z}_{\mathrm{M}_1}$ and $\widehat{Z}_{\mathrm{M}_2}$ for $\mathbb{P}(D \mid \mathrm{M}_1)$ and $\mathbb{P}(D \mid \mathrm{M}_2)$. In this setting our initial density f is the prior and we set our performance function S to be the likelihood function (alternatively, one could use the log-likelihood). Consequentially, the SSA approximation of the Bayes factor B_{21} is given by the ratio estimator $\widehat{\mathrm{B}}_{21} = \widehat{Z}_{\mathrm{M}_2} / \widehat{Z}_{\mathrm{M}_1}$.

Our experimental setting for the SSA algorithm is as follows. We use a sample size of $N = 10,000$ for each level set. For sampling, we use the random walk sampler for each f_t, with $\sigma^2 = 1000$. In order to benchmark the SSA performance in the sense of the closeness to the real B_{21} value, both $\widehat{Z}_{\mathrm{M}_1}$ and $\widehat{Z}_{\mathrm{M}_2}$ were estimated (via R independent replications) until the relative error for both is less than 0.5%. The levels used for M1 and M2, respectively, are:

$$\{0,\ 9.40 \times 10^{-146},\ 1.88 \times 10^{-134},\ 7.39 \times 10^{-133},\ 1.66 \times 10^{-132},\ \infty\}$$

and

$$\{0,\ 1.20 \times 10^{-141},\ 2.77 \times 10^{-130},\ 4.26 \times 10^{-129},\ 7.83 \times 10^{-129},\ \infty\}.$$

To reach the predefined 0.5% RE, the SSA stopped after 258 iterations. The average estimates are 2.5123×10^{-135} and 1.2213×10^{-131} for $\widehat{Z}_{\mathrm{M}_1}$ and $\widehat{Z}_{\mathrm{M}_2}$, respectively, $\widehat{\mathrm{B}}_{21} = \widehat{Z}_{\mathrm{M}_2}/\widehat{Z}_{\mathrm{M}_1} \approx 4861$. The associated 95% confidence interval (obtained via the Delta method) for z_2/z_1 is $(4798.8, 4923.8)$.

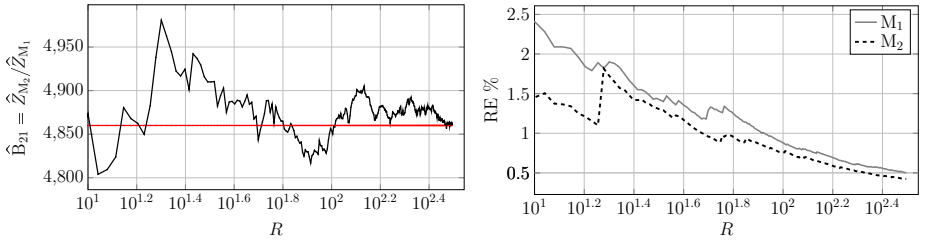

Fig. 1: The leftmost plot shows the convergence of $\widehat{\mathrm{B}}_{21}$ as a function of the SSA iteration number (R), to the true value — 4862. The rightmost plot shows the RE as a function of R for both $\widehat{Z}_{\mathrm{M}_1}$ and $\widehat{Z}_{\mathrm{M}_2}$.

In practice, however, we could stop the SSA execution much earlier. In particular, the leftmost plot in Fig. 1 clearly indicates a convergence of $\widehat{\mathrm{B}}_{21}$ to a factor that is greater than 4000 after as few as 20 iterations. That is, by the well-known Jeffrey's (Jeffrey, 1961) scale, Model 2 can be safely declared as superior since $\widehat{\mathrm{B}}_{21} \gg 10^2$, indicating decisive evidence for the second model. It is worth noting that, under the above setting, our precision result is comparable with the power-posterior results obtained

by Friel and Pettitt (2008), which outperform reversible jump MCMC for this problem.

We note that in the setting of estimating model evidence (and in Bayesian inference as in the next section), the SSA bears some similarities to Nested Sampling. Namely, the use of a population of particles, and the sampling from increasing thresholds of the likelihood function. Indeed, our tests on various problems indicate that both methods perform similarly in terms of estimator variance, however it is important to note that unlike the SSA, Nested Sampling is not unbiased. In fact, as mentioned earlier, it is not even known to be consistent when MCMC sampling is used.

5.3. Bayesian inference

Suppose that a pdf h is known up to its normalization constant, that is $h \propto L \cdot f$. For example, it is convenient to think of $L \cdot f$ as likelihood multiplied by prior, and of h as the corresponding posterior distribution. Under this setting, we are interested in the estimation of $\mathbb{E}_h[H(\mathbf{X})]$ for any $H : \mathscr{X} \to \mathbb{R}$. Recall that the generic multilevel splitting framework and in particular the SSA, provide the set of samples $\{\mathcal{Z}_t\}_{1 \le t \le n}$. These samples can be immediately used for an estimation of $\mathbb{E}_h[H(\mathbf{X})]$, via

$$\frac{\sum_{r=1}^{R}\left[\sum_{t=1}^{n}\widehat{P}_t\frac{1}{|\mathcal{Z}_t|}\sum_{\mathbf{X}\in\mathcal{Z}_t}H(\mathbf{X})L(\mathbf{X})\right]}{\sum_{r=1}^{R}\left[\sum_{t=1}^{n}\widehat{P}_t\frac{1}{|\mathcal{Z}_t|}\sum_{\mathbf{X}\in\mathcal{Z}_t}L(\mathbf{X})\right]},$$

since by the law of large numbers the above ratio converges (as $R \to \infty$) to

$$\frac{\mathbb{E}_f\left[\sum_{t=1}^{n}\widehat{P}_t\frac{1}{|\mathcal{Z}_t|}\sum_{\mathbf{X}\in\mathcal{Z}_t}H(\mathbf{X})L(\mathbf{X})\right]}{\mathbb{E}_f\left[\sum_{t=1}^{n}\widehat{P}_t\frac{1}{|\mathcal{Z}_t|}\sum_{\mathbf{X}\in\mathcal{Z}_t}L(\mathbf{X})\right]} = \frac{\mathbb{E}_f\left[H(\mathbf{X})L(\mathbf{X})\right]}{\mathbb{E}_f\left[L(\mathbf{X})\right]}$$
$$= \frac{z_h\int H(\mathbf{x})\frac{L(\mathbf{x})f(\mathbf{x})}{z_h}\mathrm{d}\mathbf{x}}{z_h\int\frac{L(\mathbf{x})f(\mathbf{x})}{z_h}\mathrm{d}\mathbf{x}} = \frac{\int H(\mathbf{x})h(\mathbf{x})\mathrm{d}\mathbf{x}}{\int h(\mathbf{x})\mathrm{d}\mathbf{x}} = \mathbb{E}_h[H(\mathbf{X})].$$

We next consider the SSA Bayesian inference applied on models M_1 and M_2 from Section 5.2. We first run the (random walk) Metropolis algorithm used by Han and Carlin (2001), and estimate the posterior

mean for each parameter. In particular, for both M_1 and M_2, we start with an initial $\boldsymbol{\theta}_0 = \left(3000, 185, 300^2\right)$, and for $t > 0$, we sample $\boldsymbol{\theta}_t$ from multivariate normal with mean $\boldsymbol{\mu} = \boldsymbol{\theta}_{t-1}$ and covariance matrix

$$\boldsymbol{\Sigma} = \begin{pmatrix} 5000 & 0 & 0 \\ 0 & 250 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

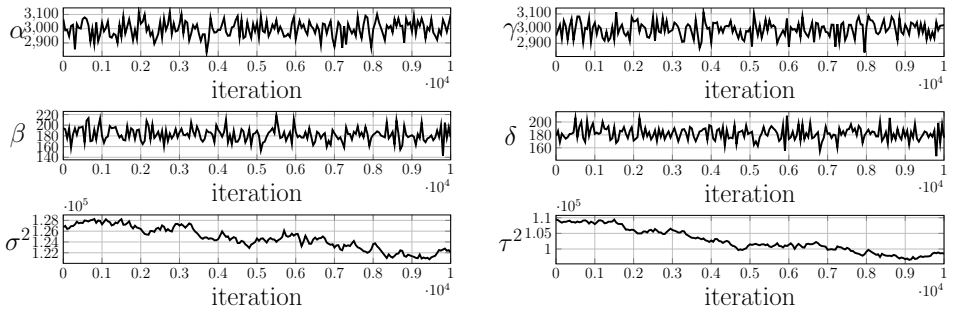

Fig. 2: The leftmost and rightmost plots correspond to the convergence of the Metropolis algorithm for the M_1 and the M_2 models, respectively. The results were obtained using $10,000$ MCMC samples after burn-in period of $90,000$.

The proposal $\boldsymbol{\theta}_t$ is then accepted with probability $\min\left\{1, \frac{p(\boldsymbol{\theta}_t)}{p(\boldsymbol{\theta}_{t-1})}\right\}$, where p denotes unnormalized posterior density function. Our experiment (see Fig. 2), indicates that the proposed Metropolis scheme has good mixing properties with respect to the slope and the intercept parameters, when applied to both M_1 and M_2. However, the mixing of the σ^2 and the τ^2 parameters does not look sufficient. Despite that we changed the $\boldsymbol{\Sigma}(3,3)$ parameter of the co-variance matrix to be $10^1, 10^2$ and 10^3 instead of 1, the typical mixing results did not change.

We next apply the SSA to the above inference problem. Using a single SSA run (with $N = 20,000$), we collected $100,000$ samples for each model. The obtained results are summarized in Table 3.

	M_1			M_2		
Algorithm	α	β	σ^2	γ	δ	τ^2
Metropolis	2993.2	184.2	1.246×10^5	2992.5	182.9	1.025×10^5
SSA	2992.3	184.2	1.136×10^5	2991.9	183.2	7.840×10^4

Table 3: Inference results for models M_1 and M_2 obtained via Metropolis and the SSA. The required CPU for each model inference is about 16 seconds for each algorithm.

The Metropolis estimates of σ^2 and τ^2 are approximately equal to 1.246×10^5 and 1.025×10^5, respectively. In contrast, the SSA estimates of these quantities are 1.125×10^5 and 7.797×10^4, respectively (see Table 3). We conjecture that the SSA estimators of σ^2 and τ^2 are superior. In order to check, we implement a Gibbs sampler ($90,000$ samples, with a $10,000$ sample burn in) and obtain the following results, supporting our suspicions.

	M_1			M_2		
Algorithm	α	β	σ^2	γ	δ	τ^2
Gibbs	2991.4	184.5	1.126×10^5	2991.8	183.3	7.792×10^4

Table 4: Inference results for models M_1 and M_2 obtained via Gibbs Sampling.

The superior performance of the SSA using a Random Walk sampler over the standard Random Walk approach is most likely due to the SSA using a collection of Markov Chain Samplers at each stage instead of a single chain, and that the SSA sampling takes place on the prior as opposed to the posterior.

5.4. Self-avoiding walks

In this section, we consider random walks of length n on the two-dimensional lattice of integers, starting from the origin. In particular, we are interested in estimating the following quantities:

(a) c_n: the number of SAWs of length n,

(b) Δ_n: the expected distance of the final SAW coordinate to the origin.

To put these SAW problems into the SSA framework, define the set of directions, $\mathscr{X} = \{\text{Left}, \text{Right}, \text{Up}, \text{Down}\}^n$, and let f be the uniform pdf on $\mathscr{X}$. Let $\xi(\mathbf{x})$ denote the final coordinate of the random walk represented by the directions vector $\mathbf{x}$. We have $c_n = \mathbb{E}_f\left[\mathbb{1}\{\mathbf{X} \text{ is SAW}\}\right]$ and $\Delta_n = \mathbb{E}_f\left[\|\xi(\mathbf{X})\| \;\middle|\; \mathbb{1}\{\mathbf{X} \text{ is SAW}\}\right]$.

Next, we let $\mathscr{X}_t \subseteq \mathscr{X}$ be the set of all directions vectors that yield a valid self-avoiding walk of length at least t, for $0 \leq t \leq n$. In addition, we define $\mathscr{Z}_t$ to be the set of all directions vectors that yield a self-avoiding walk of length (exactly) t, for $1 \leq t \leq n$. The above gives the required

partition of $\mathscr{X}$. Moreover, the simulation from $f_t(\mathbf{x}) = f\left(\mathbf{x} \mid \mathbf{x} \in \mathscr{X}_{t-1}\right)$, reduces to the uniform selection of the SAW's direction at time $1 \leq t \leq n$.

Our experimental setting for SAWs of lengths n is as follows. We set the sample size of the SSA to be $N_t = 1000$ for all $t = 1, \ldots, n$. In this experiment, we are interested in both the probability that $\mathbf{X}$ lies in $\mathscr{Z}_n$, and the expected distance of $\mathbf{X} \in \mathscr{Z}_n$ (uniformly selected) to the origin. These give us the required estimators of c_n and Δ_n, respectively. The leftmost plot of Fig. 3 summarizes a *percent error* (PE), which is defined by

$$\mathrm{PE} = 100\,\frac{\widehat{c}_n - c_n}{c_n},$$

where $\widehat{c}_n$ stands for the SSA's estimator of c_n.

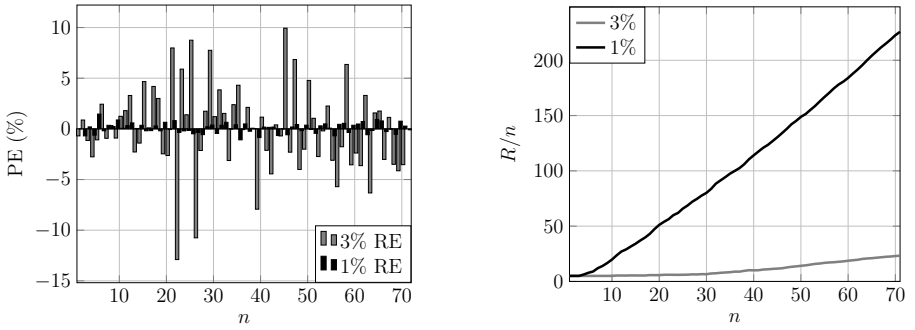

Fig. 3: The PE (leftmost plot) and the number of independent runs divided by SAW's length (rightmost plot) of the SSA as a function of SAW length n for 3% and 1% RE.

In order to explore the convergence of the SSA estimates to the true quantity of interest, the SSA was executed for a sufficient number of times to obtain 3% and 1% relative error (RE) (Rubinstein and Kroese, 2017), respectively. The exact c_n values for $n = 1, \ldots, 71$ were taken from (Guttmann and Conway, 2001; Jensen, 2004)); naturally, when we allow a smaller RE, that is, when we increase R, the estimator converges to the true value c_n, as can be observed in leftmost plot of Fig. 3. In addition, the rightmost plot of Fig. 3 shows that regardless of the RE, the required number of independent SSA runs (R) divided by SAW's length (n), is growing linearly with n.

Finally, we investigate the SSA convergence by considering the following asymptotic property (Noonan, 1998):

$$\mu = \lim_{n\to\infty} c_n^{\frac{1}{n}} \in [\underline{\mu}, \overline{\mu}] = [2.62002, 2.679192495].$$

Fig. 4 summarizes our results compared to the theoretical bound. In particular, we run the SSA to achieve the 3% RE for $1 \leq n \leq 200$. It can be clearly observed, that the estimator $\widehat{c}_n^{1/n}$ converges toward the $[\underline{\mu}, \overline{\mu}]$ interval as n grows.

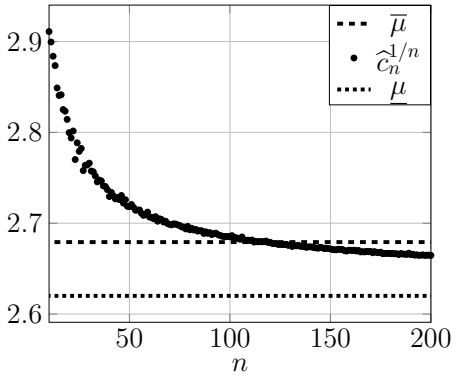

Fig. 4: The $\widehat{c}_n^{1/n}$ as a function of the SAW's length n.

6. Discussion

In this paper we described a general sequential Monte Carlo procedure for multi-dimensional integration, the SSA, and applied it to various problems from different research domains. We showed that this method belongs to a very general class of SMC algorithms and developed its theoretical foundation. The proposed SSA is relatively easy to implement and our numerical study indicates that the SSA yields good performance in practice. However, it is important to note that generally speaking, the efficiency of the SSA and similar sequential algorithms is heavily dependent on the mixing time of the corresponding Markov chains that are used for sampling. A rigorous analysis of the mixing time for different problems is thus of great interest. Opportunities for future research

include coconducting a similar analysis for other SMC algorithms, such as the resample-move method. Finally, based on our numerical study, it will be interesting to apply the SSA to a further variety of problems that do not admit to conditions of the efficiency Theorem 3.1.

Acknowledgement

This work was supported by the Australian Research Council Centre of Excellence for Mathematical & Statistical Frontiers, under grant number CE140100049.

A. Technical arguments

Proof of Theorem 2.1. Recall that $\mathscr{Z}_1, \ldots, \mathscr{Z}_n$ is a partition of the set $\mathscr{X}$, so from the law of total probability we have

$$\mathbb{E}_f\left[\varphi\left(\mathbf{X}\right)\right] = \sum_{t=1}^{n} \mathbb{E}_f\left[\varphi\left(\mathbf{X}\right) \mid \mathbf{X} \in \mathscr{Z}_t\right] \mathbb{P}_f\left(\mathbf{X} \in \mathscr{Z}_t\right).$$

By the linearity of expectation, and since $\widehat{Z} = \sum_{t=1}^{n} \widehat{Z}_t$, it will be sufficient to show that for all $t = 1, \ldots, n$, it holds that

$$\mathbb{E}\left[\widehat{Z}_t\right] = \mathbb{E}_f\left[\varphi\left(\mathbf{X}\right) \mid \mathbf{X} \in \mathscr{Z}_t\right] \mathbb{P}_f\left(\mathbf{X} \in \mathscr{Z}_t\right).$$

To see this, we need the following.

(a) Although that the samples in the $\mathcal{Z}_t$ set for $1 \leq t \leq n$ are not independent due to MCMC and splitting usage, they still have the same distribution; that is, for all t, it holds that

$$\mathbb{E}\left[\sum_{\mathbf{X} \in \mathcal{Z}_t} \varphi(\mathbf{X}) \,\middle|\, |\mathcal{Z}_t|\right] = |\mathcal{Z}_t| \, \mathbb{E}_f\left[\varphi\left(\mathbf{X}\right) \mid \mathbf{X} \in \mathcal{Z}_t\right]. \tag{10}$$

(b) From the unbiasedness of multilevel splitting (Kroese et al., 2011; Botev and Kroese, 2012), it holds for all $1 \leq t \leq n$ that

$$\mathbb{E}\left[\widehat{P}_t\right] = \mathbb{E}\left[\left(1 - \widehat{R}_t\right) \prod_{j=0}^{t-1} \widehat{R}_j\right] = \mathbb{P}_f\left(\mathbf{X} \in \mathscr{Z}_t\right). \tag{11}$$

Combining (10) and (11) with a conditioning on the cardinalities of the $\mathcal{Z}_t$ sets, we complete the proof with:

$$\begin{aligned}
\mathbb{E}\left[\widehat{Z}_t\right] &= \mathbb{E}\left[\widehat{\Phi}_t\,\widehat{P}_t\right] = \mathbb{E}\left[\widehat{P}_t \frac{1}{|\mathcal{Z}_t|}\sum_{\mathbf{X}\in\mathcal{Z}_t}\varphi(\mathbf{X})\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\widehat{P}_t \frac{1}{|\mathcal{Z}_t|}\sum_{\mathbf{X}\in\mathcal{Z}_t}\varphi(\mathbf{X})\,\middle|\,|\mathcal{Z}_0|,\ldots,|\mathcal{Z}_t|\right]\right] \\
&= \mathbb{E}\left[\widehat{P}_t \frac{1}{|\mathcal{Z}_t|}\mathbb{E}\left[\sum_{\mathbf{X}\in\mathcal{Z}_t}\varphi(\mathbf{X})\,\middle|\,|\mathcal{Z}_0|,\ldots,|\mathcal{Z}_t|\right]\right] \\
&\underbrace{=}_{(10)} \mathbb{E}\left[\widehat{P}_t \frac{1}{|\mathcal{Z}_t|}|\mathcal{Z}_t|\,\mathbb{E}_f\left[\varphi\left(\mathbf{X}\right)\mid\mathbf{X}\in\mathscr{Z}_t\right]\right] \\
&= \mathbb{E}_f\left[H\left(\mathbf{X}\right)\mid\mathbf{X}\in\mathscr{Z}_t\right]\mathbb{E}\left[\widehat{P}_t\right]\underbrace{=}_{(11)}\mathbb{E}_f\left[H\left(\mathbf{X}\right)\mid\mathbf{X}\in\mathscr{Z}_t\right]\mathbb{P}_f\left(\mathbf{X}\in\mathscr{Z}_t\right).
\end{aligned}$$

□

Proof of Theorem 3.1. The proof of this theorem consists of the following steps.

(a) In Lemma A.1, we prove that an existence of an $\left(\varepsilon, \frac{\delta}{n}\right)$-approximation to $\{z_t\}_{1\le t\le n}$ implies an existence of an (ε,δ)-approximation to $z = \sum_{t=1}^{n} z_t$.

(b) In Lemma A.2, we prove that an existence of an $\left(\frac{\varepsilon}{4}, \frac{\delta}{2n}\right)$-approximation to $\{\varphi_t\}_{1\le t\le n}$ and $\{p_t\}_{1\le t\le n}$ implies an $\left(\varepsilon, \frac{\delta}{n}\right)$-approximation existence to $\{z_t\}_{1\le t\le n}$.

(c) In Lemmas A.3, A.4 and A.5, we provide the required $\left(\frac{\varepsilon}{4}, \frac{\delta}{2n}\right)$-approximations to φ_t and p_t for $1 \le t \le n$.

LEMMA A.1. *Suppose that for all* $t = 1,\ldots,n$, *an* $\left(\varepsilon, \frac{\delta}{n}\right)$*-approximation to* z_t *exists. Then,*

$$\mathbb{P}\left(z(1-\varepsilon) \le \widehat{Z} \le z(1+\varepsilon)\right) \ge 1-\delta.$$

PROOF. From the assumption of the existence of the $\left(\varepsilon, \frac{\delta}{n}\right)$-approximation to z_t for each $1 \leq t \leq n$, we have

$$\mathbb{P}\left(\left|\widehat{Z}_t - z_t\right| \leq \varepsilon z_t\right) \geq 1 - \frac{\delta}{n}, \quad \text{and} \quad \mathbb{P}\left(\left|\widehat{Z}_t - z_t\right| > \varepsilon z_t\right) < \frac{\delta}{n}.$$

By using the Boole's inequality (union bound), we arrive at

$$\mathbb{P}\left(\exists\, t \;:\; \left|\widehat{Z}_t - z_t\right| > \varepsilon z_t\right) \leq \sum_{t=1}^{n} \mathbb{P}\left(\left|\widehat{Z}_t - z_t\right| > \varepsilon z_t\right) < n\frac{\delta}{n} = \delta,$$

that is, it holds for all $t = 1, \ldots, n$, that

$$\mathbb{P}\left(\forall\, t \;:\; \left|\widehat{Z}_t - z_t\right| \leq \varepsilon z_t\right) = 1 - \mathbb{P}\left(\exists\, t \;:\; \left|\widehat{Z}_t - z_t\right| > \varepsilon c_t\right) \geq 1 - \delta,$$

and hence,

$$\mathbb{P}\left((1-\varepsilon)\sum_{t=1}^{n} z_t \leq \sum_{t=1}^{n} \widehat{Z}_t \leq (1+\varepsilon)\sum_{t=1}^{n} z_t\right) = \mathbb{P}\left(\widehat{Z} \in\leq z(1 \pm \varepsilon)\right) \geq 1 - \delta.$$

□

LEMMA A.2. *Suppose that for all $t = 1, \ldots, n$, there exists an $\left(\frac{\varepsilon}{4}, \frac{\delta}{2n}\right)$-approximation to φ_t and p_t. Then,*

$$\mathbb{P}\left(z_t(1-\varepsilon) \leq \widehat{Z}_t \leq z_t(1+\varepsilon)\right) \geq 1 - \frac{\delta}{n} \quad \textit{for all } t = 1, \ldots, n.$$

PROOF. By assuming an existence of $\left(\frac{\varepsilon}{4}, \frac{\delta}{2n}\right)$-approximation to φ_t and p_t, namely:

$$\mathbb{P}\left(\left|\frac{\widehat{\Phi}_t}{h_t} - 1\right| \leq \varepsilon/4\right) \geq 1 - \delta/2n, \quad \text{and} \quad \mathbb{P}\left(\left|\frac{\widehat{P}_t}{p_t} - 1\right| \leq \varepsilon/4\right) \geq 1 - \delta/2n,$$

and combining it with the union bound, we arrive at

$$\mathbb{P}\left(1 - \varepsilon \underbrace{\leq}_{(*)} \left(1 - \frac{\varepsilon/2}{2}\right)^2 \leq \frac{\widehat{\Phi}_t \widehat{P}_t}{\varphi_t\, p_t} \leq \left(1 + \frac{\varepsilon/2}{2}\right)^2 \underbrace{\leq}_{(*)} 1 + \varepsilon\right) \geq 1 - \delta/n,$$

where $(*)$ follows from the fact that for any $0 < |\varepsilon| < 1$ and $n \in \mathbb{N}$ we have

$$1 - \varepsilon \leq \left(1 - \frac{\varepsilon/2}{n}\right)^n \quad \text{and} \quad \left(1 + \frac{\varepsilon/2}{n}\right)^n \leq 1 + \varepsilon. \tag{12}$$

To see that (12) holds, note that by using exponential inequalities from Bullen (1998), we have that $\left(1 - \frac{\varepsilon/2}{n}\right)^n \geq 1 - \frac{\varepsilon}{2} \geq 1 - \varepsilon$. In addition, it holds that $|\mathrm{e}^{\varepsilon} - 1| < 7\varepsilon/4$, and hence:

$$\left(1 + \frac{\varepsilon/2}{n}\right)^n \leq \mathrm{e}^{\varepsilon/2} \leq 1 + \frac{7(\varepsilon/2)}{4} \leq 1 + \varepsilon. \qquad \square$$

To complete the proof of Theorem 3.1, we need to provide $\left(\frac{\varepsilon}{4}, \frac{\delta}{2n}\right)$-approximations to both φ_t and p_t. However, at this stage we have to take into account a specific splitting strategy, since the SSA sample size bounds depend on the latter. Here we examine the independent setting, for which the samples in each $\{\mathcal{X}_t\}_{1 \leq t \leq n}$ set are independent. That is, we use multiple runs of the SSA at each stage $(t = 1, \ldots, n)$ of the algorithm execution. See Remark 2.1 for further details.

Remark A.1 (Alternative splitting mechanisms). Our choice of applying the independent setting does not impose a serious limitation from theoretical time-complexity point of view, and is more convenient for an analysis. In particular, when dealing with the independent setting, we can apply a powerful concentration inequalities (Chernoff, 1952; Hoeffding, 1963). Alternatively, one could compromise the independence, and use Hoeffding-type inequalities for dependent random variables, such as the ones proposed in (Glynn and Ormoneit, 2002; Paulin, 2015).

The key to obtaining the desired approximation results is summarized in Lemma A.3.

Lemma A.3. *Let $X \sim \widehat{\pi}$ be a strictly positive univariate random variable such that $a \leq X \leq b$, and let $X_1, \ldots, X_m$ be its independent realizations. Then, provided that*

$$\|\widehat{\pi} - \pi\|_{\mathrm{TV}} \leq \frac{\varepsilon\, a}{4(b-a)}, \quad \textit{and} \quad m \geq \frac{(b - a^2)\ln(2/\delta)}{2(\varepsilon/4)^2 a^2},$$

it holds that:

$$\mathbb{P}\left((1-\varepsilon)\mathbb{E}_{\pi}[X] \leq \frac{1}{m}\sum_{i=1}^{m} X_i \leq (1+\varepsilon)\mathbb{E}_{\pi}[X]\right) \geq 1 - \delta.$$

PROOF. Recall that

$$\|\widehat{\pi} - \pi\|_{\mathrm{TV}} = \frac{1}{b-a} \sup_{\varphi\,:\,\mathbb{R}\to[a,b]} \left| \int \varphi(x)\,\widehat{\pi}(\mathrm{d}x) - \int \varphi(x)\,\pi(\mathrm{d}x) \right|,$$

for any function $\varphi : \mathbb{R} \to [a, b]$, (Proposition 3 in Roberts et al. (2004)). Hence,

$$\mathbb{E}_{\pi}[X] - (b-a)\frac{\varepsilon\, a}{4(b-a)} \le \mathbb{E}_{\widehat{\pi}}[X] \le \mathbb{E}_{\pi}[X] + (b-a)\frac{\varepsilon\, a}{4(b-a)}.$$

Combining this with the fact that $X \ge a$, we arrive at

$$1 - \frac{\varepsilon}{4} \le 1 - \frac{\varepsilon\, a}{4\mathbb{E}_{\pi}[X]} \le \frac{\mathbb{E}_{\widehat{\pi}}[X]}{\mathbb{E}_{\pi}[X]} \le 1 + \frac{\varepsilon\, a}{4\mathbb{E}_{\pi}[X]} \le 1 + \frac{\varepsilon}{4}. \tag{13}$$

Next, since

$$\mathbb{E}_{\widehat{\pi}}\left[\frac{1}{m}\sum_{i=1}^{m} X_i\right] = \mathbb{E}_{\widehat{\pi}}[X],$$

we can apply the Hoeffding (1963) inequality, to obtain

$$\mathbb{P}\left(1 - \frac{\varepsilon}{4} \le \frac{\frac{1}{m}\sum_{i=1}^{m} X_i}{\mathbb{E}_{\widehat{\pi}}[X]} \le 1 + \frac{\varepsilon}{4}\right) \ge 1 - \delta, \tag{14}$$

for

$$m = \frac{(b-a)^2 \ln(2/\delta)}{2(\varepsilon/4)^2 \left(\mathbb{E}_{\widehat{\pi}}[X]\right)^2} \ge \frac{(b-a)^2 \ln(2/\delta)}{2(\varepsilon/4)^2 a^2}.$$

Finally, we complete the proof by combining (13) and (14), to obtain:

$$\mathbb{P}\left(1+\varepsilon \le \left(1 - \frac{\varepsilon/2}{2}\right)^2 \le \frac{\mathbb{E}_{\widehat{\pi}}[X]}{\mathbb{E}_{\pi}[X]} \frac{\frac{1}{m}\sum_{i=1}^{m} X_i}{\mathbb{E}_{\widehat{\pi}}[X]} \le \left(1 + \frac{\varepsilon/2}{2}\right)^2 \le 1+\varepsilon\right)$$

$$= \mathbb{P}\left((1-\varepsilon)\mathbb{E}_{\pi}[X] \le \frac{1}{m}\sum_{i=1}^{m} X_i \le (1+\varepsilon)\mathbb{E}_{\pi}[X]\right) \ge 1 - \delta. \qquad \square$$

REMARK A.2 (LEMMA A.3 FOR BINARY RANDOM VARIABLES). For a binary random variable $X \in \{0, 1\}$, with a known lower bound on its mean, Lemma A.3 can be strengthened via the usage of Chernoff

(1952) bound instead of the Hoeffding (1963) inequality. In particular, the following holds.

Let $X \sim \widehat{\pi}(x)$ be a binary random variable and let $X_1, \ldots, X_m$ be its independent realizations. Then, provided that $\mathbb{E}_{\widehat{\pi}}[X] \geq \mathbb{E}'_{\widehat{\pi}}[X]$,

$$\|\widehat{\pi} - \pi\|_{\mathrm{TV}} \leq \frac{\varepsilon\, \mathbb{E}'_{\widehat{\pi}}[X]}{4}, \quad \text{and} \quad m \geq \frac{3\ln(2/\delta)}{(\varepsilon/4)^2(\mathbb{E}'_{\widehat{\pi}}[X])^2},$$

it holds that:

$$\mathbb{P}\left((1-\varepsilon)\mathbb{E}_{\pi}[X] \leq \frac{1}{m}\sum_{i=1}^{m} X_i \leq (1+\varepsilon)\mathbb{E}_{\pi}[X]\right) \geq 1-\delta.$$

The corresponding proof is almost identical to the one presented in Lemma A.3. The major difference is the bound on the sample size in (14), which is achieved via the Chernoff bound from (Mitzenmacher and Upfal, 2005, Theorem 10.1) instead of Hoeffding's inequality.

LEMMA A.4. *Suppose that* $a_t = \min_{\mathbf{x}\in\mathscr{X}_t}\{\varphi(\mathbf{x})\}$, $b_t = \max_{\mathbf{x}\in\mathscr{X}_t}\{\varphi(\mathbf{x})\}$ *for all* $t = 1, \ldots, n$. *Then, provided that the samples in the* $\mathcal{Z}_t$ *set are independent, and are distributed according to* $\widehat{\nu}_t$ *such that*

$$\|\widehat{\nu}_t - \nu_t\|_{\mathrm{TV}} \leq \frac{\varepsilon\, a_t}{16(b_t - a_t)}, \quad \textit{and} \quad |\mathcal{Z}_t| \geq \frac{128(b_t-a_t)^2\ln(4n/\delta)}{\varepsilon^2 a_t^2},$$

then $\widehat{\Phi}_t = |\mathcal{Z}_t|^{-1}\sum_{\mathbf{X}\in\mathcal{Z}_t}\varphi(\mathbf{X})$ *is an* $\left(\frac{\varepsilon}{4}, \frac{\delta}{2n}\right)$*-approximation to* φ_t.

PROOF. The proof is an immediate consequence of Lemma A.3. In particular, note that

$$\|\widehat{\nu}_t - \nu_t\|_{\mathrm{TV}} \leq \frac{\frac{\varepsilon}{4}\, a_t}{4(b_t - a_t)} = \frac{\varepsilon\, a_t}{16(b_t - a_t)},$$

and that

$$|\mathcal{Z}_t| \geq \frac{(b_t-a_t)^2)\ln(2/\frac{\delta}{2n})}{2(\frac{\varepsilon}{4}/4)^2 a^2} = \frac{128(b_t-a_t)^2\ln(4n/\delta)}{\varepsilon^2 a_t^2}. \qquad \square$$

LEMMA A.5. *Suppose that the samples in the* $\mathcal{X}_t$ *set are independent, and are distributed according to* $\widehat{\mu}_t$, *such that*

$$\|\widehat{\mu}_t - \mu_t\|_{\mathrm{TV}} \leq \frac{\varepsilon\, \underline{r}_t}{32n}, \quad \textit{and} \quad |\mathcal{X}_t| \geq \frac{3072\, n^2\ln(4n^2/\delta)}{\varepsilon^2 \underline{r}_t^2},$$

where $\underline{r}_t = \min\{r_t, 1-r_t\}$ *for* $1 \le t \le n$. *Then,* $\widehat{P}_t$ *is an* $\left(\frac{\varepsilon}{4}, \frac{\delta}{2n}\right)$*-approximation to* p_t.

PROOF. Recall that $\widehat{P}_t = \left(1-\widehat{R}_t\right)\prod_{j=0}^{t-1}\widehat{R}_j$ for $t = 1, \dots, n$. Again, by combining the union bound with (12), we conclude that the desired approximation to p_t can be obtained by deriving the $\left(\frac{\varepsilon}{8n}, \frac{\delta}{2n^2}\right)$-approximations for each r_t and $1-r_t$. In this case, the probability that for all $t = 1, \dots, n$, $\widehat{R}_t/r_t$ satisfies $1-\varepsilon/8n \le \widehat{R}_t/r_t \le 1+\varepsilon/8n$ is at least $1-\delta/2n^2$. The same holds for $(1-\widehat{R}_t)/(1-r_t)$, and thus, we arrive at:

$$\mathbb{P}\left(1-\varepsilon/4 \le \left(1-\frac{\frac{\varepsilon}{4}/2}{n}\right)^n \le \frac{\widehat{P}_t}{p_t} \le \left(1+\frac{\frac{\varepsilon}{4}/2}{n}\right)^n \le 1+\varepsilon/4\right) \ge 1-\delta/2n.$$

The bounds for each $\widehat{R}_t$ and $(1-\widehat{R}_t)$ are easily achieved via Remark A.2. In particular, it is not very hard to verify that in order to get an $\left(\frac{\varepsilon}{8n}, \frac{\delta}{2n^2}\right)$-approximation, it is sufficient to take

$$\|\widehat{\mu}_t - \mu_t\|_{\mathrm{TV}} \le \frac{\frac{\varepsilon}{8n}\,\underline{r}_t}{4} = \frac{\varepsilon\,\underline{r}_t}{32n},$$

and

$$|\mathcal{X}_t| \ge \frac{3\ln\left(2/\frac{\delta}{2n^2}\right)}{\left(\frac{\varepsilon}{8n}/4\right)^2} = \frac{3072\,n^2\ln(4n^2/\delta)}{\varepsilon^2\underline{r}_t^2}. \qquad \square$$

LEMMA A.6. *Suppose without loss of generality that* $\mathbf{w} = (w_1, \dots, w_k)$ *satisfies* $w_1 \le w_2 \le \cdots \le w_k$, *that is* $\underline{w} = w_1$. *Then, for* $\mathscr{X}_b$ *and* $\mathscr{X}_{b-w_1}$ *sets defined via* (7), *it holds that:*

$$r = \frac{|\mathscr{X}_{b-w_1}|}{|\mathscr{X}_b|} \ge \frac{1}{k+1}.$$

PROOF. For any $b \in \mathbb{R}$ and $\mathbf{x} = (x_1, \dots, x_k)$, define a partition of $\mathscr{X}_b$ via

$$\mathscr{X}_b^{(w_1)} = \{\mathbf{x} \in \mathscr{X}_b \,:\, x_1 = 1\}, \quad \text{and} \quad \mathscr{X}_b^{(-w_1)} = \{\mathbf{x} \in \mathscr{X}_b \,:\, x_1 = 0\}.$$

Then, the following holds.

(a) For any $\mathbf{x} \in \mathscr{X}_b^{(w_1)}$, replace $x_1 = 1$ with $x_1 = 0$, and note that the resulting vector is in $\mathscr{X}_{b-w_1}^{(-w_1)}$ set, since its performance is at most

$b-w_1$, that is $\left|\mathscr{X}_b^{(w_1)}\right| \leq \left|\mathscr{X}_{b-w_1}^{(-w_1)}\right|$. Similarly, for any $\mathbf{x} \in \mathscr{X}_{b-w_1}^{(-w_1)}$, setting $x_1 = 1$ instead of $x_1 = 0$, results in a vector which belongs to the $\mathscr{X}_b^{(w_1)}$ set. That is:

$$\left|\mathscr{X}_b^{(w_1)}\right| = \left|\mathscr{X}_{b-w_1}^{(-w_1)}\right|. \tag{15}$$

(b) For any $\mathbf{x} \in \mathscr{X}_b^{(w_1)}$, replace $x_1 = 1$ with $x_1 = 0$ and note that the resulting vector is now in the $\mathscr{X}_b^{(-w_1)}$ set, that is $|\mathscr{X}_b^{(w_1)}| \leq |\mathscr{X}_b^{(-w_1)}|$. In addition, for any $\mathbf{x} \in \mathscr{X}_b^{(-w_1)}$, there are at most $k-1$ possibilities to replace $\mathbf{x}$'s non-zero entry with zero and set $x_1 = 1$, such that the result will be in the $\mathscr{X}_b^{(w_1)}$ set. That is, $\left|\mathscr{X}_b^{(-w_1)}\right| \leq (k-1)\left|\mathscr{X}_b^{(w_1)}\right| + 1$, (where $+1$ stands for the vector of zeros), and we arrive at

$$\left|\mathscr{X}_b^{(w_1)}\right| \leq \left|\mathscr{X}_b^{(-w_1)}\right| \leq (k-1)\left|\mathscr{X}_b^{(w_1)}\right| + 1 \leq \; k\left|\mathscr{X}_b^{(w_1)}\right|. \tag{16}$$

Combining (15) and (16), we complete the proof by noting that

$$\frac{|\mathscr{X}_{b-w_1}|}{|\mathscr{X}_b|} = \frac{\left|\mathscr{X}_{b-w_1}^{(w_1)}\right| + \left|\mathscr{X}_{b-w_1}^{(-w_1)}\right|}{\left|\mathscr{X}_b^{(w_1)}\right| + \left|\mathscr{X}_b^{(-w_1)}\right|} \geq \frac{\left|\mathscr{X}_{b-w_1}^{(-w_1)}\right|}{\left|\mathscr{X}_b^{(w_1)}\right| + \left|\mathscr{X}_b^{(-w_1)}\right|}$$
$$\underbrace{=}_{(15),(16)} \frac{\left|\mathscr{X}_b^{(w_1)}\right|}{\left|\mathscr{X}_b^{(w_1)}\right| + k\left|\mathscr{X}_b^{(w_1)}\right|} = \frac{1}{k+1}.$$

□

References

Andrieu, C., A. Doucet, and R. Holenstein (2010). Particle Markov Chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 72*(3), 269–342.

Asmussen, S. and P. W. Glynn (2007). *Stochastic Simulation: Algorithms and Analysis*. Applications of Mathematics. Springer Science and Business Media, LLC.

Bartolucci, F. and L. Scaccia (2004). A new approach for estimating the Bayes factor. Technical report, University di Perugia.

Botev, Z. I. and D. P. Kroese (2012). Efficient Monte Carlo simulation via the Generalized Splitting method. *Statistics and Computing 22*, 1–16.

Bullen, P. (1998). *A Dictionary of Inequalities*. Monographs and Research Notes in Mathematics. Oxfordshire: Taylor & Francis.

Chen, Y., J. Xie, and J. S. Liu (2005). Stopping-time resampling for sequential Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67*(2), 199–217.

Chernoff, H. (1952, 12). A Measure of Asymptotic Efficiency for Tests of a Hypothesis Based on the sum of Observations. *Ann. Math. Statist. 23*(4), 493–507.

Chib, S. (1995). Marginal likelihood from the Gibbs output. *Journal of the American Statistical Association 90*(432), 1313–1321.

Chopin, N. and C. P. Robert (2010). Properties of nested sampling. *Biometrika 97*(3), 741–755.

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

Evans, M. (2007). Bayesian statistics 8. Chapter Discussion of Nested sampling for Bayesian computations by John Skilling, pp. 491–524. New York: Oxford University Press.

Feroz, F. and J. Skilling (2013). Exploring multi-modal distributions with nested sampling. *AIP Conference Proceedings 1553*(1), 106–113.

Forsythe, G. E., M. A. Malcolm, and C. B. Moler (1977). *Computer methods for mathematical computations*. Prentice-Hall series in automatic computation. Englewood Cliffs (N.J.): Prentice-Hall.

Friedman, H. (1980). A consistent Fubini-Tonelli theorem for nonmeasurable functions. *Illinois J. Math. 24*(3), 390–395.

Friel, N. and A. N. Pettitt (2008, July). Marginal likelihood estimation via power posteriors. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 70*(3), 589–607.

Gelman, A., J. B. Carlin, H. S. Stern, and D. B. Rubin (2003, July). *Bayesian Data Analysis* (3 ed.). Oxfordshire: Taylor & Francis.

Gilks, W. R. and C. Berzuini (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.

Glasserman, P. (2004). *Monte Carlo methods in financial engineering*. Applications of mathematics. New York: Springer. Permire parution en dition broche 2010.

Glasserman, P. and J. Li (2005). Importance sampling for portfolio credit risk. *Management Science 51*(11), 1643–1656.

Glynn, P. W. and D. Ormoneit (2002, January). Hoeffding's inequality for uniformly ergodic Markov chains. *Statistics & Probability Letters 56*(2), 143–146.

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

Guttmann, A. and A. Conway (2001). Square lattice self-avoiding walks and polygons. *Annals of Combinatorics 5*(3), 319–345.

Han, C. and B. P. Carlin (2001). Markov Chain Monte Carlo methods for computing Bayes factors: A comparative review. *Journal of the American Statistical Association 96*, 1122–1132.

Heiss, F. and V. Winschel (2008). Likelihood approximation by numerical integration on sparse grids. *Journal of Econometrics 144*(1), 62–80.

Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association 58*(301), 13–30.

Hooper, M. (2013). Richard Price, Bayes' Theorem, and God. *Significance 10*(1), 36–39.

Jeffrey, H. (1961). *Regression Analysis* (3 ed.). Oxford, England: Oxford.

Jensen, I. (2004). Enumeration of self-avoiding walks on the square lattice. *Journal of Physics A: Mathematical and General 37*(21), 5503.

Jerrum, M. and A. Sinclair (1996). The Markov chain Monte Carlo method: an approach to approximate counting and integration. In D. Hochbaum (Ed.), *Approximation Algorithms for NP-hard Problems*, Boston, pp. 482–520. PWS Publishing.

Jerrum, M., L. G. Valiant, and V. V. Vazirani (1986). Random Generation of Combinatorial Structures from a Uniform Distribution. *Theor. Comput. Sci. 43*, 169–188.

Kahn, H. and T. E. Harris (1951). Estimation of particle Transmission by Random Sampling. *National Bureau of Standards Applied Mathematics Series 12*, 27–30.

Koller, D. and N. Friedman (2009). *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. Cambridge, Massachusetts: The MIT Press.

Kroese, D. P., T. Taimre, and Z. I. Botev (2011). *Handbook of Monte Carlo methods*. New York: John Wiley and Sons.

Lee, P. M. (2004). *Bayesian Statistics - An Introduction*. London: Arnold.

Levin, D. A., Y. Peres, and E. L. Wilmer (2009). *Markov chains and mixing times*. Providence, R.I. American Mathematical Society. With a chapter on coupling from the past by James G. Propp and David B. Wilson.

McGrayne, S. (2011). *The Theory that Would Not Die: How Bayes' Rule Cracked the Enigma Code, Hunted Down Russian Submarines, & Emerged Triumphant from Two Centuries of Controversy*. London: Yale University Press.

Mitzenmacher, M. and E. Upfal (2005). *Probability and computing : randomized algorithms and probabilistic analysis*. New York: Cambridge University Press.

Morokoff, W. J. and R. E. Caflisch (1995). Quasi-Monte Carlo Integration. *Journal of Computational Physics 122*(2), 218–230.

Morris, B. and A. Sinclair (2004). Random walks on truncated cubes and sampling 0-1 knapsack solutions. *SIAM Journal on Computing 34*(1), 195–226.

Murray, I., D. J. C. MacKay, Z. Ghahramani, and J. Skilling (2005). Nested sampling for Potts models. pp. 947–954.

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

Newman, M. and G. Barkema (1999). *Monte Carlo Methods in Statistical Physics*. Oxford, New York: Clarendon Press.

Noonan, J. (1998). New upper bounds for the connective constants of self-avoiding walks. *Journal of Statistical Physics 91*(5), 871–888.

O'Hagan, A. (1991). Bayes-Hermite quadrature. *Journal of Statistical Planning and Inference 29*(3), 245–260.

O'Hagan, A. (1995). Fractional Bayes factors for model comparison. *Journal of the Royal Statistical Society. Series B (Statistical Methodology) 57*(1), 99–138.

Paulin, D. (2015). Concentration inequalities for Markov chains by marton couplings and spectral methods. *Electron. J. Probab. 20*, 32 pp.

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

Roberts, G. O., J. S. Rosenthal, et al. (2004). General state space Markov chains and MCMC algorithms. *Probability Surveys 1*, 20–71.

Rubinstein, R. Y. and D. P. Kroese (2017). *Simulation and the Monte Carlo Method* (3 ed.). New York: John Wiley & Sons.

Rubinstein, R. Y., A. Ridder, and R. Vaisman (2013). *Fast Sequential Monte Carlo Methods for Counting and Optimization*. New York: John Wiley & Sons.

Russell, S. and P. Norvig (2009). *Artificial Intelligence: A Modern Approach* (3 ed.). Englewood Cliffs (N.J.): Prentice Hall.

Skilling, J. (2006, 12). Nested sampling for general Bayesian computation. *Bayesian Anal. 1*(4), 833–859.

Vaisman, R., D. P. Kroese, and I. B. Gertsbakh (2016). Splitting sequential Monte Carlo for efficient unreliability estimation of highly reliable networks. *Structural Safety 63*, 1 – 10.

Valiant, L. G. (1979). The complexity of enumeration and reliability problems. *SIAM Journal on Computing 8*(3), 410–421.

Willams, E. (1959). *Regression Analysis*. New York: John Wiley & Sons.