Subsampling sequential Monte Carlo for static Bayesian models

David Gunawan1,4 · Khue-Dung Dang2,4 · Matias Quiroz2,4,5 · Robert Kohn3,4 · Minh-Ngoc Tran4,6

Received: 1 May 2019 / Accepted: 27 August 2020 / Published online: 9 September 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract
We show how to speed up sequential Monte Carlo (SMC) for Bayesian inference in large data problems by data subsampling. SMC sequentially updates a cloud of particles through a sequence of distributions, beginning with a distribution that is easy to sample from such as the prior and ending with the posterior distribution. Each update of the particle cloud consists of three steps: reweighting, resampling, and moving. In the move step, each particle is moved using a Markov kernel; this is typically the most computationally expensive part, particularly when the dataset is large. It is crucial to have an efficient move step to ensure particle diversity. Our article makes two important contributions. First, in order to speed up the SMC computation, we use an approximately unbiased and efficient annealed likelihood estimator based on data subsampling. The subsampling approach is more memory efficient than the corresponding full data SMC, which is an advantage for parallel computation. Second, we use a Metropolis within Gibbs kernel with two conditional updates. A Hamiltonian Monte Carlo update makes distant moves for the model parameters, and a block pseudo-marginal proposal is used for the particles corresponding to the auxiliary variables for the data subsampling. We demonstrate both the usefulness and limitations of the methodology for estimating four generalized linear models and a generalized additive model with large datasets.

Keywords Hamiltonian Monte Carlo · Large datasets · Likelihood annealing

1 Introduction

The aim of Bayesian inference is to obtain the posterior distribution of unknown parameters, and in particular the posterior expectations of functions of the parameters. This is usually done by estimating the expectation using samples from the posterior distribution. Exact approaches such as Markov Chain Monte Carlo (MCMC) (Brooks et al. 2011) have mostly been used for sampling from complex posterior distributions. However, MCMC methods have some notable drawbacks and limitations. One drawback, often overlooked by practitioners when fitting complex models, is the failure to converge caused by poorly mixing chains. While Hamiltonian Monte Carlo (Neal 2011, HMC) is a remedy in many cases, it can be notoriously difficult to tune. Limitations of MCMC methods include the difficulties of assessing convergence, parallelizing the computation, and estimating the marginal likelihood efficiently from MCMC output, the latter being useful for model selection (Kass and Raftery 1995). Sequential Monte Carlo methods (see Doucet et al. 2001 for an introductory overview) provide an alternative exact simulation approach to MCMC methods and overcome some of their drawbacks. Moreover, in contrast to MCMC methods, SMC can provide online updates of the parameters as data is collected, which is particularly useful for dynamic (time-varying parameters) models. SMC is also useful for static (non-time-varying parameters) models (Chopin 2002; Del Moral et al. 2006), and, in such cases, can explore multi-modal posterior distributions more easily than MCMC. Our method can be applied to static models for which the likelihood is analytically tractable, and can be written as a product where each term depends on a unique piece of data information.
Despite its advantages, SMC is used much less frequently than MCMC for static models. A possible explanation is that, while amenable to computer parallelization, it is still very computationally expensive, particularly for large datasets. Another obstacle caused by large datasets is that they prevent efficient computer parallelization of SMC, as the full dataset needs to be available for each worker which is infeasible as it consumes too much random-access memory (RAM).

We propose an efficient data subsampling approach which significantly reduces both the computational cost of the algorithm and the memory requirements when parallelizing: see Sect. 3.6 for a detailed explanation of the latter. Our approach builds on the methods previously developed for Subsampling MCMC (Quiroz et al. 2019; Dang et al. 2019) and places them within the SMC framework. See Quiroz et al. (2018b) for an introduction to Subsampling MCMC.

In the Bayesian context, SMC moves a cloud of particles through a sequence of distributions, with the initial distribution both easy to sample from and to evaluate, while the final distribution is the posterior distribution. The cloud of particles at step \( p \) is an estimate of the \( p \)th distribution in the sequence. The particles consist of the unknown parameters and any additional latent variables that are part of the model. The evolution of the particle cloud from one step to another consists of three steps: reweighting, resampling and moving. Of these, the first two steps are common to all SMC schemes and are straightforward. The move step is the most expensive and helps ensure that the particle cloud is a good approximation to its target distribution.

To the best of our knowledge, data subsampling has not been explored in SMC. Although Wang et al. (2020) call their algorithm Subsampling SMC, their approach is different to ours as they combine data annealing and likelihood annealing, whereas we use data subsampling to estimate the likelihood. In particular, data annealing requires handling all the data, whereas the data subsampling approach only deals with a small fraction of the data at each stage. Specifically, we consider a likelihood annealing approach where the annealed likelihood is efficiently estimated using an approximately unbiased estimator. Likelihood estimates for SMC in a non-subsampling context are used in Duan and Fulop (2015), who estimate the likelihood unbiasedly using a particle filter in a state space context. However, Duan and Fulop (2015) use a random walk MCMC kernel for the move step of the model parameters, which is inefficient in high dimensions.

The literature has focused on accelerating SMC algorithms by designing efficient MCMC kernels for the move step to obtain particle diversity. Efficiency here means the ability of the MCMC kernel to generate distant proposals which have a high probability of being accepted. The advantage of an efficient move step is that few iterations of the kernel are needed, making computation cheaper. Various approaches exist to achieve this. For example, adaptive SMC tunes the parameters of the kernel adaptively to improve its efficiency (Jasra et al. 2011; Fearnhead and Taylor 2013; Buchholz et al. 2018). South et al. (2019) use SMC with a flexible copula-based independent proposal, while Sim et al. (2012) and South et al. (2017) use derivatives to construct efficient proposals through the Metropolis adjusted Langevin algorithm (Roberts and Stramer 2002, MALA). It is now well known that the MALA proposal is a special case of the more general proposal utilizing Hamiltonian dynamics proposed in Duane et al. (1987) (see Neal 2011; Betancourt 2017 for an introduction to HMC). Although South et al. (2017) mention HMC in their introduction, they only consider MALA in their paper and use neural networks to adaptively choose the tuning parameters. Daviet (2016) considers HMC proposals for particle diversity. However, HMC is very slow for very large datasets and therefore this approach does not scale well in the number of observations.

We propose data subsampling to achieve scalability in the number of observations and HMC Markov move steps to achieve particle diversity. Section 3.6 shows that data subsampling lowers the memory requirements of the algorithm, making it possible to parallelize the computing on very large datasets. Our framework combines that of Duan and Fulop (2015) for carrying out SMC with an estimated likelihood, Quiroz et al. (2019) for estimating the likelihood and controlling the error in the target density and Dang et al. (2019) for constructing efficient proposals for high-dimensional targets in a subsampling context.

The rest of the article is organized as follows. Section 2 reviews SMC for static models. Section 3 outlines the methodology. Section 4 applies the methodology in a variety of settings for simulated data. Section 5 presents an application of our method in model selection for a real dataset. Section 6 concludes.

### 2 Sequential Monte Carlo

#### 2.1 SMC for static Bayesian models

Denote the observed data \( y = (y_1^T, \ldots, y_n^T)^T \), with \( y_k \in \mathbb{Y} \subset \mathbb{R}^d \), where \( \mathbb{R}^m \) is an \( m \)-dimensional Euclidean space. Let \( \theta \) be the vector of unknown parameters, \( \theta \in \Theta \subset \mathbb{R}^{k} \), with \( p(\theta) \) and \( p(y|\theta) \) the prior and likelihood. In Bayesian inference, the uncertainty about \( \theta \) is specified by the posterior density \( \pi(\theta) \), which by Bayes’ theorem is

\[
\pi(\theta) = \frac{p(\theta) p(y|\theta)}{p(y)},
\]

here \( p(y) = \int_\Theta p(y|\theta) p(\theta) \, d\theta \) is the marginal likelihood which is often used for Bayesian model selection.
An important problem in Bayesian inference is estimating the posterior expectation of a function $\varphi$ of $\theta$,

$$
\mathbb{E}_\pi (\varphi(\theta)) = \int_\Theta \varphi(\theta) \pi(\theta) \, d\theta.
$$

(2)

In simulation-based inference, this is typically achieved by sampling from (1) and computing (2) by Monte Carlo integration. Another important problem is computing the marginal likelihood in (1). However, it is well known that standard Monte Carlo integration is very inefficient for this task.

SMC (Doucet et al. 2001; Del Moral et al. 2006) is a powerful approach for computing the posterior distribution and the marginal likelihood. Likelihood tempered SMC specifies a sequence of $P$ densities, connecting the prior density $p(\theta)$ to the posterior density $\pi(\theta)$ in (1). The sequence is obtained through temperature annealing (Neal 2001), in which the likelihood is tempered as $p(\eta | \theta)^{a_p}$, with $a_0 = 0 < a_1 < \cdots < a_P = 1$. The number of temperatures $P$, as well as $a_0, \ldots, a_P$, is frequently chosen adaptively, as in Sect. 2.2. The tempered likelihood $p(\eta | \theta)^{a_p}$ is estimated by data subsampling as in Sect. 3. The $p$th tempered posterior is

$$
\pi_p(\theta) = \frac{\eta_p(\theta)}{Z_p}, \text{ where } \eta_p(\theta) = p(\eta | \theta)^{a_p} \, p(\theta) \quad \text{and}
$$

$$
Z_p = \int_\Theta p(\eta | \theta)^{a_p} \, p(\theta) \, d\theta.
$$

(3)

SMC starts by sampling a set of $M$ particles from the prior $p(\theta)$ and moves them through the sequence of densities $\pi_p(\theta)$, $p = 1, \ldots, P$, such that, for each $p$, the reweighting, resampling and move steps are performed on the particles. Here, we assume for simplicity that it is possible to sample from the prior; otherwise, one can sample from some initial distribution $\pi_0(\theta)$ whose support covers that of the prior $p(\theta)$. At the final $p = P$, the particles are a (weighted) sample from $\pi(\theta)$. We now discuss this in more detail.

The initial particle cloud and weights $\{\theta^{(0)}_{1:M}, W^{(0)}_{1:M}\}$ are obtained by generating the $\{\theta^{(0)}_{1:M}\}$ from $p(\theta)$, and giving them equal weight, i.e., $W^{(0)}_{1:M} = 1/M$. The weighted particles $\{\theta^{(p-1)}_{1:M}, W^{(p-1)}_{1:M}\}$ at the $(p - 1)$st stage, $p = 1, \ldots, P$, are (weighted) samples from $\pi_{p-1}(\theta)$. At the $p$th stage, the transition from $\pi_{p-1}(\theta)$ to $\pi_p(\theta)$ is obtained by the reweighting step,

$$
W^{(p)}_i = \frac{\eta_p(\theta^{(p-1)}_i)}{\eta_{p-1}(\theta^{(p-1)}_i)} \frac{W^{(p-1)}_i}{\pi_{p-1}(\theta^{(p-1)}_i)} = W^{(p-1)}_i \, p(|\theta^{(p-1)}_i|)^{a_p - a_{p-1}},
$$

and then normalizing $W^{(p)}_i = W^{(p)}_i / \sum_{i=1}^M W^{(p)}_i$. Reweighting usually assigns small weights to particles which are unlikely under the tempered likelihood; this might cause “particle degeneracy”, where all the mass is placed on a small fraction of the particles, causing a small effective sample size (explained in Sect. 2.2). This is resolved by the resampling step, in which the particles $\theta^{(p)}_{1:M}$ are sampled with a probability equal to their normalized weights $W^{(p)}_{1:M}$; the $W^{(p)}_{1:M}$ are then set to $1/M$. We use multinomial resampling for all the experiments and applications in the paper. Although this eliminates particles with small weights, it causes the so-called particle depletion problem because resampling might lead to only a few distinct particles. This is resolved by the move step, in which a $\pi_p$-invariant Markov kernel $K_p$ is applied to move each of the particles $R$ steps. Since a particle after the resampling step at stage $p$ is approximately a sample from $\pi_p(\theta)$ and $K_p$ is $\pi_p$-invariant, no burn-in period is required as in MCMC methods, where often a very large number of burn-in iterations are required. Finally, we note that the algorithm is easy to parallelize with respect to the $M$ particles, because the computations required for each particle are independent of those of the other particles. Thus, provided that $p(\eta | \theta)$ can be computed at each worker without storage issues, it is straightforward to implement the parallelization.

Del Moral et al. (2006) provide consistency results and central limit theorems for estimating (2) based on the SMC output.

### 2.2 Statistical efficiency of SMC

The statistical efficiency of the reweighting part at stage $p$ is measured by the effective sample size (ESS) defined as (Liu 2001)

$$
\text{ESS}_p := \left( \sum_{i=1}^M (W^{(p)}_i)^2 \right)^{-1}.
$$

The $\text{ESS}_p$ varies between 1 and $M$, where a low value of $\text{ESS}_p$ indicates that the weights are concentrated only on a few particles. It is necessary to choose the tempering sequence $\{a_p, p = 1, \ldots, P\}$ carefully because it has a substantial impact on the $\text{ESS}_p$. We follow Del Moral et al. (2012) and choose the tempering sequence adaptively to ensure sufficient particle diversity by selecting the next value of $a_p$ such that $\text{ESS}_p$ stays close to some target value $\text{ESS}_{\text{target}}$; this is done by evaluating the $\text{ESS}_p$ over a grid points $a_{1:S}$, $S$ of potential values of $a_p$ for a given $p$ and select $a_p$ as that value of $a_s, s = 1, \ldots, S$, whose $\text{ESS}_p$ is closest to $\text{ESS}_{\text{target}}$. Throughout our article, $\text{ESS}_{\text{target}} = 0.8M$.

For this adaptive choice of tempering sequence, Beskos et al. (2016) establish consistency results and central limit theorems for estimating (2) based on the SMC output. Other
adaptive methods to choose the tempering sequence such as the approach by Del Moral et al. (2012) may be used instead.

### 2.3 SMC estimation of the marginal likelihood

The marginal likelihood $p(\theta)$ is often used in the Bayesian literature to compare models by their posterior model probabilities (Kass and Raftery 1995). An advantage of SMC is that it automatically produces an estimate of $p(\theta)$.

Using the notation of Sect. 2.1, $Z_p = p(\theta)$, $Z_0 = 1$, and

$$
p(\theta) = \prod_{p=1}^{n} \frac{Z_p}{Z_{p-1}} \text{ with } \frac{Z_p}{Z_{p-1}} = 1 - \left( 1 - \frac{\eta_p(\theta)}{\eta_{p-1}(\theta)} \right) \pi_{p-1}(\theta) d\theta.
$$

Because the particle cloud $\left\{ \theta^{(p-1)}_{1:M}, W^{(p-1)}_{1:M} \right\}$ at the $(p-1)$st stage is an approximate sample from $\pi_{p-1}(\theta)$, it is plausible to estimate the ratios above by

$$
\hat{Z}_p \approx \sum_{i=1}^{M} W^{(p-1)}_i \frac{\eta_p(\theta^{(p-1)}_i)}{\eta_{p-1}(\theta^{(p-1)}_i)},
$$

giving the estimate of the marginal likelihood

$$
\hat{p}(\theta) = \prod_{p=1}^{n} \frac{\hat{Z}_p}{Z_{p-1}}.
$$

### 3 Methodology

#### 3.1 Sequence of target densities

Suppose that $y_k, k = 1, \ldots, n$, are independent given $\theta$ so that the likelihood and log-likelihood can be written as

$$
L(\theta) = \prod_{k=1}^{n} p(y_k | \theta) \quad \text{and} \quad \ell(\theta) = \sum_{k=1}^{n} \ell_k(\theta),
$$

where $\ell_k(\theta) = \log p(y_k | \theta)$. We consider the case where the log-likelihood is computationally very costly, because $n$ is so large that repeatedly computing this sum is impractical, or $n$ is moderately large but each term is expensive to evaluate.

Quiroz et al. (2019) propose to subsample $m$ observations and estimate $L(\theta)$ by

$$
\hat{L}(\theta) = \exp \left( \hat{\ell}_m(\theta) - \frac{1}{2} \hat{\sigma}_m^2(\theta) \right),
$$

where $\hat{\ell}_m(\theta)$ is an unbiased estimator of $\ell(\theta)$ and $\hat{\sigma}_m^2(\theta)$ is an estimate of $\sigma^2(\theta) = \nabla \cdot (\hat{\ell}_m(\theta))$. The motivation for (6) is that $\exp(\hat{\ell}_m(\theta) - \sigma^2(\theta)/2)$ is an unbiased estimator of $L(\theta)$ when $\hat{\ell}_m(\theta)$ is normal (Ceyerly and Dewing 1999).

We note that by the central limit theorem, $\hat{\ell}_m(\theta)$ is likely to be normal for moderate $m$ when $n$ is large even if $m$ is a small fraction of $n$. More generally, (6) is an unbiased estimator for $L_{(m,n)}(\theta) := \mathbb{E}(\hat{L}(\theta))$, which we call the perturbed likelihood. The expectation with respect to the subsampling indices $u$ is discussed below. Quiroz et al. (2019) show that when using the control variate in Sect. 3.2 in the estimator $\hat{\ell}_m(\theta)$, and under some extra plausible assumptions, the fractional error of the perturbed likelihood is

$$
\left| \frac{L_{(m,n)}(\theta) - L(\theta)}{L(\theta)} \right| = O \left( \frac{1}{nm^2} \right).
$$

Our approach is based on extending the target at the $p$th density, i.e., $\pi_p(\theta)$ in (3), to include the set of subsampling indices $u = (u_1, \ldots, u_m)$, where $u \in \mathcal{U} \subset \{1, \ldots, n\}^m$, when sampling data observations with replacement. Let $\hat{L}(\theta)$ be an estimator of the tempered likelihood $L(\theta)^{a_p}$. Similarly to Quiroz et al. (2019), we can unbiasedly estimate $a_p \ell(\theta)$ by $a_p \hat{\ell}(\theta)$; since $\forall (a_p \hat{\ell}(\theta)) = a_p^2 \sigma^2(\theta)$ and motivated by (6), we propose the annealed likelihood estimator

$$
\hat{L}_p(\theta) = \exp \left( a_p \hat{\ell}_m(\theta) - \frac{1}{2} a_p^2 \hat{\sigma}_m^2(\theta) \right).
$$

The extended target at the $p$th density is

$$
\pi_p(\theta, u) \propto \hat{L}_p(\theta) p(\theta) p(u) = \exp \left( a_p \hat{\ell}_m(\theta) - \frac{1}{2} a_p^2 \hat{\sigma}_m^2(\theta) \right) p(\theta) p(u),
$$

where $p(u)$ is the density of $u$ (or, more correctly, a probability mass function since $u$ is discrete). At the final annealing step, (8) becomes $\pi(\theta, u) \propto \hat{L}(\theta) p(\theta) p(u)$, which is the target considered in Quiroz et al. (2019). Quiroz et al. (2019) show that the perturbed marginal density for $\theta$, $\pi_{(m,n)}(\theta) = \int_{\mathcal{U}} \pi_p(\theta, u) du$ converges in total variation to $\pi(\theta)$ at the rate $O \left( \frac{1}{nm^2} \right)$. Hence, our proposed approach is approximate but can be very accurate, while also scaling well with respect to the subsample size. For example, if we take $m = O(\sqrt{n})$, then by Quiroz et al. (2019, Part (ii) of Theorem 1)

$$
\int_{\Theta} \left| \pi_{(m,n)}(\theta) - \pi(\theta) \right| d\theta = O \left( \frac{1}{n^2} \right).
$$

Moreover, suppose that $\varphi(\theta)$ is a scalar function with finite second moment. Then, by Quiroz et al. (2019, Part (ii) of Theorem 1)
Thus, the approximation obtained by our approach converges to the posterior (in total variation norm) at a very fast rate as do the posterior moment estimates. Sections 4 and 5 confirm empirically that we obtain very accurate estimates in most of our applications, even for an $m$ very small relative to $n$.

### 3.2 Efficient estimator of the log-likelihood

Quiroz et al. (2019) estimate $\ell(\theta)$ in (5) by the unbiased difference estimator,

$$\hat{\ell}_m(\theta) = \sum_{k=1}^{n} q_k(\theta) + \frac{n}{m} \sum_{j=1}^{m} \ell_{u,j}(\theta) - q_{u,j}(\theta),$$

where

$$Pr(u_j = k) = \frac{1}{n} \quad \text{for all } k = 1, \ldots, n \quad \text{and } j = 1, \ldots, m,$$

and $q_k(\theta) \approx \ell_k(\theta)$ are control variates. The estimator is based on writing

$$\ell(\theta) = \sum_{k=1}^{n} q_k(\theta) + \sum_{k=1}^{n} d_k(\theta) = q(\theta) + d(\theta),$$

with $d_k(\theta) = \ell_k(\theta) - q_k(\theta)$, $q(\theta) = \sum_k q_k(\theta)$, and $d(\theta) = \sum_k d_k(\theta)$. The last term on the right-hand side of (9) is an unbiased estimator of $d(\theta)$. We discuss below a choice of control variates due to Bardenet et al. (2017), which computes $q(\theta)$ in $O(1)$ time. It follows that the cost of computing the estimator is $O(m)$ and we can take $m = O(\sqrt{n})$ in order to achieve the convergence rates $O(1/n^2)$ for both the perturbed density and its moments as discussed in Sect. 3.1.

Let $\overline{\theta}$ be an estimate of posterior location, for example the posterior mean, obtained from a current particle cloud from $p_{\theta}(\theta, u)$. A second-order Taylor series expansion of the log-density around $\overline{\theta}$ is

$$\ell_k(\theta) = \ell_k(\overline{\theta}) + \nabla_\theta \ell_k(\overline{\theta})^\top (\theta - \overline{\theta}) + \frac{1}{2} (\theta - \overline{\theta})^\top \left( \nabla^2_{\theta\theta} \ell_k(\overline{\theta}) \right) (\theta - \overline{\theta}) + o \left( ||\theta - \overline{\theta}|| \right),$$

where $o(\delta)$ means that $o(\delta)/\delta \to 0$ as $\delta \to 0$. We approximate $\ell_k(\theta)$ by

$$q_k(\theta) = \ell_k(\overline{\theta}) + \nabla_\theta \ell_k(\overline{\theta})^\top (\theta - \overline{\theta}) + \frac{1}{2} (\theta - \overline{\theta})^\top \left( \nabla^2_{\theta\theta} \ell_k(\overline{\theta}) \right) (\theta - \overline{\theta}).$$

Then,

$$q(\theta) = A(\overline{\theta}) + B(\theta - \overline{\theta}) + \frac{1}{2} (\theta - \overline{\theta})^\top C(\overline{\theta}) (\theta - \overline{\theta}),$$

where

$$A(\overline{\theta}) = \sum_k \ell_k(\overline{\theta}), \quad B(\overline{\theta}) = \sum_k \nabla_\theta \ell_k(\overline{\theta})^\top$$

and

$$C(\overline{\theta}) = \sum_k \nabla^2_{\theta\theta} \ell_k(\overline{\theta}).$$

The sums $A(\overline{\theta})$, $B(\overline{\theta})$, and $C(\overline{\theta})$ are computed only once at every stage of the SMC, regardless of the number of particles. Then, for each particle, estimating $d(\theta)$ by $\tilde{d}_m(\theta) = (n/m) \sum_j d_{u,j}(\theta)$ is computed in $O(m)$ time, as does (9), because $q(\theta)$ is $O(1)$. We estimate $\sigma_m^2(\theta) = \nabla (\hat{\ell}_m(\theta))$ by

$$\hat{\sigma}_m^2(\theta) = \frac{n^2}{m^2} \sum_{j=1}^{m} (d_{u,j}(\theta) - \overline{d}_u(\theta))^2,$$

where $\overline{d}_u(\theta)$ denotes the mean of the $d_{u,j}$ for the sample $u = (u_1, \ldots, u_m)$. The estimate $\hat{\sigma}_m^2(\theta)$ is virtually costless since it involves terms that are already computed when obtaining $\overline{d}_m(\theta)$.

### 3.3 The reweighting and resampling steps

The initial particle cloud and weights are now $\left\{ \theta^{(0)}_{1:M}, u^{(0)}_{1:M}, W^{(0)}_{1:M} \right\}$, obtained by generating the $\left\{ \theta^{(0)}_{1:M}, u^{(0)}_{1:M} \right\}$ from $p(\theta)$ and $p(u)$, and assigning equal weights, i.e., $W^{(0)}_{1:M} = 1/M$. The weighted particles $\left\{ \theta^{(p-1)}_{1:M}, u^{(p-1)}_{1:M}, W^{(p-1)}_{1:M} \right\}$ at the $(p - 1)$st stage are a sample from $p_{\theta}(\theta, u)$ and are propagated to $p_{\theta}(\theta, u)$, by updating the weights $W^{(p)}_{1:M} = W^{(p-1)}_{1:M} / \sum_{i=1}^{M} w_{i}^{(p)}$, where

$$w_{i}^{(p)} = W^{(p-1)}_{i} \exp \left( (a_{p} - a_{p-1}) \tilde{e}_{m}(\theta_i^{(p-1)}) - \frac{1}{2} (a_{p}^2 - a_{p-1}^2) \hat{\sigma}_{m}^2(\theta_i^{(p-1)}) \right).$$

The particles $\left\{ \theta^{(p-1)}_{1:M}, u^{(p-1)}_{1:M} \right\}$ are then resampled using the weights $W^{(p)}_{1:M}$ to obtain the equally weighted particles $\left\{ \theta^{(p)}_{1:M}, u^{(p)}_{1:M} \right\}$.
3.4 The Markov move step

The Markov move step uses Hamiltonian dynamics to propose distant particle moves and data subsampling to speed up the computation of the dynamics. Similarly to Sect. 2.1, the Markov move is designed to leave each of the sequence of target densities $π_p(θ, u)$ invariant, for $p = 0, \ldots, P$. Algorithm 1 describes the Markov move step and is divided into two parts to accommodate subsampling. See Dang et al. (2019) for the details.

Algorithm 1 Single Markov move with a kernel invariant for $π_p(θ, u)$ in (8).

For $i = 1, \ldots, M$.

1. Sample $u_i|θ, y$: Propose $u_i^* \sim p(u)$, and set $u_i = u_i^*$, with probability

$$a_u = \min \left( 1, r := \frac{\exp \left( a_p^2 m_i(θ, u_i) \right) - \frac{1}{2} a_p^2 m_i^2(θ, u_i) \right)}{\exp \left( a_p m_i(θ, u) \right) - \frac{1}{2} a_p^2 m_i^2(θ, u) \right) \right)$$

(10)

The proposal $u_1^*$ is independent of the current value of $u_i$, so the difference between the log of the numerator and log of the denominator of the ratio $r$ in (10) can be highly variable. This move might get stuck when the denominator is significantly overestimated. A remedy is to induce a high correlation between the log of the estimated annealed likelihood at the current and proposed draws in (10). This can be achieved either through corrupting the $u$ as in Deligiannidis et al. (2018) (see Quiroz et al. 2019 for discrete $u$) or by block updates of $u$ as in Tran et al. (2017); Quiroz et al. (2018a). We implement the block updates with $G$ blocks, which gives an approximate correlation $1 - \frac{1}{G}$.

2. Sample $θ_i|u_i, y$: Given a subset of data $u_i$, the particle $θ_i$ is moved using a Hamiltonian Monte Carlo (HMC) proposal in a Metropolis–Hastings (MH) algorithm. This becomes a standard HMC move for a given subset $u$.

Note that the above is a Gibbs update of $θ, u|y$. The MH within Gibbs performed in Step 1 is valid (Johnson et al. 2013) and so is the HMC within Gibbs (Neal 2011) in Step 2. Therefore, this kernel has $π_p(θ, u)$ as its invariant distribution. Dang et al. (2019) previously proposed an MCMC version of this algorithm.

Algorithm 2 Subsampling Sequential Monte Carlo

1. Sample the particles $\{θ_i^{(0)}, u_i^{(0)}\}$ from the prior densities $p(θ)$ and $p(u)$ and give all particles equal weights, $W_i = 1/M$, $i = 1, \ldots, M$.

2. While the tempering sequence $α_p \neq 1$ do

(a) Set $p \leftarrow p + 1$

(b) Find $α_p$ adaptively to maintain the ESS around ESS$_{\text{target}}$ (Sect. 2.2).

(c) Reweighting: compute the unnormalized weights

$$W_i^{(p)} = W_i^{(p-1)} \frac{η_p(θ_i^{(p-1)}, u_i^{(p-1)})}{η_{p-1}(θ_i^{(p-1)}, u_i^{(p-1)})}$$

$$= W_i^{(p-1)} \exp \left( \frac{1}{2} \left( a_p^2 - a_{p-1}^2 \right) σ_m^2(θ_i^{(p-1)}) \right)$$

and normalize as $W_i^{(p)} = W_i^{(p)}/\sum_{i=1}^M W_i^{(p)}$, $i = 1, \ldots, M$.

(d) Compute $θ_\text{bar}$ as $θ_\text{bar} = \sum_{i=1}^M W_i^{(p)} θ_i^{(p-1)}$ and then obtain

$$\frac{1}{n} \sum_{k=1}^n θ_k(θ_\text{bar}), \frac{1}{n} \sum_{k=1}^n \nabla_θ θ_k(θ_\text{bar}), \frac{1}{n} \sum_{k=1}^n \nabla_u θ_k(θ_\text{bar})$$

and the mass matrix $H = \Sigma^{-1}$(θ_\text{bar}), where $Σ$ is the sample covariance matrix of current particles.

(e) Resample the particles $\{θ_i^{(p-1)}, u_i^{(p)}\}_{i=1}^M$ using the weights $\{W_i^{(p)}\}_{i=1}^M$ to obtain resampled particles $\{θ_i^{(p)}, u_i^{(p)}\}_{i=1}^M$ and set $W_i^{(p)} = 1/M$.

(f) Apply $R$ Markov moves to each particle $θ_i^{(p)}, u_i^{(p)}$ using Algorithm 1. $R$ is tuned by increasing it until 90% of the product of component-wise autocorrelations of the particles drops below a threshold; see Buchholz et al. (2018) for more details.

3.5 Marginal likelihood estimation

Our approach naturally extends that of Sect. 2.3 by considering the augmented target density $π_p(θ, u)$ in (8). Define

$$γ_p(θ, u) = \frac{η_p(θ, u)}{η_{p-1}(θ, u)}.$$  

Then,

$$\int_Θ \int_Θ \int_Θ γ_p(θ, u)π_{p-1}(θ, u)\text{d}θ\text{d}u = \int_Θ \int_Θ \frac{η_p(θ, u)}{η_{p-1}(θ, u)} \frac{π_{p-1}(θ, u)}{Z_{p-1}} p(u)\text{d}θ\text{d}u$$

$$= \frac{Z_p}{Z_{p-1}}.$$  

 Springer
Thus, if \( \{ \theta_{1:M}^{(p-1)}, u_{1:M}^{(p-1)}, W_{1:M}^{(p-1)} \} \) at the \((p-1)\)st sequence is an approximate sample from \( \pi_{a,p-1}(\theta, u) \), we estimate the ratio \( Z_p / Z_{p-1} \) by

\[
\frac{Z_p}{Z_{p-1}} = \sum_{i=1}^{M} \frac{W_i^{(p-1)}}{\eta_{p-1}} \frac{\eta_p(\theta_i^{(p-1)}, u_i^{(p-1)})}{\eta_{p-1}(\theta_i^{(p-1)}, u_i^{(p-1)})},
\]

and the marginal likelihood estimate is obtained by using this expression in (4).

### 3.6 Efficient memory management by data subsampling

We now explain in detail how data subsampling helps to parallelize the computing in terms of efficient memory utilization. Suppose first that we perform standard SMC (using all the data) and that we parallelize using \( N \) workers, so that each worker deals, on average, with \( M/N \) particles. Then, for each stage \( p \), the computations performed for each particle require repeated likelihood evaluations (using \( n \) data) when applying \( R \) Markov move steps. Hence, each worker needs to have access to the full dataset.

Suppose now that our data subsampling approach is used in the same setting using \( M/N \) particles for each of the \( N \) workers. Then, at the beginning of each stage \( p \) of the algorithm, we still require a full data evaluation for computing \( A(\theta), B(\theta) \) and \( C(\theta) \) in Sect. 3.2. However, at each \( p \), we can now subsample the data according to \( u_i^{(p)} \) for each particle and then perform the \( R \) Markov move steps; these now require repeated evaluations of the estimated annealed likelihood (using \( m \ll n \) observations) and, in addition, \( A(\theta), B(\theta) \) and \( C(\theta) \). Now each worker needs to have access only to the subsampled dataset, as well as \( A(\theta), B(\theta) \) and \( C(\theta) \). However, these are only summaries of the full dataset and are therefore very memory efficient.

We are aware that parallelization of SMC methods is not straightforward to do efficiently when resampling occurs often (Murray et al. 2016; Lee et al. 2010). We note that in all our applications the number of annealing steps is relatively small and therefore resampling does not really affect the efficiency of our algorithm. In applications where resampling occurs more frequently, both SMC methods can benefit from the ideas in Heine et al. (2019) and Guldas et al. (2015). Their methods are variations of augmented resampling and butterfly resampling that aim to boost the efficiency of the resampling steps in parallel and distributed architectures. Moreover, the reweighting and the computationally expensive Markov move steps of our algorithm are easily parallelized for each SMC sample because the computations required for each sample are independent of those of the other samples. Subsampling therefore does not affect the parallelization of the algorithm because only the part of the data specified by the particles \( u_i \) are sent to each worker and the \( u_i \) are independent of each other.

### 4 Methodology evaluation

#### 4.1 Experiments

We now evaluate the methodology through the following experiments.

- **Experiment 1:** Evaluating the usefulness of the Hamiltonian Monte Carlo kernel.
  We show the effectiveness of a HMC kernel for the Markov move step compared to random walk and MALA kernels.

- **Experiment 2:** Evaluating the speed and the accuracy of the estimates of the marginal likelihood and the approximate posterior density when the posterior is unimodal.
  We show that the subsampling approach is accurate by comparing the estimates of the marginal likelihood and posterior densities to those obtained by the full data SMC (representing the gold standard).

- **Experiment 3:** Evaluating the speed and the accuracy of the estimates of the marginal likelihood and the approximate posterior density when the posterior is non-Gaussian.
  We use the subsampling approach when the posterior is bimodal or skewed and show that the method still performs effectively.

- **Experiment 4:** Evaluating the effect of the accuracy of the control variate.
  We show that the subsampling approach can be made faster by using a first-order control variate instead of the second-order one. This experiment also shows the effect of poor control variates on the performance of our method.

All the SMC algorithms are tuned as in Buchholz et al. (2018) using 280 particles, a choice motivated by our cluster with 28 cores with each core dealing (on average) with 10 particles. The only exception is the first scenario in Experiment 3 where we use 420 particles for both algorithms to capture the multimodal posterior more accurately. Each experiment was repeated 10 times to compute the standard error of the log marginal likelihood estimator. Experiments 1, 2 and the bankruptcy application in Sect. 5 were done using the Australia NCI High Performance Computing System Raijin\(^1\).

---

\(^1\) [https://nci.org.au/our-systems/hpc-systems](https://nci.org.au/our-systems/hpc-systems).
Experiments 3 and 4 were done using the University of New South Wales computational cluster Katana\textsuperscript{2}.

We remark that the choice of priors can affect the computational efficiency of SMC methods. In general, a prior that resembles the likelihood requires less tempering steps. However, this is unlikely to influence the comparison between Subsampling SMC and SMC, which is our primary goal.

4.2 Experiment 1: Evaluating the Markov move kernel

We first consider a logistic regression to evaluate the effectiveness of the Hamiltonian Monte Carlo Markov move step compared to the random walk and MALA kernels. The model for the response $y_i \in \{0, 1\}$ given a $d \times 1$ set of covariates and parameters is

$$p(y_i | x_i, \theta) = \frac{\exp(y_i x_i^\top \theta)}{1 + \exp(x_i^\top \theta)}.$$  

We fit this model to the HIGGS dataset (Baldi et al. 2014), having $n = 11,000,000$ observations and 28 covariates. The response is “detected particle” and 21 of the covariates are kinematic properties measured by particle detectors, while 7 are high-level features included to capture nonlinearities. This means that $d = 29$, including the intercept. We take the prior $\theta \sim \mathcal{N}(0, I_d)$, where $I_d$ is the $d \times d$ identity matrix and follow Buchholz et al. (2018) to set the tuning parameters, including the number of Markov moves $R$. The mass matrix in both HMC and MALA is $\Sigma^{-1}$, which is the estimated inverse covariance matrix of the tempered posterior. We note that each step in the sequence has a corresponding estimate of this inverse covariance matrix, obtained using the corresponding particles from that step. For the random walk, the optimal scaling ($2.38^2/d)\Sigma$ (Roberts et al. 1997) resulted in numerical errors, so that we decreased it by a factor of 10.

Table 1 summarizes the results obtained using the second-order control variate in Sect. 3.2. The log-likelihood estimator has $\text{ESS} = 8,000$ particles. All methods use the second-order control variate in Sect. 3.2. The results are averaged over 10 runs, which are used to compute the standard error of the estimator available at each worker (we use 28) as explained in Sect. 3.6, in order to compute the likelihood together with its gradient and Hessian, which would quickly consume the RAM of the computer. Instead, we consider the following two models.

Student-t regression We consider a univariate Student-t regression

$$y_i = x_i^\top \theta + \epsilon_i, \quad \epsilon_i \sim t_5,$$

where $t_5$ is the Student-t distribution with 5 degrees of freedom. We generated a simulated dataset with $n = 500,000$ observations and $d = 50$ covariates. The covariates were generated so that their marginal variances are 1 and their pairwise correlations are 0.9. The parameters $\theta$ were simulated independently from a Uniform($-5, 5$) distribution; the prior for $\theta$ is $\mathcal{N}(0, 10I_d)$.

Poisson regression We also considered a Poisson regression, where the univariate $y$ follows a Poisson distribution with an expectation that is log-linear, i.e.,

$$y_i | x_i \sim \text{Poisson}(\exp(x_i^\top \theta)).$$

We generated $n = 200,000$ observations with $d = 30$ covariates, 29 of them simulated from $x_i \sim \mathcal{N}(0, I_{29})$ and the last one is 1. The parameters are simulated independently from Uniform($-0.2, 0.2$) and are assigned the prior $\theta \sim \mathcal{N}(0, 0.1I_d)$. We found that both Subsampling SMC and SMC were particularly sensitive to the prior choice for the Poisson regression, resulting in numerical overflow for priors that were too noninformative.

Table 1 Comparing the performances of three kernels for the Markov move: Hamiltonian Monte Carlo (HMC), Metropolis adjusted Langevin algorithm (MALA) and random walk (RW)

| Method | log marginal likelihood | CPU time (h) | $P$ | $R$ |
|--------|------------------------|-------------|-----|-----|
| HMC    | -7,013,460.90          | 2.31        | 106 | 5   |
| MALA   | -7,013,462.49          | 4.77        | 106 | 20  |
| RW     | -7,013,461.43          | 33.43       | 106 | 200 |

2 https://research.unsw.edu.au/katana.
correspond to a sample fraction of about 0.0025. Table 2 summarizes the results and shows that the subsampling approach is about 6.5 to 10.5 times faster and, moreover, confirms the accuracy of the marginal likelihood estimate of our method.

Finally, Figs. 1 and 2 show that the marginal posterior densities are very well approximated for both the Student-t regression and the Poisson regression; the same accuracy was obtained for all parameters (not shown).

### 4.4 Experiment 3: Evaluating speed and accuracy of Subsampling SMC on non-Gaussian targets

To evaluate the performance of Subsampling SMC when the posterior is non-Gaussian, we consider the fixed effects model

$$y_{ij} = \alpha_i + x_{ij}^T \beta + e_{ij}, \quad i = 1, \ldots, n, \quad j = 1, \ldots, n_i, \quad e_{ij} \sim \mathcal{N}(0, \sigma_i^2).$$

For simplicity, we set $\sigma_i^2 = 1$ for all individuals; two different scenarios are used for the individual fixed effects $\alpha_i$: a) a mixture of normals prior for the $\alpha_i$, b) a truncated normal prior for the $\alpha_i$. For each scenario, we generated a dataset of $n = 10$ individuals, with $n_1 = \cdots = n_5 = 20$ observations and $n_6 = \cdots = n_{10} = 50,000$ observations. The covariates were generated independently from $\mathcal{N}(0, 1)$; the $\beta$ parameters were generated from $\mathcal{N}(0, 2^2)$. The prior for $\beta$ in both scenarios is $\mathcal{N}(0, 3^2)$.

#### Mixture of normals prior

The first prior is motivated by a scenario where we believe some coefficients may be 0 or very close to 0 and would like the posterior to take that into account. For each of the fixed effects $\alpha_i$, we used a mixture of normals prior

$$p(\alpha_i | w, \sigma_1, \sigma_2) = w\phi(\alpha_i | \sigma_1^2) + (1 - w)\phi(\alpha_i | \sigma_2^2),$$

where $i = 1, \ldots, n$;

$$\phi(\cdot | \sigma^2)$$ is the density of the normal distribution with mean 0 and variance $\sigma^2$, and we set $w = 0.8$, $\sigma_1 = 0.1$ and $\sigma_2 = 3.5$. In this experiment, the first 5 individual fixed effects $\alpha_i$ were generated from $\mathcal{N}(0.5, 0.05^2)$ and the rest from $\mathcal{N}(0.5, 0.2^2)$.

Even though the likelihood for each individual is likely to be unimodal, the prior leads to more complicated posteriors for those individual effects that correspond to subjects with a small number of observations. The likelihood is

$$L(\alpha, \beta) = \prod_{i=1}^n p(y_i | \alpha_i, \beta), \quad \text{where}$$

$$p(y_i | \alpha_i, \beta) = \prod_{j=1}^{n_i} p(y_{ij} | \alpha_i, \beta)$$

is the contribution to the likelihood from subject $i$. The likelihood $L(\alpha, \beta)$ and the annealed likelihoods are estimated by estimating the individual densities with subsampling. The subsample size is $m = 5$, with no blocking for the first 5 individuals and $m = 100$ with $G = 100$ blocks for the remaining 5 individuals. In practice, it is unnecessary to estimate the likelihood contribution for the individuals with few observations since it is relatively cheap computationally to evaluate their full likelihood contribution; however, we do so.
in our experiment to gain a better understanding about the benefits and limitations of our subsampling approach.

We ran subsampling SMC with second-order Taylor series expansions in both scenarios. Table 3 summarizes the results and shows that Subsampling SMC produces similar results to full data SMC but is about 9 times faster. All SMC methods used here require the maximum number of Markov moves at most temperatures, indicating that the posterior is challenging to explore. The problem is due to the posterior being multimodal; however, our results show that the HMC proposal works adequately in this example and Subsampling SMC is significantly faster than full data SMC. Figure 3 shows that even when some of the marginal posteriors ($\alpha_1$, $\alpha_3$ and $\alpha_5$) are bimodal, Subsampling SMC is also able to capture the bimodality and gives the same approximation as full data SMC. In comparison, we also include in the figure the result from running 10,000 post-burn-in iterations of Subsampling MCMC. It is well known that conventional MCMC methods may not be able to sample efficiently from multimodal targets, and in this experiment Subsampling MCMC can detect the posterior modes, but there is still some visible discrepancy between its result and that of full data SMC. We do not show the marginal posterior densities of $\beta$ which appear to be Gaussian, but confirm that both methods give similar results.

Our method works in this example because the bimodality is caused by the prior and not the likelihood; if the bimodality was caused by the likelihood, different control variates would be necessary since our control variates assume the log-density is quadratic $\theta$. We leave the development of more flexible control variates for Subsampling SMC for future research.

**Truncated normal prior** The second scenario is motivated by situations in which there is strong prior knowledge that the coefficients are positive. To create such a situation, the fixed effects $\alpha_i$ were generated from a truncated normal distribution $T\mathcal{N}(0, 0.12)$. We assigned the prior $\alpha_i \sim T\mathcal{N}(0, 3^2)$, $i = 1, \ldots, 10$, to the individual fixed effects to reflect this prior knowledge. The subsample size is $m = 20$ (all observations) with no blocking for the first 5 individuals and $m = 200$ with $G = 100$ blocks for the $6^{th}$ individual. The remaining 4 individuals have $m = 100$ with $G = 100$. Note that the subsample size affects the variance of the log-likelihood estimator and hence the accuracy of our method; see Quiroz et al. (2019) and Dang et al. (2019) for further
Fig. 2 Kernel density estimates of a subset of the marginal posterior densities of \( \theta \) for the Poisson regression model with simulated data. The density estimates are obtained by full data SMC and Subsampling SMC.

4.5 Experiment 4: Evaluating the effect of the control variate

The results above show that the subsampling approach accurately estimates the marginal likelihood and marginal posterior densities using a second-order Taylor series expansion. We now study robustness of the results to the quality of the control variates. The first study uses first-order Taylor expansions for the control variates for subsampling applied to the logistic regression for the HIGGS dataset in Sect. 4.2. Table 4 summarizes the results, and confirms that the marginal likelihood estimates using first-order control variates are as accurate as using second-order control variates, but five times faster. Figure 5 shows that the marginal posterior densities remain accurate; we have confirmed similar accuracy for all the parameters.

We now present an example where inaccurate likelihood estimates lead to a biased result. We consider again the fixed effects model described in Sect. 4.4 with the individual effects \( \alpha_i \) having a truncated normal prior, \( p(\alpha_i) \sim TN\left(0, 3^2\right); m = 5 \) is used for the first 5 individuals and \( m = 100 \) with \( G = 100 \) for the remaining 5 individuals. We deliberately decrease the subsample size for the 6\(^{th} \) individual to 100 to increase the variance of the likelihood estimate, in order to test the robustness of our method.

Table 4 and Fig. 6 summarize the results; they show that Subsampling SMC has difficulties exploring the skewed posteriors and gives inaccurate results when \( m \) is too small. Our approach works poorly here because the posteriors for the first 5 individual effects are highly skewed, and their skewness is caused by the truncated prior and the small number of observations. This causes the posterior to be concentrated at
### Table 3
Comparing the performances of Subsampling SMC and full data SMC

|                                | log marginal likelihood | CPU time (h) | $P$ | $R$ |
|--------------------------------|-------------------------|-------------|-----|-----|
| **Mixture of normals priors**  |                         |             |     |     |
| $(M = 420)$                    |                         |             |     |     |
| Full data SMC                 | $-354,914.89$           | 14.36       | 81  | 99  |
|                               | (0.78)                  |             |     |     |
| Subsampling SMC               | $-354,915.22$           | 1.60        | 81  | 100 |
|                               | (1.18)                  |             |     |     |
| **Truncated normal priors**   |                         |             |     |     |
| $(M = 280)$                    |                         |             |     |     |
| Full data SMC                 | $-354,445.12$           | 0.74        | 79  | 5   |
|                               | (0.26)                  |             |     |     |
| Subsampling SMC               | $-354,444.04$           | 0.68        | 88  | 20  |
|                               | (2.2)                   |             |     |     |

The table shows the log of the estimate of the marginal likelihood (with standard errors in brackets), the CPU time, the number of annealing steps $P$ (tuned to maintain $ESS \approx 0.8M$) and the average number of Markov moves $R$ (tuned as in Buchholz et al. 2018, and the maximum number of Markov moves at each temperature is set to be 100). The results are for the fixed effects model estimated using the simulated datasets. All the methods use the second-order control variate in Sect. 3.2. The results are averaged over 10 runs, which are used to compute the standard error of the estimator.

**Fig. 3** Kernel density estimates of a subset of the marginal posterior densities of $\alpha$ for the fixed effects model with mixture of normals priors, using simulated data. The density estimates are obtained by full data SMC, Subsampling SMC and Subsampling MCMC.
the tail of the log-density, where the control variates using a quadratic approximation are inaccurate. Therefore, updating $\theta$ by the posterior mean as specified in Algorithm 2 does not produce good control variates, even though the log-density is well behaved. Our likelihood estimate is inaccurate with high variance even when we use a slightly smaller subsample size compared to the previous section for estimating these skewed posteriors. Dang et al. (2019) discuss a similar issue with inaccurate control variates for Subsampling MCMC. We leave the development of more flexible control variates and the guidelines to choose an optimal subsample size, especially for complex posteriors, for future research. Finally, Subsampling SMC is not faster than full data SMC here because of the much larger $P$ and $R$ chosen by using the adaptive tuning method by Buchholz et al. (2018).

5 Application: Modeling firm bankruptcy

The application of our model selection method is now illustrated using a Swedish firm bankruptcy dataset containing $n = 4,748,089$ observations; the response variable is firm default and there are eight firm-specific and macroeconomic covariates, giving 9 covariates, including an intercept. The data is treated as cross-sectional and the bank status is modeled by the logistic regression discussed in Sect. 4.2. A generalized additive model is also fitted to the data and is compared to a linear model; both models use a similar prior $\theta \sim \mathcal{N}(0, 10^2 I_d)$. We compare the marginal posterior density estimates of Subsampling SMC against those of Subsampling MCMC as implemented by Dang et al. (2019) and find them nearly indistinguishable. We also compare both methods to the full data MCMC as in Dang et al. (2019). However, it is unclear how to use Subsampling MCMC for model selection. Frequently used methods such as Chib and Jeliazkov (2001) are not useful for Subsampling MCMC since the (perturbed) likelihood cannot be evaluated; this is a major advantage of Subsampling SMC compared to Subsampling MCMC.

We select between model $M_1$ which is linear in the data on the logit scale and has 9 coefficients, and model $M_2$ which is a semi-parametric additive model on the logit scale using B-splines as in Dang et al. (2019); model $M_2$ is nonlinear in the data (on the logit scale) and has 81 coefficients. Nonlinear bankruptcy models for this dataset were previously analyzed in Quiroz and Villani (2013) and Giordani et al. (2014). Given the marginal likelihood estimates, the estimated Bayes factor (BF) for the nonlinear model $M_2$ vs the linear model $M_1$ is...
Table 4 Comparing the performance of the less accurate control variate (first order) to the more accurate control variate (second order)

|                        | log marginal likelihood | CPU time (h) | $P$  | $R$ |
|------------------------|-------------------------|-------------|------|-----|
| **Logistic regression**|                         |             |      |     |
| 1st order              | -7,013,461.07           | 0.47        | 106  | 5   |
| (0.46)                 |                         |             |      |     |
| 2nd order              | -7,013,460.90           | 2.31        | 106  | 5   |
| (0.32)                 |                         |             |      |     |
| **Truncated normal priors** |                   |             |      |     |
| Full data SMC          | -354,445.12             | 0.74        | 79   | 5   |
| (0.26)                 |                         |             |      |     |
| Subsampling SMC        | -354,437.34             | 1.13        | 141  | 36  |
| (4.64)                 |                         |             |      |     |

The table shows the log of the estimate of the marginal likelihood (with standard errors in brackets), the CPU time, the number of annealing steps $P$ (tuned to maintain $ESS \approx 0.8M$) and the average number of Markov moves $R$ (tuned as in Buchholz et al. (2018)). The results are for the logistic regression model, estimated with the HIGGS dataset, using $M = 280$ particles. The results are averaged over 10 runs, which are used to compute the standard error of the estimator.

\[ \hat{BF}_{21} = \frac{\hat{Pr}(y|\mathcal{M}_2)}{\hat{Pr}(y|\mathcal{M}_1)}; \quad (12) \]

this is also the estimated ratio of posterior model probabilities when the prior model probabilities are equal. We use the strength of evidence guidelines in Jeffreys (1961, p. 438) to choose between the models; Jeffreys considers $10^{3/2} < BF_{21} < 10^2$ as very strong evidence for model $\mathcal{M}_2$ and $BF_{21} > 10^2$ as decisive evidence.

The number of blocks was set to $G = 100$ with the subsample size set to $m = 3,000$; for Subsampling MCMC these tuning parameters were set as in Dang et al. (2019). The estimates from the full data MCMC are considered as the “gold standard” when assessing the accuracy of the algorithms.

Fig. 5 Kernel density estimates of a subset of the marginal posterior densities of $\theta$ for the logistic model with the HIGGS data. The density estimates are both obtained by Subsampling SMC, using different control variates based on a first- and second-order Taylor series expansion as explained in Sect. 3.2.
This was achieved through an MCMC chain of 2,000 post-burn-in MCMC samples, with the burn-in = 1,000 iterations. The MCMC mixed well and we believe that the iterates represent the posterior adequately.

Table 5 reports the log of the estimated marginal likelihood for both models and the corresponding Bayes factors obtained by Subsampling SMC. The table shows decisively that the nonlinear model is superior. We again stress that producing marginal likelihood estimates is straightforward by SMC, whereas it is currently infeasible with Subsampling MCMC.

Figure 7 shows the kernel density estimates of the marginal posterior of selected parameters of the nonlinear model for the bankruptcy dataset. It is evident that both Subsampling SMC and Subsampling MCMC are very accurate and we have confirmed the accuracy of the kernel density estimates for all the parameters, which we do not show to save space. Instead, Fig. 8 shows the estimated marginal posterior expectations and posterior variances by the two algorithms for all the parameters in the nonlinear model; this confirms the accuracy of the estimates of each parameter. We also confirmed that the kernel density estimates and the estimated marginal posterior expectations and posterior variances are accurate for the linear model (not shown).

Figure 9 shows that the nonlinear model \( M_2 \) fits the data better than the linear model \( M_1 \), and is consistent with the estimated Bayes Factor.
6 Conclusions

A simple and effective approach is proposed to speed up sequential Monte Carlo for static Bayesian models using data subsampling. Its key ingredients are an efficient annealed likelihood estimator and an effective Markov kernel move step based on Hamiltonian Monte Carlo to boost particle diversity. This kernel is computationally expensive for large datasets and data subsampling is crucial to obtain a feasible approach. We argue that the subsampling approach is also very convenient for managing computer memory when implementing SMC using parallel computing, because it avoids the need for each worker to store the full dataset. We demonstrate that the method performs efficiently and accu-
Fig. 9 Realized and estimated bankruptcy probabilities. The figure shows the results with respect to the size variable (logarithm of deflated sales) for $M_1$ (left panel) and $M_2$ (right panel). The data are divided into 100 equally sized groups based on the size variable. For each group, the empirical estimate of the bankruptcy probability is the fraction of bankrupt firms. These empirical estimates are represented as dots, where the corresponding $x$-value (size) has been set to the mean within the group. The model estimates for each of the 100 groups are obtained by, for each posterior sample $\theta$, averaging the posterior predictive $\Pr(\tilde{y}_k = 1 | y, x_k)$ for all observations $k$ in a group, and subsequently computing the posterior predictive mean $\mathbb{E}(\tilde{y}_k = 1 | y, x_k)$ (solid line) and 90% prediction interval (quantiles 5–95, shaded region).

rately for four generalized linear models and a generalized additive model. Moreover, it allows Bayesian model selection through accurate estimates of the marginal likelihood, which is a major advantage compared to Subsampling MCMC. We also illustrate that the limitation of our method is that its performance depends on good control variates, which can be challenging to construct in certain models. An anonymous reviewer suggested we may use the SMC particles to construct a surrogate function to use as control variate in more complex models. Another anonymous reviewer specifically suggested that, at each temperature, we divide the particles into $K$ clusters and then develop control variates for each cluster. How to do this in a computationally efficient way is an open question, which we leave for future research.

Acknowledgements We thank the Associate Editor and two reviewers for helping to improve both the content and the presentation of the article. Khue-Dung Dang, David Gunawan, Matias Quiroz and Robert Kohn were partially supported by Australian Research Council Center of Excellence grant CE140100049.

References

Baldi, P., Sadowski, P., Whiteson, D.: Searching for exotic particle in high energy physics with deep learning. Nat. Commun. 5, 1–9 (2014)

Barndenet, R., Doucet, A., Holmes, C.: On Markov chain Monte Carlo methods for tall data. J. Mach. Learn. Res. 18(1), 1515–1557 (2017)

Beskos, A., Jasra, A., Kantas, N., Thiery, A.: On the convergence of adaptive sequential Monte Carlo methods. Ann. Appl. Probab. 26(2), 1111–1146 (2016)

Betancourt, M.: A conceptual introduction to Hamiltonian Monte Carlo. ArXiv preprint arXiv:1701.02434 (2017)

Brooks, S., Gelman, A., Jones, G., Meng, X.-L.: Handbook of Markov chain Monte Carlo. CRC Press, Boca Raton (2011)

Buchholz, A., Chopin, N., Jacob, P.E.: Adaptive tuning of Hamiltonian Monte Carlo within sequential Monte Carlo. ArXiv preprint arXiv:1808.07730 (2018)

Cepferley, D., Dewing, M.: The penalty method for random walks with uncertain energies. J. Chem. Phys. 110(20), 9812–9820 (1999)

Chib, S., Jeliazkov, I.: Marginal likelihood from the Metropolis–Hastings output. J. Am. Stat. Assoc. 96(453), 270–281 (2001)

Chopin, N.: A sequential particle filter method for static models. Biometrika 89(3), 539–552 (2002)

Dang, K.-D., Quiroz, M., Kohn, R., Tran, M.-N., Villani, M.: Hamiltonian Monte Carlo with energy conserving subsampling. J. Mach. Learn. Res. 20(100), 1–31 (2019)

Daviet, R.: Inference with Hamiltonian sequential Monte Carlo simulators. http://www.remiдавиет.com/files/HSMC-paper.pdf (2016)

Del Moral, P., Doucet, A., Jasra, A.: Sequential Monte Carlo samplers. J. Roy. Stat. Soc. B 68(3), 411–436 (2006)

Del Moral, P., Doucet, A., Jasra, A.: An adaptive Sequential Monte Carlo for approximate Bayesian computation. Stat. Comput. 22(5), 1009–1020 (2012)

Deligianni, G., Doucet, A., Pitt, M.K.: The correlated pseudomarginal method. J. R. Stat. Soc. Ser. B Stat. Methodol. 80(5), 839–870 (2018)

Doucet, A., De Freitas, N., Gordon, N.: An introduction to sequential Monte Carlo methods. In: Sequential Monte Carlo Methods in Practice, pp. 3–14. Springer (2001)

Duan, J.C., Fulop, A.: Density-tempered marginalised sequential Monte Carlo samplers. J. Bus. Econ. Stat. 33(2), 192–202 (2015)

Duane, S., Kennedy, A.D., Pendleton, B.J., Roweth, D.: Hybrid Monte Carlo. Phys. Lett. B 195(2), 216–222 (1987)

Fearnhead, P., Taylor, B.M.: An adaptive sequential Monte Carlo sampler. Bayesian Anal. 8(2), 411–438 (2013)

Giordani, P., Jacobson, T., Von Schedvin, E., Villani, M.: Taking the twists into account: predicting firm bankruptcy risk with splines of financial ratios. J. Financ. Quant. Anal. 49(4), 1071–1099 (2014)
Guldas, H., Cemgil, A.T., Whiteley, N., Heine, K.: A practical introduction to butterfly and adaptive resampling in sequential Monte Carlo. IFAC-PapersOnLine 48(28), 787–792 (2015)
Heine, K., Whiteley, N., Cemgil, A.T.: Parallelizing particle filters with butterfly interactions. Scand. J. Stat. 47, 361–396 (2019)
Jasra, A., Stephens, D.A., Doucet, A., Tsagaris, T.: Inference for Lévy-driven stochastic volatility models via adaptive Sequential Monte Carlo. Scand. J. Stat. 38(1), 1–22 (2011)
Jeffreys, H.: The Theory of Probability. OUP, Oxford (1961)
Johnson, A.A., Jones, G.L., Neath, R.C.: Component-wise Markov chain Monte Carlo: uniform and geometric ergodicity under mixing and composition. Stat. Sci. 28(3), 360–375 (2013)
Kass, R.E., Raftery, A.E.: Bayes factors. J. Am. Stat. Assoc. 90(430), 773–795 (1995)
Lee, A., Yau, C., Giles, M.B., Doucet, A., Holmes, C.C.: On the utility of graphics cards to perform massively parallel simulation of advanced Monte Carlo methods. J. Comput. Graph. Stat. 19(4), 769–789 (2010)
Liu, J.S.: Monte Carlo Strategies in Scientific Computing. Springer, New York (2001)
Murray, L.M., Lee, A., Jacob, P.E.: Parallel resampling in the particle filter. J. Comput. Graph. Stat. 25(3), 789–805 (2016)
Neal, R.: Annealed importance sampling. Stat. Comput. 11, 125–139 (2001)
Neal, R.M.: MCMC using Hamiltonian dynamics. Handbook of Markov chain Monte Carlo (2011)
Quiroz, M., Villani, M.: Dynamic mixture-of-experts models for longitudinal and discrete-time survival data. https://github.com/mattiasvillani/Papers/raw/master/DynamicMixture.pdf (2013)
Quiroz, M., Tran, M.-N., Villani, M., Kohn, R., Dang, K.-D.: The block-Poisson estimator for optimally tuned exact subsampling MCMC. ArXiv preprint arXiv:1603.08232v5 (2018a)
Quiroz, M., Villani, M., Kohn, R., Tran, M.-N., Dang, K.-D.: Subsampling MCMC: an introduction for the survey statistician. Sankhya A 80, 33–69 (2018b)
Quiroz, M., Kohn, R., Villani, M., Tran, M.N.: Speeding up MCMC by efficient data subsampling. J. Am. Stat. Assoc. 114, 831–843 (2019)
Roberts, G.O., Stramer, O.: Langevin diffusions and Metropolis-Hastings algorithms. Methodol. Comput. Appl. Probab. 4(4), 337–357 (2002)
Roberts, G.O., Gelman, A., Gilks, W.R.: Weak convergence and optimal scaling of random walk Metropolis-Hastings. Ann. Appl. Probab. 7(1), 110–120 (1997)
Sim, A., Filippi, S., Stumpf, M.P.: Information geometry and sequential Monte Carlo. ArXiv preprint arXiv:1212.0764 (2012)
South, L.F., Pettitt, A.N., Friel, N., Drovandi, C.C.: Efficient use of derivative information within SMC methods for static Bayesian models. https://eprints.qut.edu.au/108150/ (2017)
South, L.F., Pettitt, A.N., Drovandi, C.C., et al.: Sequential Monte Carlo samplers with independent Markov chain Monte Carlo proposals. Bayesian Anal. 14(3), 753–776 (2019)
Tran, M.N., Kohn, R., Quiroz, M., Villani, M.: The block-pseudo marginal sampler. ArXiv preprint arXiv:1603.02485v5 (2017)
Wang, L., Wang, S., Bouchard-Côté, A.: An annealed sequential Monte Carlo method for Bayesian phylogenetics. Syst. Biol. 69(1), 155–183 (2020)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.