SIAM/ASA J. Uncertainty Quantification
Vol. 7, No. 2, pp. 444–471

© 2019 SIAM and ASA. Published by SIAM and ASA under the terms of the Creative Commons 4.0 license

Ensemble Transport Adaptive Importance Sampling*

Colin Cotter†, Simon Cotter‡, and Paul Russell‡

Abstract. Markov chain Monte Carlo methods are a powerful and commonly used family of numerical methods for sampling from complex probability distributions. As applications of these methods increase in size and complexity, the need for efficient methods increases. In this paper, we present a particle ensemble algorithm. At each iteration, an importance sampling proposal distribution is formed using an ensemble of particles. A stratified sample is taken from this distribution and weighted under the posterior, and then a state-of-the-art ensemble transport resampling method is used to create an evenly weighted sample ready for the next iteration. We demonstrate that this method, called ensemble transport adaptive importance sampling (ETAIS), outperforms Markov chain Monte Carlo (MCMC) methods with equivalent proposal distributions for low-dimensional problems, and in fact it shows better-than-linear improvements in convergence rates with respect to the number of ensemble members. We also introduce a new resampling strategy, multinomial transformation (MT), which, while not as accurate as the ensemble transport resampler, is substantially less costly for large ensemble sizes, and it can then be used in conjunction with ETAIS for complex problems. We also focus on how algorithmic parameters regarding the mixture proposal can be quickly tuned to optimize performance. In particular, we demonstrate this methodology's superior sampling for multimodal problems, such as those arising from inference for mixture models, and for problems with expensive likelihoods requiring the solution of a differential equation, for which speed-ups of orders of magnitude are demonstrated. Likelihood evaluations of the ensemble could be computed in a distributed manner, suggesting that this methodology is a good candidate for parallel Bayesian computations.

Key words. MCMC, importance sampling, Bayesian, inverse problems, ensemble, resampling

AMS subject classifications. 62F15, 65C05, 65C40

DOI. 10.1137/17M1114867

Downloaded 11/28/19 to 90.194.4.112. Redistribution subject to CCBY license

1. Introduction.

Markov chain Monte Carlo (MCMC) methods, first developed in the early 1970s [22], have gained increasing importance and interest in the last 20 years. They allow us to sample from complex probability distributions, which we would otherwise not be able to sample from directly. In particular, these methods have revolutionized the way inverse problems can be tackled, allowing full posterior sampling when using a Bayesian framework.

However, this sampling often comes at a very high cost, with a very large number of iterations required in order for the empirical approximation of the posterior to be considered good enough. As the cost of computing likelihoods can be extremely large, this means that many problems of interest are simply computationally intractable.

*Received by the editors February 6, 2017; accepted for publication (in revised form) January 30, 2019; published electronically April 25, 2019.
http://www.siam.org/journals/juq/7-2/M111486.html
Funding: The second author was supported by EPSRC First grant award EP/L023393/1.
†Department of Mathematics, Imperial College London, London SW7 2AZ, UK (colin.cotter@imperial.ac.uk).
‡School of Mathematics, University of Manchester, Manchester M13 9PL, UK (simon.cotter@manchester.ac.uk, paul.russell-2@manchester.ac.uk).

© 2019 SIAM and ASA. Published by SIAM and ASA under the terms of the Creative Commons 4.0 license

This problem has been tackled in a variety of ways. One approach constructs increasingly complex MCMC methods able to use the structure of the posterior to make more intelligent proposals, leading to more thorough exploration of the posterior with fewer iterations. For example, the Hamiltonian, or hybrid, Monte Carlo (HMC) algorithm uses gradient information and symplectic integrators to make very large moves in state with relatively high acceptance probabilities [43]. Nonreversible methods are also becoming quite popular because they can improve mixing [3]. Riemann manifold Monte Carlo methods exploit the Riemann geometry of the parameter space and are able to take advantage of the local structure of the target density to produce more efficient MCMC proposals [20]. This methodology has been successfully applied to Metropolis-adjusted Langevin algorithm (MALA)-type proposals and methods which exploit even higher order gradient information [6].

Since the clock speed of an individual processor is no longer following Moore's law [32], improvements in computational power are largely a result of the parallelization of multiple cores. As such, the area of parallel MCMC methods is increasingly gaining interest. One class of parallel MCMC method uses multiple proposals, with only one of these proposals being accepted. Examples of this approach include multiple try MCMC [27] and ensemble MCMC [33]. In [7], a general construction for the parallelization of MCMC methods was presented, which demonstrated speed-ups of up to two orders of magnitude when compared with serial methods on a single core. Another approach involves prefetching, where the possible future acceptances/rejections are calculated in advance [2].

A variety of other methods have been designed with particular scenarios in mind. For instance, sampling from high- and infinite-dimensional posterior distributions is of interest in many applications. The majority of Metropolis–Hastings algorithms suffer from the curse of dimensionality, requiring more samples for a given degree of accuracy because the parameter dimension is increased. However, some dimension-independent methods have been developed that are based on Crank–Nicolson discretizations of certain stochastic differential equations (SDEs) [14]. Other methods, such as chain adaptation [21] and early rejection of samples, can also help to reduce the computational workload [45].

However, high dimensionality is not the only challenge that we may face. Complex structure in low dimensions can cause significant issues, which may arise due to large correlations between parameters in the posterior, leading to long, thin curved structures that many standard methods struggle with. These features are common, for example, in inverse problems related to epidemiology and other biological applications [23].

Multimodality of the posterior can also lead to incredibly slow convergence. Many methods allow for good exploration of the current mode, but the waiting time until the chain switches to another mode may be long. Since many switches are required in order for the correct weighting to be given to each mode and for all of the modes to be explored fully, this presents a significant challenge. This is an ever-present problem in the application of mixture models. Given a dataset, where we know that the data is from two or more different distributions, we wish to identify the parameters, e.g., the mean and variance and relative weight, of each part of the mixture [29]. The resulting posterior distribution is invariably a multimodal distribution, since the likelihood is invariant to permutations. Metropolis–Hastings algorithms, for example, will often fail to converge in a reasonable time frame for problems such as this. Since the posterior may be multimodal, independent of this label switching, it is important to be able to efficiently sample from the whole posterior.

Downloaded 11/28/19 to 90.194.4.112. Redistribution subject to CCBY license

© 2019 SIAM and ASA. Published by SIAM and ASA under the terms of the Creative Commons 4.0 license

Importance samplers are another class of methods which allow us to sample from complex probability distributions. A related class of algorithms, adaptive importance sampling (AIS) [26], received less attention until its practical applicability was demonstrated in the mid-2000s [4, 9, 11, 24]; see also the review in [5]. AIS methods produce a sequence of approximating distributions, constructed from mixtures of standard distributions, from which samples can be easily drawn. At each iteration the samples are weighted, often using standard importance sampling methods. The weighted samples are used to train the adapting sequence of distributions so that samples are drawn more efficiently as the iterations progress. The weighted samples form a sample from the posterior distribution under some mild conditions [30, 37].

Ensemble importance sampling schemes also exist; for example, population Monte Carlo (PMC) [10] uses an ensemble to build a mixture or kernel density estimate (KDE) of the posterior distribution. The efficiency of this optimization is restricted by the component kernel(s) chosen and the quality of the current sample from the posterior [8, 15, 16]. Extensions, such as layered adaptive importance sampling [31], adaptive multiple importance sampling (AMIS) [12], and adaptive population importance sampling (APIS) [30], have enabled application of these methods in various fields, including population genetics [44].

In this paper, we present a framework for ensemble importance sampling that can be built around many of the current Metropolis-based methodologies in order to create an efficient target distribution from the current ensemble. The method uses a resampler based on optimal transport, which has been used in the context of particle filters [36]. We also detail how algorithmic parameters can be quickly tuned to optimal regimes, with respect to the effective sample size statistic. In particular, we demonstrate the advantages of this method when attempting to sample from multimodal posterior distributions, such as those arising from inference for mixture models. The method that we present is specifically designed to tackle the challenges of certain types of low-dimensional inverse problems which have complex posterior structure, for example, multimodality or strong correlations, and expensive likelihood evaluations. We will demonstrate that this methodology allows us to sample from such a posterior distribution with greater precision per likelihood evaluation than standard MCMC chains of the same type.

The method considered is similar to PMC methods. which have been previously studied, for example, in [18]. However, our approach exploits current state-of-the-art resamplers and adaptively optimizes algorithmic parameters. Higher accuracy resamplers that provide a deterministic resample optimally reflecting the statistics of the original sample lead to better mixture approximations of the posterior. These approximations can then be used to construct stable and efficient importance sampling proposals. The cost of the state-of-art resamplers can be prohibitive for large ensemble sizes, which are necessary for sampling in moderately higher dimensions, and as such we present a greedy approximation of the optimal transport resampler, which provides us with good results for a fraction of the cost. We also detail how scaling parameters in the MCMC proposals can be quickly tuned, leading to an efficient and fully automated algorithm. We will demonstrate that, despite additional overhead, the ETAIS methodology can outperform its plain Metropolis–Hastings cousin by orders of magnitude to reach a histogram of a particular degree of convergence.

In section 2 we introduce some mathematical preliminaries on which we will later rely. In

Downloaded 11/28/19 to 90.194.4.112. Redistribution subject to CCBY license

© 2019 SIAM and ASA. Published by SIAM and ASA under the terms of the Creative Commons 4.0 license

section 3 we present the general framework of the ETAIS algorithm. In section 4 we consider adaptive versions of ETAIS which automatically tune algorithmic parameters concerned with the proposal distributions. In section 5 we introduce the multinomial transformation (MT) algorithm, which is a less accurate but faster alternative to resamplers that solve the optimal transport problem exactly. In section 6 we consider consistency of the ETAIS algorithm. In section 7 we briefly look at one particular advantageous property of this approach. In section 8 we present some numerical examples, and section 9 contains a brief conclusion and discussion.

2. Preliminaries.

In this section we will introduce preliminary topics and algorithms that will be referred to throughout the paper.

2.1. Bayesian inverse problems.

In this paper, we focus on the use of MCMC methods for characterizing posterior probability distributions arising from Bayesian inverse problems. We wish to learn about a particular unknown quantity x, of which we are able to make direct or indirect noisy observations. For now we say that x is a member of a Hilbert space X.

The quantity x is mapped onto observable space by the observation operator $\mathcal{G} : X \to \mathbb{R}^d$. We assume that the observations, D, are subject to Gaussian noise,

$$D = \mathcal{G}(x) + \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \Sigma). \tag{2.1}$$

These modeling assumptions allow us to construct the likelihood of observing the data D given the quantity $x = x^*$. Rearranging (2.1) and using the distribution of ε, we get

$$\mathbb{P}(D|x = x^*) \propto \exp\left(-\frac{1}{2}\|\mathcal{G}(x^*) - D\|_\Sigma^2\right) = \exp\left(-\Phi(x^*)\right), \tag{2.2}$$

where $\|y_1 - y_2\|_\Sigma^2 = (y_1 - y_2)^\top \Sigma^{-1}(y_1 - y_2)$ for $y_1, y_2 \in \mathbb{R}^d$.

Then, according to Bayes' theorem the posterior density that we are interested in characterizing is given, up to a constant of proportionality, by

$$\pi \propto \exp\left(-\Phi(x^*)\right) \pi_0(x^*),$$

where π_0 is the prior density on the quantity of interest x^*.

2.2. Particle filters and resamplers.

In this subsection we briefly review particle filters, since the development of the resampler that we incorporate into the ETAIS is motivated by this area. Particle filters are a class of Monte Carlo algorithm designed to solve the filtering problem, that is, to find the best estimate of the true state of a system when given only noisy observations of the system. The solution of this problem has been of importance since the middle of the 20th century in fields such as molecular biology, computational physics, and signal processing. In recent years the data assimilation community has contributed several efficient particle filters, including the ensemble Kalman filter (EnKF) [19] and the ensemble transform particle filter (ETPF) [36].

The ETPF defines a coupling T between two random variables Y and X, allowing us to use the induced map as a resampler. An *optimal coupling* T^* maximizes the correlation between X and Y [13]. This coupling is the solution to a linear programming problem in M^2 variables with $2M - 1$ constraints, where M is the size of the sample. Maximizing the correlation preserves the statistics of X in the new sample.

Downloaded 11/28/19 to 90.194.4.112. Redistribution subject to CCBY license

© 2019 SIAM and ASA. Published by SIAM and ASA under the terms of the Creative Commons 4.0 license

In this work we use the resampler used within these particle filters. We approximate the posterior density π, with an ensemble of weighted samples, $\{(w_i, y_i)\}_{i=1}^M$. In filtering problems the weights are found by incorporating newly observed data, whereas here we simply use importance weights. These weighted samples can then be resampled, using the ETPF or otherwise, into a set of new equally weighted samples $\{x_i\}_{i=1}^M$, such that

$$\sum_{i=1}^{M} w_i \delta_{y_i}(\cdot) \quad \xrightarrow{\text{ETPF}} \quad \sum_{i=1}^{M} \frac{1}{M} \delta_{x_i}(\cdot), \tag{2.3}$$

where $\delta_x(\cdot)$ is the Dirac delta measure.

The resamplers used within particle filters such as the ETPF are well suited to this problem because it is easy to introduce conditions into the resampling to ensure we obtain the behavior required. One downside, however, is that the required ensemble size increases quickly with dimension, making it difficult to apply in high-dimensional problems.

2.3. Deficiencies of Metropolis-type MCMC schemes. All MCMC methods are naively parallelizable. One can take a method and simply implement it multiple times over an ensemble of processors. All of the states of all of the ensemble members can be recorded, and in the time it takes one MCMC chain to draw N samples, M ensemble members can draw NM samples.

However, we argue that this is not an optimal scenario. First of all, unless we have a lot of information about the posterior, we will initialize the algorithm's initial state in the tails of the distribution. The samples that are initially made as the algorithm finds its way to the mode(s) of the distribution cannot be considered to be from the target distribution and must be thrown away. This process is known as the burn-in. In a naively parallelized scenario, each ensemble member must perform this process independently, and therefore mass parallelization makes no inroads toward cutting this cost.

Moreover, many MCMC algorithms suffer from poor mixing, especially in multimodal systems. The number of samples needed for an MCMC trajectory to switch between modes can be large, and given that we require a large number of switches before we can get a good idea of the relative probability densities of these different regions, this can be prohibitively expensive.

Another aspect of Metropolis-type samplers is that information computed about a proposed state is simply lost if we choose to reject that proposal in the Metropolis step. An advantage of importance samplers is that no evaluations of $\mathcal{G}$ are ever wasted since all samples are saved along with their relative weightings.

These deficiencies of MCMC methods motivated the development of the ensemble transport adaptive importance sampler (ETAIS). In the next section we will introduce the method in its most general form.

3. The ensemble transform adaptive importance sampler (ETAIS). Importance sampling can be a very efficient method for sampling from a probability distribution. A proposal density is chosen, from which we can draw samples. Each sample is assigned a weight given by the ratio of the target density and the proposal density at that point. This sampling is efficient when the proposal density is concentrated in areas similar to the target density, and

Downloaded 11/28/19 to 90.194.4.112. Redistribution subject to CCBY license

© 2019 SIAM and ASA. Published by SIAM and ASA under the terms of the Creative Commons 4.0 license

they are incredibly inefficient when this is not the case. The aim of the ETAIS is to use an ensemble of states to construct a proposal distribution which will be as close as possible to the target density. If this ensemble is large enough, the distribution of states will be approximately representative of the target density.

The proposal distribution could be constructed in many different ways, but we choose to use a mixture distribution made up of a sum of MCMC proposal kernels, e.g., Gaussians centered at the current state, as in random walk Metropolis–Hastings (RWMH). Once the proposal is constructed, we can sample a new ensemble from the proposal distribution, and each sample is assigned a weight given by the ratio of the target density and the proposal mixture density. Assuming that our proposal distribution is good, the variance of the weights will be small, and we will have many useful samples. Finally, we need to create a set of evenly weighted samples which best represent this set of weighted samples. This is achieved by implementing a resampling algorithm. These samples are not stored in order to characterize the posterior density, since the resampling process is not exact. They are simply needed in order to inform the mixture proposal distribution for the next iteration of the algorithm.

Initially we use the ETPF resampler algorithm [36], although we suggest an alternative strategy in section 5 for examples where a large ensemble is required and for which the ETPF may become more expensive. The resampling algorithm gives us a set of evenly weighted samples which represents the target distribution well and which can be used to iterate the process again. The algorithm is summarized in Algorithm 1.

The choice of resampling algorithm is important since its output is used to formulate the proposal distribution for the next iteration. A basic resampler may be cheap to implement, but the resulting sample and, in turn, the next iteration's proposal distribution, may not be as representative of the target distribution. This leads to higher variances of the weights in the importance sampler and therefore to slower convergence to the target density.

We wish to sample states $x \in X$ from a posterior probability distribution with density π. Since we have M ensemble members, we represent the current state of all of the Markov chains as a vector $\mathbf{X} = [x_1, x_2, \ldots, x_M]^\top$. We are also given a transition kernel $\nu(\cdot, \cdot)$, which might come from an MCMC method, for example, the RWMH proposal density $\nu(\cdot, x) \sim \mathcal{N}(x, \beta^2)$, where $\beta^2 \in \mathbb{R}$ defines the variance of the proposal.

Since the resampling does not give us a sample statistically identical to that which is input, we cannot assume that the samples $\mathbf{X}^{(i)}$ are from the posterior. Therefore, as with serial importance samplers, the weighted samples $(\mathbf{W}, \mathbf{Y})_{i=1}^N$ are from the posterior that we will analyze.

In each iteration of the algorithm, we are required to compute the likelihood for each member of the ensemble. In the case where the likelihood is expensive to compute, for example, because it requires the numerical approximation of an ODE or PDE and/or there is a very large amount of observations in the data, it will be expedient to spread this computational effort across multiple cores. Whether this is efficient will depend on the architecture of the machine and on the cost of communication across the cores relative to the cost of the likelihood evaluation. Evaluation of the denominator in the weights also contributes to the overhead costs of this approach beyond a standard MCMC method, but this too can be parallelized efficiently, depending on communication costs of the architecture of the machine, for example, reducing the $\mathcal{O}(M^2)$ cost to $\mathcal{O}(M)$ over M cores.

Downloaded 11/28/19 to 90.194.4.112. Redistribution subject to CCBY license

© 2019 SIAM and ASA. Published by SIAM and ASA under the terms of the Creative Commons 4.0 license

Algorithm 1: The ensemble transform adaptive importance sampler (ETAIS).

Set $\mathbf{X}^{(0)} = \mathbf{X}_0 = [x_1^{(0)}, x_2^{(0)}, \ldots, x_M^{(0)}]^\top$.
for $i = 1, \ldots, N$ **do**
  Sample $\mathbf{Y}^{(i)} = [y_1^{(i)}, y_2^{(i)}, \ldots, y_M^{(i)}]^\top, \quad y_j^{(i)} \sim \nu(\cdot; x_j^{(i-1)})$.
  Calculate $\mathbf{W}^{(i)} = [w_1^{(i)}, w_2^{(i)}, \ldots, w_M^{(i)}]^\top, \quad w_j^{(i)} = \frac{\pi(y_j^{(i)})}{\chi(y_j^{(i)}; \mathbf{X}^{(i-1)})}$, where

$$\chi(\cdot; \mathbf{X}^{(i-1)}) = \frac{1}{M} \sum_{j=1}^{M} \nu(\cdot; x_j^{(i-1)}).$$

  Resample: $(\mathbf{W}^{(i)}, \mathbf{Y}^{(i)}) \to (\frac{1}{M}\mathbf{1}, \mathbf{X}^{(i)})$.
Output $(\mathbf{W}, \mathbf{Y})$.

The key is to choose a suitable transition kernel ν such that if $\mathbf{X}^{(i)}$ is a representative sample of the posterior, then the mixture density $\chi(\cdot; \mathbf{X}^{(i)})$ is a good approximation of the posterior distribution. If this is the case, the newly proposed states $\mathbf{Y}^{(i)}$ will also be a good sample of the posterior with low variance in the weights $\mathbf{W}^{(i)}$.

In section 8, we will demonstrate how the algorithm performs, primarily using random walk (RW) proposals. We do not claim that this choice is optimal—it is simply chosen as an example to show that sharing information across ensemble members can improve the original Metropolis–Hastings (MH) algorithm and lead to tolerances being achieved in fewer evaluations of $\mathcal{G}$. This is important because if the inverse problem being tackled involves computing the likelihood from a very large data set, or if the likelihood requires the numerical solution of a differential equation, this could lead to a large savings in computational cost. We have observed that using more complex (and hence more expensive) kernels ν does not significantly improve the convergence speed of the algorithm for posteriors that we have considered, although kernels such as those used in MALA can be more stable for certain problems [42].

Care needs to be taken when choosing kernel(s) for the proposal distribution to ensure that the overall mixture is absolutely continuous with respect to the posterior distribution. For example, in many inverse problems we may be looking for the value of certain physical parameters, which may be strictly positive or even bounded on an interval. In this case, we should choose proposal kernels ν that have the same support as the target. In section 8.3 we present an example of an unknown parameter being bounded where β-distributed proposals are made to ensure that the distribution is supported on $[0, 1]$.

It is possible, especially during early stages of the algorithm where the mixture proposal is a poor approximation of the target distribution, for a sample to be produced with a weight orders of magnitude bigger than weights of the other samples in the ensemble. This is a problem because the resampling step will then lead to a vector of samples all centered at the outlier sample. Fortunately these spikes in weight can easily be detected, and in such a case the sample for this iteration can be removed. The ETPF can then be used to formulate a new mixture sample, with one sample placed at the outlier point and the others distributed

Downloaded 11/28/19 to 90.194.4.112. Redistribution subject to CCBY license

© 2019 SIAM and ASA. Published by SIAM and ASA under the terms of the Creative Commons 4.0 license

according to the sample positions before the problematic iteration. This ensures that the region where the problem occurred is better represented by the mixture, and it prevents further spikes in this region.

4. Automated tuning of algorithm parameters. Efficient selection of scaling parameters in MCMC algorithms is critical to achieving optimal mixing rates and hence achieving fast convergence to the target density. It is well known that a scaling parameter which is either too large or too small results in a Markov chain with high autocorrelation. One worthy aspect of the ETAIS is that it finds an appropriate proposal kernel ν such that the mixture distribution χ is a close approximation to the posterior density π.

Most MCMC proposals have parametric dependence, which allows the user to control their variance. For example, in the RW proposal $y = x + \beta\eta$, the parameter β is the standard deviation of the proposal distribution. Therefore the proposal distributions can be tuned such that they are slightly overdispersed. This tuning can take place during the burn-in phase of the algorithm. Algorithms that use this method to find optimal proposal distributions are known as adaptive MCMC algorithms and have been shown to be convergent provided that they satisfy certain conditions [40, 41]. Alternatively, a costly trial-and-error scheme with short runs of the MCMC algorithm can be used to find an acceptable value of β.

Algorithms that use mixture proposals, e.g., ETAIS, must tune the variance of the individual kernels within the proposal mixture. This adaptivity during early iterations has some added benefits beyond finding an optimal parameter regime for the algorithm. If the initial value of the proposal variance is chosen to be very large, then the early mode-finding stages of the algorithm are expedited. Adaptively reducing the proposal variances to an optimal value then allows us to explore each region efficiently. Using an ensemble of chains allows quick and effective assessment of the value of the optimal scaling parameter.

In many MCMC algorithms, such as the RWMH, the optimal scaling parameter can be found by searching for the parameter value which gives an optimal acceptance rate; e.g., for near Gaussian targets the optimal rates are 23.4% for RWMH and 57.4% for MALA [39]. This method is not applicable to ETAIS, so we must use other statistics to optimize the scaling parameter. Section 4.1 gives some possible methods for tuning β.

4.1. Statistics for determining the optimal scaling parameter.

4.1.1. Determining optimal scaling parameter using error analysis. When available, an analytic form for the target distribution allows us to assess the convergence of sampling algorithms to the target distribution. Common metrics for this task include the relative error between the sample moments and the target's moments, or the relative L^2 error between the sample histogram and the target density, $\pi(x|D)$. The relative error in the nth moment, $\hat{m}_n$, is given by

$$e = \left| \frac{\hat{m}_n - \mathbb{E}[X^n]}{\mathbb{E}[X^n]} \right|, \quad \text{where} \quad \hat{m}_n = \frac{1}{N} \sum_{i=1}^{N} x_i^n, \tag{4.1}$$

and $\{x_i\}_{i=1}^N$ is a sample of size N.

The relative L^2 error, E, between a continuous density function and a piecewise constant approximation of that density can be defined by considering the following difference in mass

Downloaded 11/28/19 to 90.194.4.112. Redistribution subject to CCBY license

© 2019 SIAM and ASA. Published by SIAM and ASA under the terms of the Creative Commons 4.0 license

between the self-normalized histogram of the samples and the posterior distribution over a set of disjoint sets or "bins":

$$E^2 = \sum_{i=1}^{n_b} \left[\int_{R_i} \pi(s|D)\,\mathrm{d}s - vB_i \right]^2 \Big/ \sum_{i=1}^{n_b} \left[\int_{R_i} \pi(s|D)\,\mathrm{d}s \right]^2, \tag{4.2}$$

where the regions $\{R_i\}_{i=1}^{n_b}$ are the d-dimensional histogram bins, so that $\bigcup_i R_i \subseteq X$ and $R_i \cap R_j = \emptyset$, n_b is the number of bins, v is the volume of each bin, and B_i is the value of the ith bin. This metric converges to the standard definition of the relative L^2 error as $v \to 0$.

In general, these statistics cannot be used to find optimal values of β because they require the analytic solution and are expensive to approximate. However, they can be used to assess the ability of other indicators to find the optimal scaling parameters in a controlled setting.

4.1.2. The effective sample size. The effective sample size, n_{eff}, can be used to assess the efficiency of importance samplers. Ideally, in each iteration, we would like all M of our samples to provide new information about the posterior distribution. In practice, we cannot achieve a perfect, effective sample size of M.

The effective sample size of a weighted sample is defined in the following way:

$$n_{\text{eff}} = \frac{\left(\sum_{i=1}^M w_i\right)^2}{\sum_{i=1}^M w_i^2} \approx \frac{M\mathbb{E}(w)^2}{\mathbb{E}(w^2)} = M\left(1 - \frac{\text{var}(w)}{\mathbb{E}(w^2)}\right).$$

The approximation is exact in the limit $M \to \infty$. From the last expression, if the variance of the weights is zero, then $n_{\text{eff}} = M$; this is our ideal scenario. In the limit $M \to \infty$, maximizing the effective sample size is equivalent to minimizing the variance of the weights.

The statistic n_{eff} is easier to deal with than the variance of the weights, as it takes values on $[1, M]$, while the variance of the weights takes values on $\mathbb{R}^+$. Moreover the variance of the weights can vary over many orders of magnitude, causing numerical instabilities, so that the effective sample size is more desirable as an indicator of optimality.

In this paper we tune our simulations using the effective sample size statistic. We calculate this statistic using a sample size of Mn_k, where $1 \leq n_k \leq N$ is sufficiently large enough to obtain a reasonable estimate of n_{eff} with scaling parameter δ_k. Calculating n_{eff} over these subsets of the simulations tends to underestimate the optimal value of the scaling parameter due to the possibility of missing unlikely proposals with extreme weights.

The effective sample size also has another useful property: if we consider the algorithm in the early stages, then we have all M samples in the tails of the target searching for a mode. The sample closest to the mode will have an exponentially higher weight, and the effective sample size will be close to 1. In later stages the ensemble populates high density regions and better represents the posterior distribution. This leads to a smaller variation in the weights and a larger effective sample size. By looking for approximations of the effective sample size which look like a stationary distribution, we can tell when the algorithm is working efficiently.

Figure 1 demonstrates that the effective sample size flattens out as the relative L^2 difference between the posterior distribution and the proposal distribution stabilizes close to its minimum.

Downloaded 11/28/19 to 90.194.4.112. Redistribution subject to CCBY license

© 2019 SIAM and ASA. Published by SIAM and ASA under the terms of the Creative Commons 4.0 license

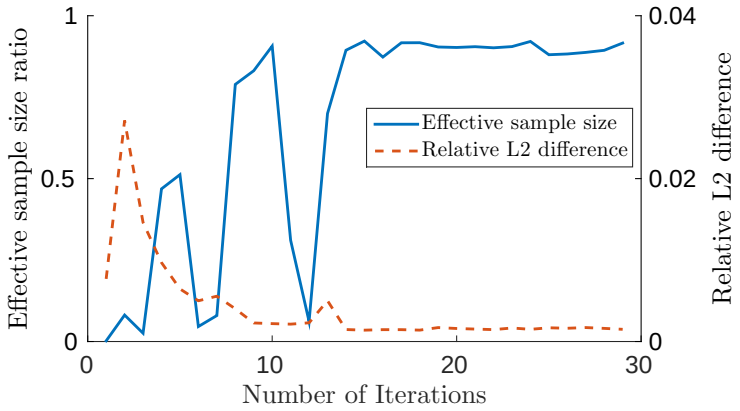
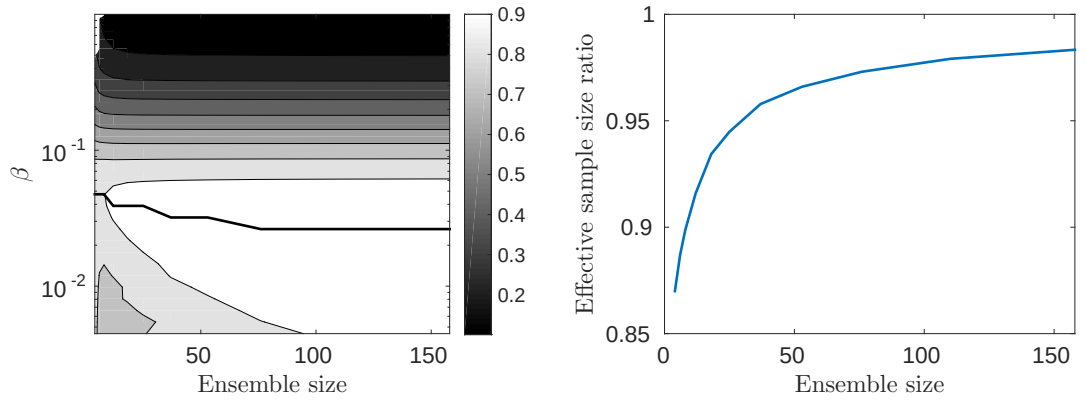


Figure 1. *The effective sample size ratio and relative L^2 difference E during the first 30 iterations. These numerics are taken from the example in section 8.3.3 using the ETAIS-RW algorithm.*



(a) Contours showing optimal ranges of the scaling parameter.

(b) The highest effective sample size achieved for each ensemble size.

Figure 2. *Behavior of the effective sample size as the ensemble size increases, considering the example in section 8.1 using the ETAIS-RW algorithm.*

Figure 2 shows how the effective sample size used in ETAIS is affected by the ensemble size. We see from subfigure (a) that as the ensemble size increases, the optimal scaling parameter decreases. This is expected since the larger ensemble allows for finer resolution in the proposal distribution approximation of the posterior distribution. We also see that the algorithm becomes less sensitive to changes in the scaling parameter as the ensemble size increases. Subfigure (b) shows that as the ensemble size increases, the algorithm becomes more efficient.

4.2. Adaptively tuned ETAIS. To adapt the scaling parameter β, we use a version of the gradient ascent method modified for a stochastic function. Other, more sophisticated examples are described in [1, 25, 41].

Downloaded 11/28/19 to 90.194.4.112. Redistribution subject to CCBY license

© 2019 SIAM and ASA. Published by SIAM and ASA under the terms of the Creative Commons 4.0 license

Algorithm 2: Adaptively tuned ETAIS algorithm.

1 Define update times $\{n_k\}_k$.
2 **for** $n = 1, \ldots, N$ **do**
3 Complete steps 3–5 of Algorithm 1.
4 **if** $n \in \{n_k\}$ **then**
5 Divide ensemble into two halves. Use these halves to estimate the gradient in n_{eff} at β_k.
6 Update β_k using gradient ascent,

$$\beta_{k+1} = \beta_k + \gamma \nabla n_{\text{eff}}.$$

Our adaptive algorithm is given in Algorithm 2. We choose update times which, as suggested in section 4.1.2, allow for a reasonable estimate of the effective sample size but do not waste too many iterations. In step 6, the gradient ascent parameter γ may decrease over time, e.g., as a function of n_k.

In the context of an expensive likelihood evaluation, slow convergence of the algorithmic parameters in this adaptive regime would be a concern. However, this is rarely an issue, due in part to the speed at which the ETAIS algorithm completes the burn-in phase. Figures 3(a)–(d) demonstrate this for two different versions of the ETAIS methodology with an RW proposal and an MALA [38] proposal. This represents our typical experience, with the scaling parameters converging in $\mathcal{O}(10^2)$ iterations to an optimal regime.

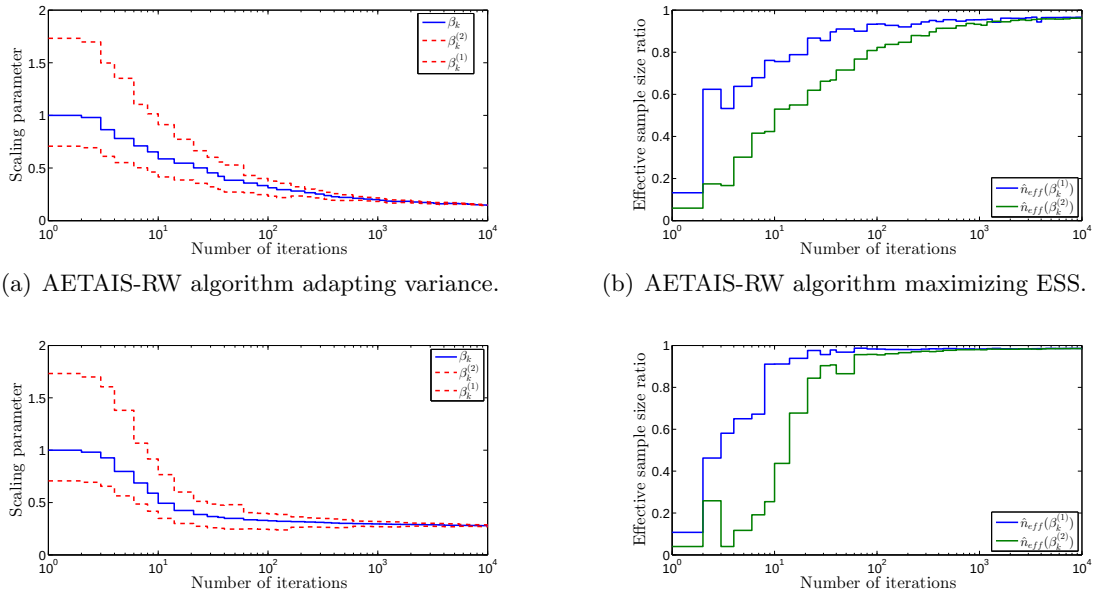
5. Multinomial transformation. Although the ETPF produces the optimal linear coupling, it can also become quite costly as the number of ensemble members is increased. It is arguable that in the context of ETAIS, we do not require this degree of accuracy, and that a faster, more approximate method for resampling could be employed. One approach would be to use the bootstrap resampler, which simply takes the M ensemble members' weights and constructs a multinomial distribution, from which M samples are drawn. This is essentially the cheapest resampling algorithm that one could construct. However, it too has some drawbacks. The algorithm is random, and as such it is possible that all of the ensemble members in a particular region are not sampled. This could be particularly problematic when attempting to sample from a multimodal distribution, where it might take a long time to find one of the modes. The bootstrap filter, unlike the ETPF, is also not guaranteed to preserve the mean of the weighted sample.

Ideally, we would like to use a resampling algorithm that is not prohibitively costly for moderately sized or large ensembles, which preserves the mean of the samples and makes it much harder for the new samples to forget a significant region in the density. This motivates the following algorithm, which we refer to as the multinomial transformation (MT) and which is a greedy approximation of the ETPF resampler.

Instead of sampling M times from an M-dimensional multinomial distribution as in the

Downloaded 11/28/19 to 90.194.4.112. Redistribution subject to CCBY license

© 2019 SIAM and ASA. Published by SIAM and ASA under the terms of the Creative Commons 4.0 license



(a) AETAIS-RW algorithm adapting variance.

(b) AETAIS-RW algorithm maximizing ESS.

(c) AETAIS-MALA algorithm adapting variance.

(d) AETAIS-MALA algorithm maximizing ESS.

Figure 3. *Demonstration of the convergence of the adaptive scaling parameter in the adaptively tuned ETAIS algorithms.*

case of the bootstrap algorithm, we sample once from each of M different multinomials. Suppose that we have M samples y_n with weights w_n. The multinomial sampled from the bootstrap filter has a vector of probabilities given by

$$\frac{1}{\sum w_n}[w_1, w_2, \ldots, w_M] = \bar{\mathbf{w}},$$

with associated states y_n. We wish to find M vectors $\{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_M\} \subset \mathbb{R}^M_{\geq 0}$ such that $\frac{1}{M}\sum \mathbf{p}_i = \bar{\mathbf{w}}$. The MT is then given by a sample from each of the multinomials defined by the vectors $\mathbf{p}_i = [p_{i,1}, p_{i,2}, \ldots, p_{i,M}]$ with associated states $\mathbf{y}_i$. Alternatively, as with the ETPF, a deterministic sample can be chosen by choosing each sample to be equal to the mean value of each of these multinomial distributions; i.e., each new sample $\hat{x}_i$ is given by

$$\hat{x}_i = \sum p_{i,j} x_j, \qquad i \in \{1, 2, \ldots, M\}. \tag{5.1}$$

The resulting sample has several properties which are advantageous when used with the ETAIS algorithm. First, we have effectively chopped the multinomial distribution used in the bootstrap filter into M pieces, and we can guarantee that exactly one sample will be taken from each section. This leads to a much smaller chance of losing entire modes in the density if each of the submultinomials is chosen in an appropriate fashion. Second, if we do not make a random sample for each multinomial with probability vector $\mathbf{p_i}$ but instead take the mean of the multinomial to be the sample, then this algorithm preserves the mean of the sample

Downloaded 11/28/19 to 90.194.4.112. Redistribution subject to CCBY license

© 2019 SIAM and ASA. Published by SIAM and ASA under the terms of the Creative Commons 4.0 license

Algorithm 3: The multinomial transformation (MT).

$\mathbf{z} = M\bar{\mathbf{w}}$.
for $i = 1, \ldots, M$ **do**
  $J = \arg\max_j z_j$.
  $p_{i,J} = \min\{1, z_J\}$.
  $z_J = z_J - p_{i,J}$.
  while $\sum_j p_{i,j} < 1$ **do**
    $K = \arg\min_{k\in\{k|z_k>0\}} \|y_J - y_k\|$.
    $p_{i,K} = \min\{1 - \sum_j p_{i,j}, z_K\}$.
    $z_K = z_K - p_{i,K}$.
  $x_i = \sum_k p_{i,k} y_k$.

exactly. Lastly, and as we will see shortly, this algorithm is significantly less computationally intensive than the ETPF.

There are of course infinitely many different ways to split the original multinomial into M parts, some of which will be far from optimal. The method we have chosen is loosely based on the idea of optimal transport. We search out states with the largest weights and choose a cluster around these points based on the closest states geographically. This method is not optimal since once most of the clusters have been selected, the remaining states may be spread across the parameter space.

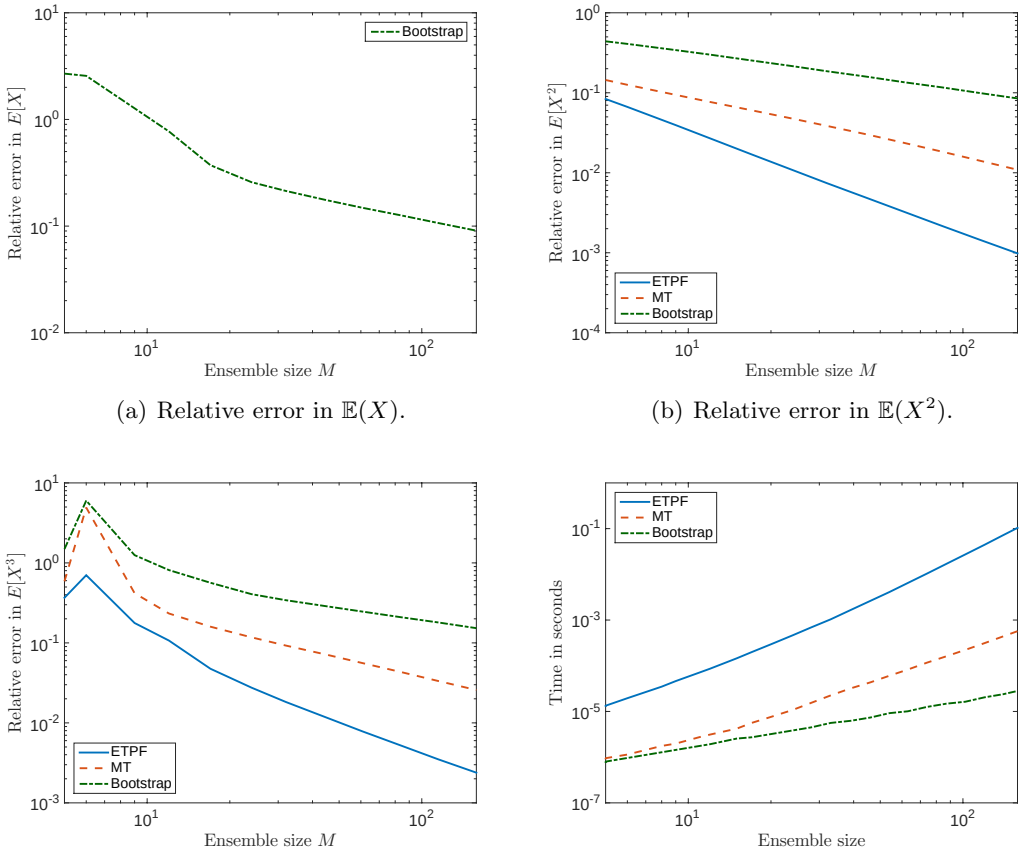
Algorithm 3 describes the basis of the algorithm with deterministic resampling, using the means of each of the submultinomials as the new samples. This resampler was designed to be numerically cheaper than the ETPF and more accurate than straight multinomial resampling. Therefore we now present numerical examples which demonstrate this.

To test the accuracy and speed of the three resamplers (ETPF, bootstrap, and MT), we drew a sample of size M from the proposal distribution $\mathcal{N}(1, 2)$. Importance weights were assigned, based on a target distribution of $\mathcal{N}(2, 3)$. The statistics of the resampled outputs were compared with the original weighted samples. Figures 4(a)–(c) show how the relative errors in the first three moments of the samples changes with ensemble size M for the three different resamplers. As expected, the MT lies somewhere between the high accuracy of the ETPF and the less accurate bootstrap resampling. Note that only the error for the bootstrap multinomial sampler is presented for the first moment since both ETPF and MT preserve the mean of the original weighted samples up to machine precision. Figure 4(d) shows how the computational cost, measured in seconds, scales with the ensemble size for the three different methods, where timings have been taken from simulations on a Dell server with four 8-core 3.3 GHz CPUs and 64 Gb memory. These results demonstrate that MT behaves to our satisfaction and, more importantly, ensures that exactly one sample of the output will lie in each region with weights up to $\frac{1}{M}$ of the total.

We will use MT in the numerics in section 8.3, where we have chosen to use a larger ensemble size. We do not claim that MT is the optimal choice within ETAIS, but it does have favorable features and demonstrates how different choices of resampler can affect the speed and efficiency of the ETAIS algorithm.

Downloaded 11/28/19 to 90.194.4.112. Redistribution subject to CCBY license

© 2019 SIAM and ASA. Published by SIAM and ASA under the terms of the Creative Commons 4.0 license



(a) Relative error in $\mathbb{E}(X)$.

(b) Relative error in $\mathbb{E}(X^2)$.

(c) Relative error in $\mathbb{E}(X^3)$.

(d) Cost of resamplers per iteration.

Figure 4. *Comparison of the performance between different resampling schemes. The example in section 8.1 is implemented for this demonstration.*

6. Consistency of ETAIS.

As with all importance sampling schemes, we must have absolute continuity of target distribution with respect to the proposal distribution if we wish to achieve convergence. This can usually be ensured by choosing the proposal kernels ν to be from the same distribution type as the prior distribution.

As outlined in [30], consistency of population AIS algorithms can be considered in two different cases. In the first case, we fix the number of iterations at $N < \infty$ but allow the population size to grow to infinity, $M \to \infty$. In the second case, we hold the population size fixed at $M < \infty$ and allow infinitely many iterations, $N \to \infty$.

In the first case, we know from standard importance sampling results that for an iteration n, as $M \to \infty$, we obtain a consistent estimator for any statistic of interest,

$$\hat{\phi}(\Theta) \approx \frac{1}{\hat{Z}} \sum_{i=1}^{M} \frac{1}{M} w_i \phi(\theta_i) \to \phi(\Theta),$$

where the normalization constant, $\hat{Z} = \frac{1}{M} \sum_{i=1}^{M} w_i$, also converges to the true normalization constant Z [37].

Downloaded 11/28/19 to 90.194.4.112. Redistribution subject to CCBY license

© 2019 SIAM and ASA. Published by SIAM and ASA under the terms of the Creative Commons 4.0 license

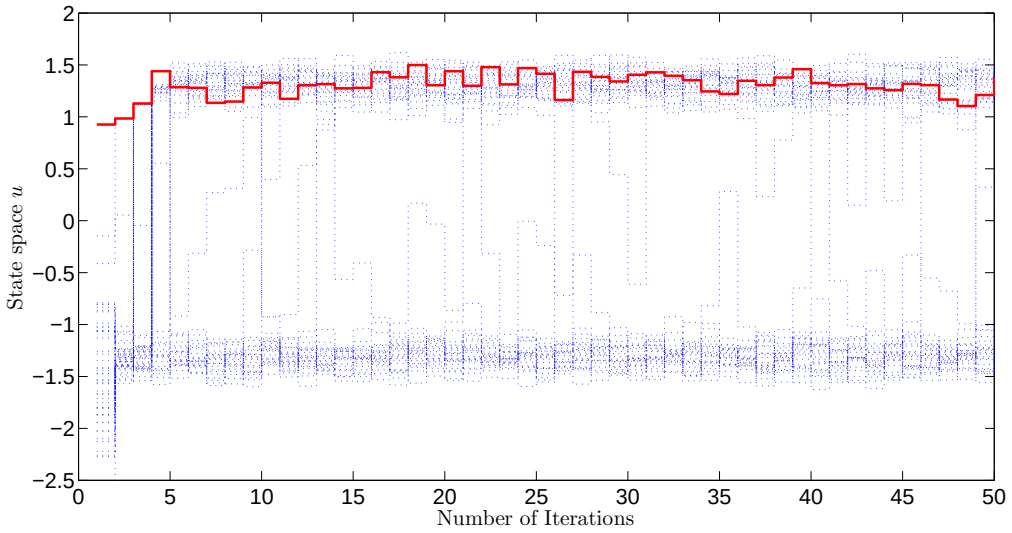


Figure 5. *This figure demonstrates the redistribution property of the ETAIS algorithm for a bimodal target with equally weighted modes. Initially there are one chain in the positive mode and 49 chains in the negative mode.*

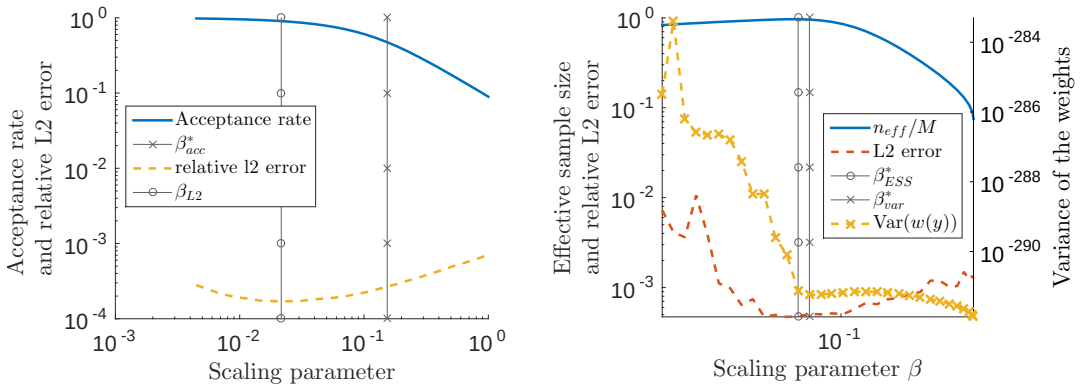
The second case is slightly more involved. Estimation of the normalization constant Z is biased, and so estimates of statistics are sums of independent but biased estimators. Since the estimators are independent, the proof of consistency of ETAIS in this second limit can be approached in the same way as the PMC algorithm, where it has been shown that $\hat{Z} \to Z$ [37]. Since the normalization constant is consistent, sums of the independent estimators are also consistent.

7. A useful property of the ETAIS algorithm for multimodal distributions. The biggest issue for the MH algorithms when sampling from multimodal posterior distributions is that frequent switches between the modes are required if the method is to converge efficiently. The ETAIS algorithm tackles this problem with its resampling step. The algorithm uses its dynamic mixture proposal distribution to build up an approximation of the posterior at each iteration, and then compares this to the posterior distribution via the weights function. Any large discrepancy in the approximation will result in a large or small weight being assigned to the relevant chain, meaning the chain will either pull other chains toward it or be pushed toward a chain with a larger weight. In this way, the algorithm allows chains to "teleport" to regions of the posterior which are in need of more exploration. Figure 5 shows the trace of a simulation of the ETAIS-RW algorithm for a bimodal example with equally weighted modes, with initially 1 chain in the positive mode and 49 chains in the negative mode. It takes only a handful of iterations for the algorithm to balance out the chains into 25 chains in each mode. The chains switch modes without having to climb the energy gradient in the middle.

8. Numerical examples. All numerical examples in this section were computed on a single core of a Dell server with four 8-core 3.3G Hz CPUs and 64 Gb memory. Throughout, we

Downloaded 11/28/19 to 90.194.4.112. Redistribution subject to CCBY license

© 2019 SIAM and ASA. Published by SIAM and ASA under the terms of the Creative Commons 4.0 license



(a) RWMH algorithm. (b) ETAIS-RW algorithm.

Figure 6. *Finding optimal values of β for a 1D Gaussian density for* (a) *the RWMH algorithm, and* (b) *for the ETAIS-RW algorithm.*

measure the efficiency of the algorithms in terms of the number of iterations (i.e., likelihood evaluations) required in order to reach a given order of accuracy. The reason for this is that this method is designed with a particular type of challenging inverse problem in mind, namely one which is low-dimensional and has an expensive likelihood (such as a PDE solve and/or very large data) that dwarfs the overhead cost of ETAIS, and which could have complex posterior structure, such as correlations, multimodality, or other highly non-Gaussian features.

The numerics contained herein compare the presented ETAIS approach against MH methods with the same type of proposal as is used by the kernels within ETAIS. Since we propose that ETAIS is a potential framework for parallelized Bayesian computations, we compare ETAIS with N ensemble members against an independent ensemble of N MH chains. All implementations were computed in serial, but the results demonstrate a clear speed-up, which could be further exploited through parallelization.

8.1. Automated variance tuning. In this first example, we target a simple 1D Gaussian distribution in order to show the approximate equivalence of optimizing the variance of the proposal kernels using the effective sample size as well as the L^2 error, which of course is not available to us in practical applications.

Figure 6(a) shows the two values of β which are optimal according to the acceptance rate and relative L^2 error criteria for the RWMH algorithm. The smaller estimate comes from the relative L^2 error, and the larger from the acceptance rate. The results in Figure 6 are summarized in Table 1. Since in general we cannot calculate the relative L^2 error, we must optimize the algorithm using the acceptance rate. From the relative L^2 error curve we can see that the minimum is very wide, and, despite the optimal values being very different, there is not a large difference in the convergence rate.

Figure 6(b) shows the effective sample size ratio compared to the error analysis and to the variance of the weights. The relative L^2 error graph is noisy, but it is clear that the maximum in the effective sample size and the minimum in the variance of the weights are both close to

Downloaded 11/28/19 to 90.194.4.112. Redistribution subject to CCBY license

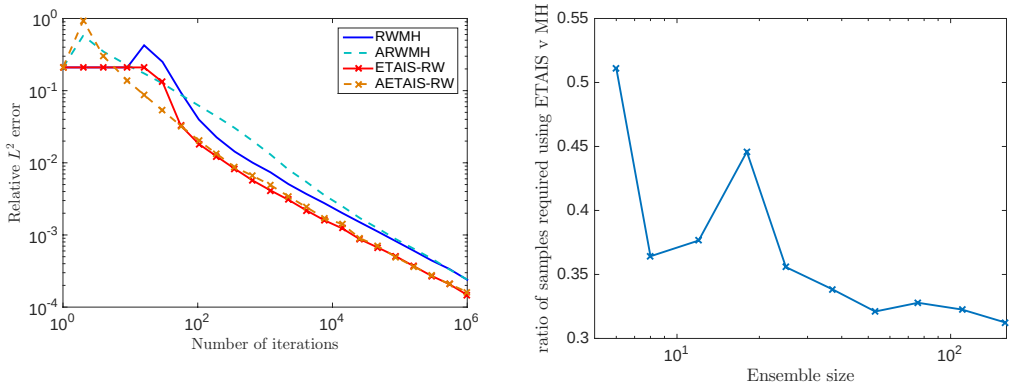
© 2019 SIAM and ASA. Published by SIAM and ASA under the terms of the Creative Commons 4.0 license

Table 1

Optimal values of β summarized from Figure 6. Statistics are calculated as described in section 4.1. The values β^*_{L2} and $\beta^*_{\%}$ are the optimal scaling parameters found by optimizing the relative L^2 errors and acceptance rate, respectively. Similarly, β^*_{eff} and $\beta^*_{var(w(y))}$ optimize the effective sample size and variance of the weight statistics.

Statistic	RWMH
β^*_{L2}	2.1e-2
$\beta^*_{\%}$	1.5e-1
Acceptance rate (β^*_{L2})	9.0e-1
Acceptance rate ($\beta^*_{\%}$)	5.0e-1

Statistic	ETAIS-RW
β^*_{eff}	4.7e-2
$\beta^*_{\mathrm{var}(w(y))}$	5.8e-2
β^*_{L2}	3.9e-2



(a) Convergence. (b) Scaling of ETAIS-RW.

Figure 7. (a) *Convergence of the (A)RWMH and (A)ETAIS-RW algorithms for a 1D Gaussian, computed using 50 ensemble members and the ETPF resampler.* (b) *Ratio of ETAIS-RW samples required to reach the same tolerance as the RWMH algorithm for a range of ensemble sizes.*

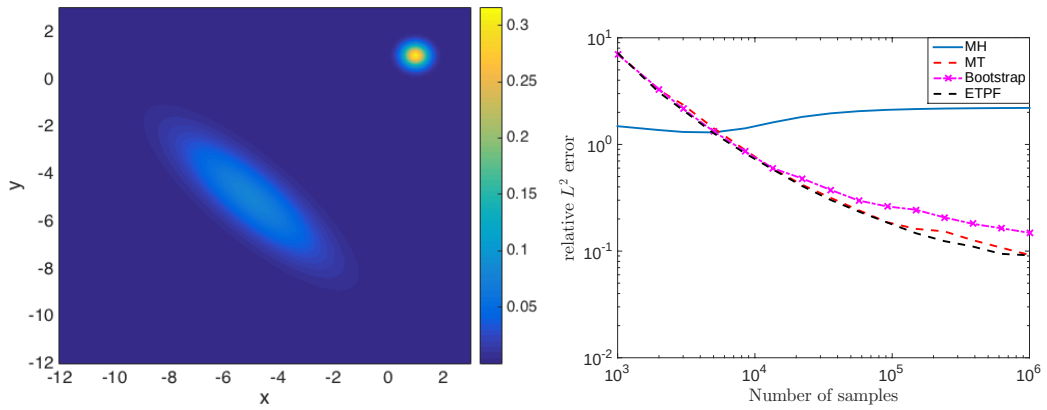
the minimum in the relative L^2 error. Due to this we say that the estimate of the effective sample size found by averaging the statistic over each iteration is a good indicator of the optimal scaling parameter.

Figure 7(a) shows that the ETAIS-RW algorithm converges to the posterior distribution significantly faster than the RWMH algorithm in both L^2 error and relative error in the moments. The adaptive algorithms are also shown. We see that both adaptive algorithms converge to the posterior at a similar speed to the respective optimized algorithm. This shows that, particularly for ETAIS, we can optimize simulations efficiently on the fly.

Figure 7(b) was produced by identifying the optimal value of the scaling parameter for a range of ensemble sizes and then computing the ratio of ETAIS samples needed in comparison with RWMH samples for the same level of error. The decreasing trend shows superlinear improvement of ETAIS with respect to ensemble size, in terms of the number of iterations required, which is a demonstration of our belief that communication between the ensemble members should give us added value over and above that provided by naive parallelism. This decrease is due to the increasing effective sample size shown in Figure 2(b).

Downloaded 11/28/19 to 90.194.4.112. Redistribution subject to CCBY license

© 2019 SIAM and ASA. Published by SIAM and ASA under the terms of the Creative Commons 4.0 license



(a) Target density. (b) Comparison of resamplers.

Figure 8. (a) *Bimodal target density.* (b) *Convergence of RWMH and ETAIS-RW with ETPF, MT, and bootstrap resamplers.*

8.2. Multimodal targets and the effect of resampler quality. In this second example we investigate the behavior of the ETAIS algorithm when applied to a bimodal problem, given by a mixture of two Gaussians,

$$\pi(x) = 0.2\pi_1(x) + 0.8\pi_2(x),$$

where π_1 is the density of an

$$\mathcal{N}\left(\begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}0.1 & 0\\0 & 0.1\end{pmatrix}\right)$$

random variable and π_2 is the density of an

$$\mathcal{N}\left(\begin{pmatrix}-5\\-5\end{pmatrix}, \begin{pmatrix}2.75 & -2.25\\-2.25 & 2.75\end{pmatrix}\right)$$

random variable.

MH methods can struggle with multimodal problems, particularly where switches between the modes are rare, resulting in incorrectly proportioned modes in the histograms. This example demonstrates that the ETAIS algorithm redistributes samples to new modes as they are found. This means that we expect the number of samples in a mode to be approximately proportional to the probability density in that mode, resulting in faster global convergence. In particular, we will look at the effect of using the ETPF, MT, or a standard bootstrap resampler.

Figure 8 shows the bimodal target density and convergence plots for the RWMH and ETAIS-RW with ETPF, MT, and bootstrap resamplers averaged over 32 repeats. As expected, a higher quality resampler leads to better proposal distributions, which in turn leads to greater stability and more reliable convergence. However, this also shows that MT is a good greedy approximation of the ETPF, and at a fraction of the cost when the number of ensemble members is larger, as shown in section 5. The RWMH algorithm fails to converge efficiently

Downloaded 11/28/19 to 90.194.4.112. Redistribution subject to CCBY license

© 2019 SIAM and ASA. Published by SIAM and ASA under the terms of the Creative Commons 4.0 license

since chains only very rarely make switches between the two modes, leading to very slow mixing. This demonstrates the advantage that ensemble-based methods can have over other methods when the target is multimodal.

8.3. A mixture model. The technique of mixture modeling employs well-known parametric families to construct an approximating distribution, which may have a complex structure. Most commonly, Gaussian kernels are used since underlying properties in the data can often be assumed to follow a Gaussian distribution. An example would be a practitioner measuring the heights of 100 adults but failing to record their gender. The data could be considered as one population with two subpopulations, male and female. The problem then might be to determine the average height of adult males from the data. In this case, since height is often considered to follow a Gaussian distribution, it makes sense to model the population as a mixture of two univariate Gaussian distributions.

A well-known problem in the Bayesian treatment of mixture modeling is that of identification, sometimes referred to as the label-switching phenomenon. The likelihood distribution for mixture models is invariant under permutations of the mixture labels. If a mixture has n means and the point $(\mu_1, \dots, \mu_n)$ maximizes the likelihood, then the likelihood will also be maximized by $(\mu_{\varphi(1)}, \dots, \mu_{\varphi(n)})$ for all permutations $\varphi(\cdot)$. This means that the number of modes in the posterior distribution is of order $\mathcal{O}(n!)$. As we have seen, standard MH algorithms may have difficulty obtaining reliable inference for posterior distributions with a large number of modes—or even a small number of modes which are separated by a large distance.

8.3.1. Target distribution. In particular, we look at a data set where we assume that there are two subpopulations within the overall population. Since both subpopulations will be approximated by Gaussian distributions we have five parameters which need to be estimated: two means $\{\mu_1, \mu_2\}$, two variances $\{\sigma_1^2, \sigma_2^2\}$, and the probability p that an individual observation belongs to the first subpopulation. We have 100 data points, D_i, which we assume to be distributed according to

$$D_i \sim p\mathcal{N}(\mu_1, \sigma_1^2) + (1-p)\mathcal{N}(\mu_2, \sigma_2^2), \quad i = 1, \dots, 100,$$

where $p \in [0,1]$, $\mu_1, \mu_2 \in \mathbb{R}$, and $\sigma_1^2, \sigma_2^2 \in \mathbb{R}^+$. Due to the domains of these parameters, as well as some prior knowledge, we assign the priors

$$p \sim \text{Beta}(1,1), \quad \mu_{1,2} \sim \mathcal{N}(0,4), \quad \text{and} \quad \sigma_{1,2}^2 \sim \text{Gamma}(\alpha = 2, \beta = 1).$$

If we collect these parameters in the vector $\theta = (p, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2)^\top$, the resulting posterior distribution is

$$\pi(\theta|D) \propto \prod_{i=1}^{100} (p\mathcal{N}(D_i; \mu_1, \sigma_1^2) + (1-p)\mathcal{N}(D_i; \mu_2, \sigma_2^2)) \prod_{i=1}^{5} \pi_0^i(\theta_i), \tag{8.1}$$

where $\pi_0^i(\cdot)$ is the prior density function corresponding to θ_i.

Figure 9 presents a visualization of the posterior distribution for this problem, created from 10 million samples produced by the ETAIS-RW algorithm.

Downloaded 11/28/19 to 90.194.4.112. Redistribution subject to CCBY license

© 2019 SIAM and ASA. Published by SIAM and ASA under the terms of the Creative Commons 4.0 license

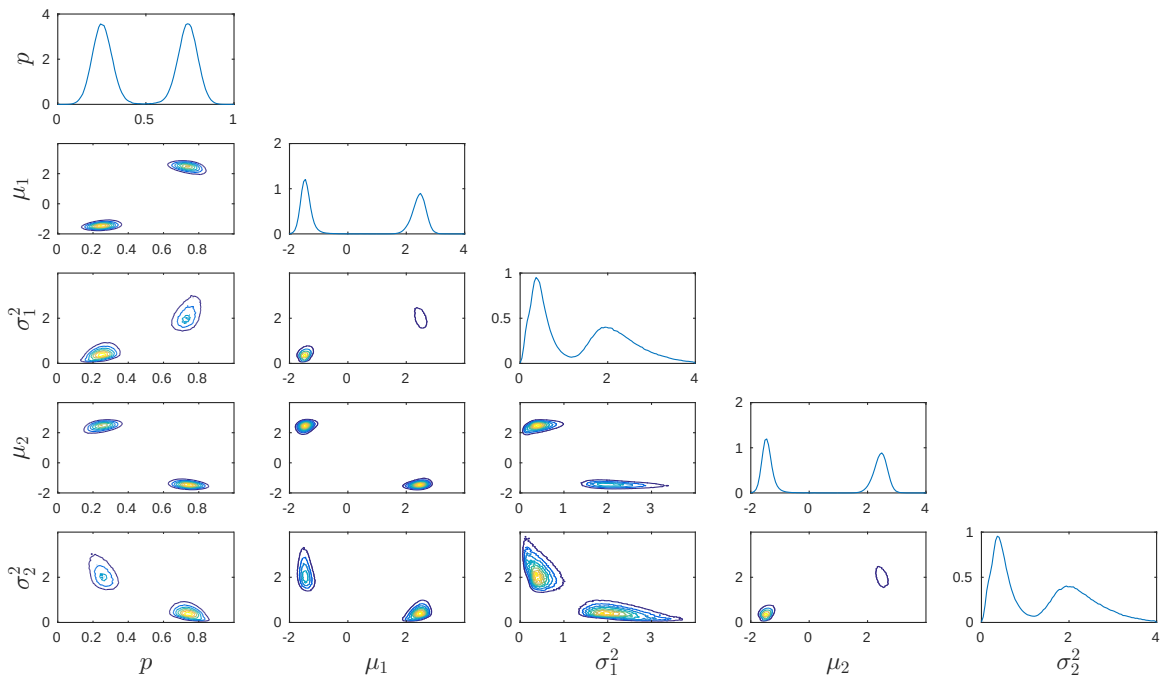


Figure 9. *The posterior distribution of θ as given in (8.1) as found from 10 million samples from the ETAIS algorithm. The main diagonal contains the marginal distributions of each θ_i, and the lower triangular contour represents the correlation between pairs of parameters.*

8.3.2. Implementation. In order to quantify error, we note that the probability density should be evenly divided between the two modes. This is due to the symmetric prior for p, to the same priors being assigned to μ_1 and μ_2, and to σ_1^2 and σ_2^2. To decide which mode a sample belongs to, we define a plane which bisects the posterior so that each point on this plane lies exactly halfway between the two true solutions to the inverse problem, i.e., the value of θ used to generate the data, and the θ obtained by a relabeling of the parameters. Now that we can assign a sample to a particular mode, we can calculate the density in each mode by summing the weights associated to all samples in that mode,

$$\bar{w}_k = \sum_{i=1}^{N} w_i I_{X_i \in \text{Mode } k}, \quad k = 1, 2,$$

and the relative error in the amount of density in each mode is then

$$w_{\text{error}} = 2 \left| \frac{\bar{w}_1}{\bar{w}_1 + \bar{w}_2} - \frac{1}{2} \right|. \tag{8.2}$$

Since the probability p is constrained to lie in the interval $[0, 1]$ and the variances must be positive, it can be wasteful to use Gaussian proposal distributions, which will produce samples outside of the support of the posterior. Moreover, the value of the variances of the unknown

Downloaded 11/28/19 to 90.194.4.112. Redistribution subject to CCBY license

© 2019 SIAM and ASA. Published by SIAM and ASA under the terms of the Creative Commons 4.0 license

Table 2

Optimal values of the scaling parameter. The MH algorithm is optimized using the acceptance rate, and the ETAIS algorithm is optimized using the effective sample size.

Algorithm	MH	ETAIS
δ^*	1.3e-1	2.3e-1

distributions are strictly nonnegative. The algorithm will be most efficient if the proposal and posterior distributions are mutually absolutely continuous. It is also useful to be able to pick proposal densities for which the variance is easily scaled so that we can tune them to optimize efficiency. Thus we pick the following proposal distributions for the p, μ, and σ^2 parameters, respectively:

$$q_p \sim \text{Beta}(\delta^{-2}p, \delta^{-2}(1-p)), \quad q_{\mu_{1,2}} \sim \mathcal{N}(\mu_{1,2}, 4\delta^2), \quad \text{and} \quad q_{\sigma^2_{1,2}} \sim \text{Gamma}(\alpha^*, \beta^*),$$

where $\alpha^* = \sigma^2_{1,2}\beta^*$, $\beta^* = \sigma^2_{1,2}/2\delta^2$, and δ is a scaling parameter to be tuned. This means that our proposal distributions will not be a mixture of multivariate Gaussians but rather independent mixtures of univariate Beta, Gamma, and Gaussian distributions.

In the numerics which follow, we have increased the ensemble size from $M = 50$ to $M = 500$ to compensate for the increase in dimension. We also use the MT algorithm for the resampling step because of the reduced computational cost. The method otherwise remains the same as in previous examples. We perform test runs to find the optimal scaling parameters, considering convergence to modes with equal density. We then calculate the convergence rates of the algorithms by producing 10 million samples from the posterior with each algorithm, and we repeat the simulation 32 times.

8.3.3. Convergence of MH versus ETAIS. Table 2 gives the optimal scaling parameters for the MH and ETAIS algorithms with the proposal distributions described in section 8.3.2.

Figure 10 displays convergence of the relative error for the two algorithms. ETAIS converges at the expected $\mathcal{O}(1/\sqrt{N})$ rate, whereas the MH algorithm converges to locally smooth histograms but with the wrong proportion of samples in each mode. The relatively low value of the error for the MH example is due to the priors covering the sample space evenly; however, since transitions are nearly impossible with a small value of the scaling parameter, this error will take a very long time to reduce. This problem was discussed in section 8.2.

In this example, we have only considered a relatively low-dimensional mixture model problem, with only two distributions in the mixture. With more elements in the mixture, and/or an undefined number in the mixture, the dimension of this problem will quickly increase. Importance sampling schemes such as this suffer from the curse of dimensionality, limiting the dimension of target density that it can efficiently sample from. However, this example does demonstrate the remarkable convergence of the ETAIS for multimodal target distributions.

8.4. Data assimilation with Lorenz '63 trajectories. The sampling algorithm we have introduced is designed for problems where the likelihood density function is expensive to calculate and the computational overhead involved in the calculation of the importance weights is dwarfed. This situation commonly occurs in inverse problems, where the model being investigated involves the solution of a differential equation. In this example we attempt to

Downloaded 11/28/19 to 90.194.4.112. Redistribution subject to CCBY license

© 2019 SIAM and ASA. Published by SIAM and ASA under the terms of the Creative Commons 4.0 license

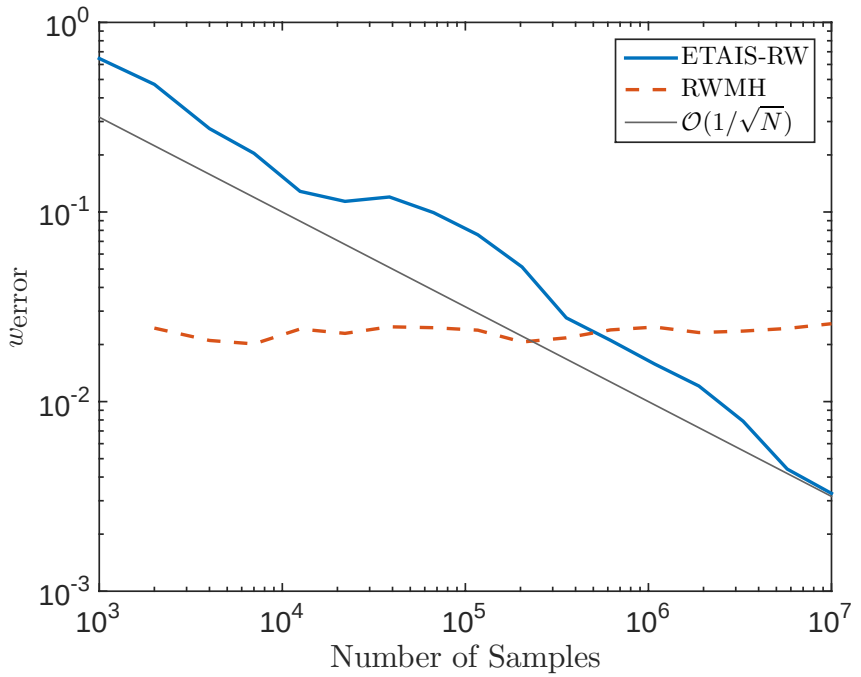


Figure 10. *Convergence of the ETAIS algorithm for the mixture model described in section 8.3, with convergence judged using the criterion in (8.2). Implementation described in sections 8.3.2 and 8.3.3. Resampling is performed using the MT scheme.*

recover the initial position of a particle with motion governed by Lorenz's 1963 atmospheric convection equations [28]:

$$\begin{aligned}\frac{\mathrm{d}x}{\mathrm{d}t} &= \sigma(y-x),\\ \frac{\mathrm{d}y}{\mathrm{d}t} &= x(\rho-z)-y,\\ \frac{\mathrm{d}z}{\mathrm{d}t} &= xy-\beta z.\end{aligned}$$

When the parameters are chosen to be $\rho = 28$, $\sigma = 10$, and $\beta = 8/3$, this system has chaotic solutions. This makes the estimation of initial conditions from noisy observations very challenging.

To demonstrate how small errors in the initial condition of a Lorenz solution can affect the trajectory of a particle, Figure 11 shows the path taken by two particles with very similar initial conditions,

$$\mathbf{x}_0^1 = \begin{pmatrix}-0.587\\-0.563\\16.870\end{pmatrix} \quad \text{and} \quad \mathbf{x}_0^2 = \begin{pmatrix}-0.590\\-0.563\\16.870\end{pmatrix}.$$

This small difference in the x-dimension of the initial condition leads to the trajectories of the two particles decoupling near $t = 5$. This chaotic behavior means that if we take only a few

© 2019 SIAM and ASA. Published by SIAM and ASA under the terms of the Creative Commons 4.0 license

Downloaded 11/28/19 to 90.194.4.112. Redistribution subject to CCBY license

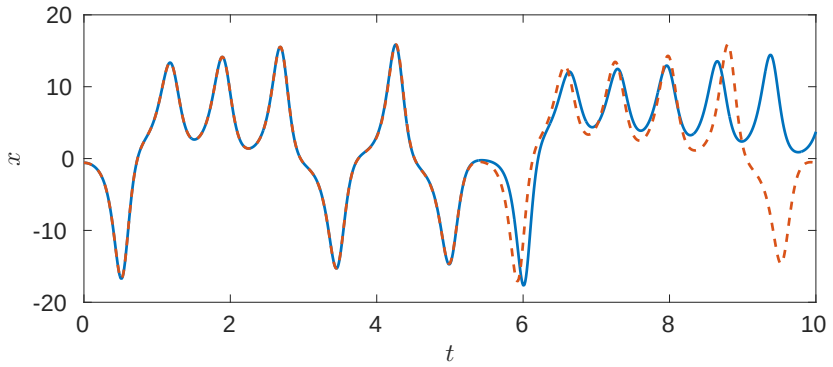


Figure 11. *Trajectory of the x component of a particle's position varying in time when motion is governed by the Lorenz '63 equations. The two trajectories are calculated using initial positions of* $[-0.587, -0.563, 16.870]$ *and* $[-0.590, -0.563, 16.870]$*, respectively.*

observations over a long time period, we will find that many trajectories which may be vastly different achieve similar values of the likelihood function, and so the posterior distribution for the initial condition can become very complex.

8.4.1. Target distribution. For this example, we observe noisily the position of a particle at 10 equally spaced points in the time interval $t \in (0, 1]$. The Lorenz equations are solved with the chaotic parameters given above and the initial condition $\mathbf{x}_0^1$. A time step of $h = 1 \times 10^{-3}$ was used to evolve the equations numerically with the explicit Euler method. The noise added to each observation was taken from the distribution $\mathcal{N}(0, 0.1^2 I)$. Priors for the initial condition coordinates were given by

$$x_0 \sim \mathcal{N}(-0.5, 0.4^2), \quad y_0 \sim \mathcal{N}(-0.5, 0.4^2), \quad \text{and} \quad z_0 \sim \mathcal{N}(15, 0.4^2).$$

The posterior density function takes the form

$$\pi(\mathbf{x}_0|D) \propto \exp\left\{-\frac{1}{2}\|\mathcal{G}(\mathbf{x}_0) - D\|_\Sigma^2\right\} \mathcal{N}(\mathbf{x}_0; m, 0.4^2\mathrm{I}),$$

where $\mathcal{G}$ calculates the solution to the Lorenz equations, using the Euler method with a time step of $h = 1 \times 10^{-3}$ starting at the initial condition $\mathbf{x}_0$, and returns the position at $t = 0.1, \ldots, 1$. The mean of the prior distribution $m = [-0.5, -0.5, 15]$.

Figure 12 shows the marginal distributions of the posterior distribution for the initial condition $\mathbf{x}_0$ as found using the MH algorithm. While the z-dimension is largely uncorrelated with the other two dimensions, the correlation between x and y is -0.97, which makes it a challenging posterior distribution to explore without a transformation of the parameter space.

8.4.2. Implementation. As in the previous example, we have no analytic form for the normalization constant of the posterior, and so we will measure convergence of the algorithms by calculating convergence to the posterior mean, where the truth is calculated using a very long chain produced by the MH algorithm.

Downloaded 11/28/19 to 90.194.4.112. Redistribution subject to CCBY license

© 2019 SIAM and ASA. Published by SIAM and ASA under the terms of the Creative Commons 4.0 license

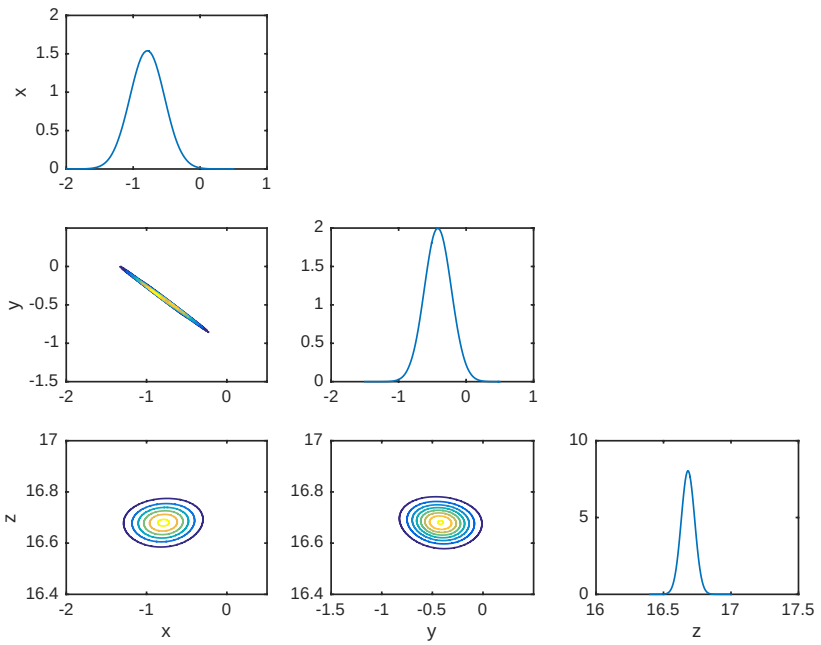


Figure 12. *Posterior distribution for the initial condition of the particle with motion governed by the Lorenz equations as described in section 8.4.1.*

For this example, we use an ensemble size of $M = 1500$ and the MT resampling algorithm. The optimal scaling parameters are calculated for both algorithms using test runs, which are not included in the convergence cost calculations. Convergence graphs are produced using 32 repeats of each algorithm with each simulation producing one million samples.

8.4.3. Convergence of MH versus ETAIS. Figure 13 shows convergence plots for this problem using RWMH and ETAIS. The burn-in of the ETAIS is remarkably better than MH, as the ensemble quickly covers the manifold where the majority of the density is supported. The plot shows convergence with respect to number of samples, but the two approaches have differing costs, due to the calculation of the denominator in the weights in the ETAIS and to the resampling step (MT). Assuming that both error curves decay like $N^{-1/2}$ from $N = 10^6$, for a given error tolerance, the number N_{MH} of MH samples required is given by

$$N_{\mathrm{MH}} = N_{\mathrm{ETAIS}} \times 3.838 \times 10^4,$$

where N_{ETAIS} is the number of ETAIS samples required for the same error tolerance. The cost per sample for the MH and ETAIS methods for this problem is 1.64×10^{-5} and 1.46×10^{-3}, respectively. Here the ETAIS results were produced with a serial implementation, so this cost may be slightly overoptimistic, since the runtimes do not include the extra overhead of communicating the ensemble states between processors. However, putting this together, we arrive at a speed-up factor of over 430, which will undoubtedly overshadow any underestimate of the cost-per-sample of the ETAIS approach, demonstrating the benefit of using this approach for problems with expensive likelihoods and challenging posteriors.

Downloaded 11/28/19 to 90.194.4.112. Redistribution subject to CCBY license

© 2019 SIAM and ASA. Published by SIAM and ASA under the terms of the Creative Commons 4.0 license

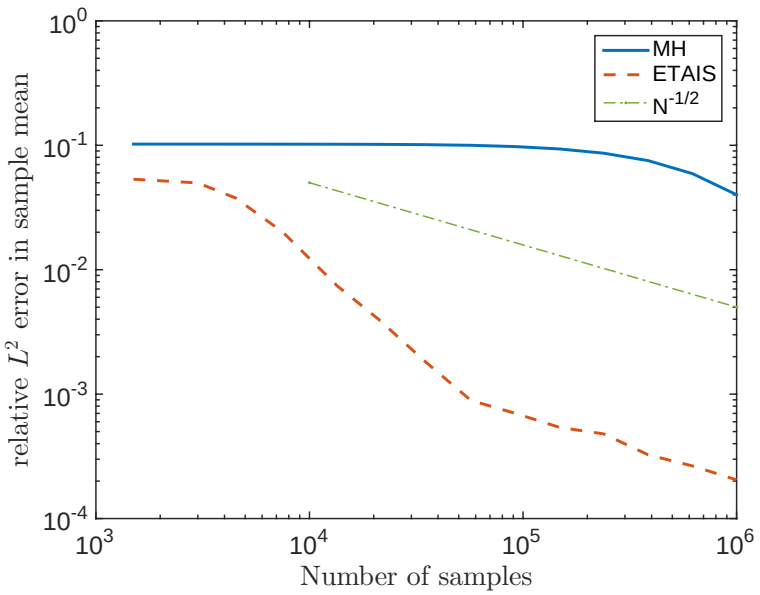


Figure 13. *Convergence of the sample mean to the posterior mean for the MH and ETAIS algorithms. Posterior distribution as described in section 8.4.1.*

9. Discussion and conclusions. We have explored the application of low-dimensional Bayesian inverse problems and demonstrated numerically that this method converges faster than the analogous naively parallelized Metropolis–Hastings (MH) algorithms. Further experimentation with the Metropolis adjusted Langevin algorithm (MALA) and the preconditioned Crank–Nicolson (pCN), preconditioned Crank–Nicolson Langevin (pCNL), and Hamiltonian Monte Carlo (HMC) proposals has yielded similar results [42].

The implementations of the ETAIS that have been presented in this paper have been run in serial, but we argue that, particularly for inverse problems with expensive likelihoods (for example, in the case where approximation of the solution of a differential equation is required), the ETAIS method is an excellent candidate for parallelization, with the likelihood evaluations for each particle computed on a different processor. That said, our numerical results demonstrate that even a serial implementation can outperform standard MH methods.

The ETAIS algorithm has a number of favorable features, for example, its ability to redistribute, through the resampling regime, the ensemble members to regions which require more exploration. This allows the method to be used to sample from complex multimodal distributions.

Another strength of the ETAIS method is that it can be used with any MCMC proposal. There are a growing number of increasingly sophisticated MCMC algorithms (nonreversible Langevin/HMC proposals, Riemann manifold MCMC, etc.) which could be incorporated into this framework, leading to even more efficient algorithms, and this introduces an opportunity for future work.

One limitation of the ETAIS approach as described above is that a direct solver of the ETPF problem (such as FastEMD [35]) has computational cost $\mathcal{O}(M^3 \log M)$, where M is the

Downloaded 11/28/19 to 90.194.4.112. Redistribution subject to CCBY license

© 2019 SIAM and ASA. Published by SIAM and ASA under the terms of the Creative Commons 4.0 license

number of particles in the ensemble. As such, we introduced a more approximate resampler, the approximate multinomial resampler, which allows us to push the approach to the limit with much larger ensemble sizes. The ETAIS framework is very flexible in terms of its ability to use any combination of proposal distributions and resampling algorithms.

We have demonstrated that the considered framework, with the use of state-of-the-art optimal transport-based resampling, can reduce the number of likelihood evaluations required to characterize complex posterior distributions in low dimensions to a given degree. We have also introduced a greedy approximation to this resampler that drastically reduces the cost but at the loss of some accuracy; however, this can be mitigated with the use of larger ensembles. We have detailed how scaling parameters in the MCMC proposals that are used within the mixture distribution can be quickly and efficiently tuned in an automated way. Lastly, we have demonstrated that when the likelihood is expensive, for instance, because it involves the numerical approximation of the solution of a differential equation, we can achieve orders of magnitude reductions in cost to reach a given error tolerance in comparison with standard MH approaches.

However, tuning the variances and covariances of the mixture proposal components globally is likely to be of limited use for multimodal problems where the modes have very different covariances or where there are curved ridges in the density. With increased dimensionality and/or complexity of the target distribution, we also require increases in the size of the ensemble if we wish to have a stable ETAIS implementation. After a certain point, the extra overhead associated with a very large ensemble will outweigh the advantages of this approach. However, a variety of possible solutions to this problem are worthy of future consideration, including the potential to use transport maps [17, 34] to map Gaussian mixtures to highly complex non-Gaussian approximations of the posterior. Such maps could encode local covariance information and lead to accelerated and stable ETAIS sampling with much smaller ensemble sizes.

Appendix A. Glossary of acronyms.

Acronym	Full name
MCMC	Markov chain Monte Carlo
RWMH	Random walk Metropolis–Hastings
MALA	Metropolis-adjusted Langevin algorithm
AIS	Adaptive importance sampling
PMC	Population Monte Carlo
ETAIS	Ensemble transport adaptive importance sampling
ETPF	Ensemble transport particle filter
MT	Multinomial transformation
ETAIS-X	ETAIS with X kernels
AETAIS-X	Adaptive ETAIS with X kernels

REFERENCES

[1] C. Andrieu and É. Moulines, *On the ergodicity properties of some adaptive MCMC algorithms*, Ann. Appl. Probab., 16 (2006), pp. 1462–1505.

[2] E. Angelino, E. Kohler, A. Waterland, M. Seltzer, and R. Adams, *Accelerating MCMC via Parallel Predictive Prefetching*, preprint, https://arxiv.org/abs/1403.7265, 2014.

[3] J. Bierkens, *Non-reversible Metropolis-Hastings*, Stat. Comput., 26 (2016), pp. 1213–1228.

[4] M. Bink, M. Boer, C. Ter Braak, J. Jansen, R. Voorrips, and W. Van de Weg, *Bayesian analysis of complex traits in pedigreed plant populations*, Euphytica, 161 (2008), pp. 85–96.

Downloaded 11/28/19 to 90.194.4.112. Redistribution subject to CCBY license

© 2019 SIAM and ASA. Published by SIAM and ASA under the terms of the Creative Commons 4.0 license

[5] M. Bugallo, L. Martino, and J. Corander, *Adaptive importance sampling in signal processing*, Digital Signal Process., 47 (2015), pp. 36–49.
[6] T. Bui-Thanh and M. Girolami, *Solving large-scale PDE-constrained Bayesian inverse problems with Riemann manifold Hamiltonian Monte Carlo*, Inverse Problems, 30 (2014), 114014.
[7] B. Calderhead, *A general construction for parallelizing Metropolis-Hastings algorithms*, Proc. Natl. Acad. Sci. USA, 111 (2014), pp. 17408–17413.
[8] O. Cappé, R. Douc, A. Guillin, J.-M. Marin, and C. Robert, *Adaptive importance sampling in general mixture classes*, Stat. Comput., 18 (2008), pp. 447–459.
[9] O. Cappé, A. Guillin, J. Marin, and C. Robert, *Population Monte Carlo for ion channel restoration.*
[10] O. Cappé, A. Guillin, J. Marin, and C. Robert, *Population Monte Carlo*, J. Comput. Graph. Statist., 13 (2012), pp. 907–929.
[11] G. Celeux, J. Marin, and C. Robert, *Iterated importance sampling in missing data problems*, Computat. Statist. Data Anal., 50 (2006), pp. 3386–3404.
[12] J. Cornuet, J. Marin, A. Mira, and C. Robert, *Adaptive multiple importance sampling*, Scand. J. Statist., 39 (2012), pp. 798–812.
[13] C. Cotter and S. Reich, *Ensemble filter techniques for intermittent data assimilation—a survey*, in Large Scale Inverse Problems. Computational Methods and Applications in the Earth Sciences, Walter de Gruyter, Berlin, 2012, pp. 91–134.
[14] S. Cotter, G. Roberts, A. Stuart, and D. White, *MCMC methods for functions: Modifying old algorithms to make them faster*, Statist. Sci., 28 (2013), pp. 424–446.
[15] R. Douc, A. Guillin, J.-M. Marin, and C. Robert, *Convergence of adaptive mixtures of importance sampling schemes*, Ann. Statist., 35 (2007), pp. 420–448.
[16] R. Douc, A. Guillin, J.-M. Marin, and C. Robert, *Minimum variance importance sampling via population Monte Carlo*, ESAIM Probab. Statist., 11 (2007), pp. 427–447.
[17] T. El Moselhy and Y. Marzouk, *Bayesian inference with optimal maps*, J. Comput. Phys., 231 (2012), pp. 7815–7850.
[18] V. Elvira, L. Martino, D. Luengo, and M. Bugallo, *Improving population Monte Carlo: Alternative weighting and resampling schemes*, Signal Process., 131 (2017), pp. 77–91.
[19] G. Evensen, *Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics*, J. Geophys. Res. Oceans, 99 (1994), pp. 10143–10162.
[20] M. Girolami and B. Calderhead, *Riemann manifold Langevin and Hamiltonian Monte Carlo methods*, J. R. Stat. Soc. Ser. B Stat. Methodol., 73 (2011), pp. 123–214.
[21] H. Haario, E. Saksman, and J. Tamminen, *Componentwise adaptation for high dimensional MCMC*, Comput. Statist., 20 (2005), pp. 265–273.
[22] W. Hastings, *Monte Carlo sampling methods using Markov chains and their applications*, Biometrika, 57 (1970), pp. 97–109.
[23] T. House, A. Ford, S. Lan, S. Bilson, E. Buckingham-Jeffery, and M. Girolami, *Bayesian uncertainty quantification for transmissibility of influenza, norovirus and Ebola using information geometry*, J. R. Soc. Interface, 13 (2016), 20160279.
[24] M. Isard and A. Blake, *Condensation—conditional density propagation for visual tracking*, Int. J Comput. Vis., 29 (1998), pp. 5–28.
[25] C. Ji and S. C. Schmidler, *Adaptive Markov chain Monte Carlo for Bayesian variable selection*, J. Comput. Graph. Statist., 22 (2013), pp. 708–728.
[26] J. Liu, *Monte Carlo Strategies in Scientific Computing*, Springer Ser. Statist., Springer, New York, 2008.
[27] J. Liu, F. Liang, and W. Wong, *The multiple-try method and local optimization in Metropolis sampling*, J. Amer. Statist. Assoc., 95 (2000), pp. 121–134.
[28] E. Lorenz, *Deterministic nonperiodic flow*, J. Atmospheric Sci., 20 (1963), pp. 130–141.
[29] J.-M. Marin, K. Mengersen, and C. Robert, *Bayesian modelling and inference on mixtures of distributions*, in Bayesian Thinking: Modeling and Computation, Handbook of Statist. 25, Elsevier/North-Holland, Amsterdam, 2005, pp. 459–507.
[30] L. Martino, V. Elvira, D. Luengo, and J. Corander, *An adaptive population importance sampler: Learning from uncertainty*, IEEE Trans. Signal Process., 63 (2015), pp. 4422–4437.
[31] L. Martino, V. Elvira, D. Luengo, and J. Corander, *Layered adaptive importance sampling*, Statist. Comput., 27 (2017), pp. 599–623.

Downloaded 11/28/19 to 90.194.4.112. Redistribution subject to CCBY license

© 2019 SIAM and ASA. Published by SIAM and ASA under the terms of the Creative Commons 4.0 license

[32] G. Moore, *Cramming more components onto integrated circuits*, IEEE Solid-State Circuits Soc. Newslett., 11 (2006), pp. 33–35.
[33] R. Neal, *MCMC Using Ensembles of States for Problems with Fast and Slow Variables such as Gaussian Process Regression*, preprint, https://arxiv.org/abs/1101.0387, 2011.
[34] M. Parno and Y. Marzouk, *Transport Map Accelerated Markov Chain Monte Carlo*, preprint, https://arxiv.org/abs/1412.5492, 2014.
[35] O. Pele and M. Werman, *Fast and robust Earth mover's distances*, in Proceedings of the 2009 IEEE 12th International Conference on Computer Vision, IEEE, Piscataway, NJ, 2009, pp. 460–467.
[36] S. Reich, *A nonparametric ensemble transform method for Bayesian inference*, SIAM J. Sci. Comput., 35 (2013), pp. A2013–A2024, https://doi.org/10.1137/130907367.
[37] C. Robert and G. Casella, *Monte Carlo Statistical Methods*, Springer-Verlag, New York, 2004.
[38] G. Roberts and J. Rosenthal, *Optimal scaling of discrete approximations to Langevin diffusions*, J. R. Stat. Soc. Ser. B Stat. Methodol., 60 (1998), pp. 255–268.
[39] G. Roberts and J. Rosenthal, *Optimal scaling for various Metropolis-Hastings algorithms*, Statist. Sci., 16 (2001), pp. 351–367.
[40] G. Roberts and J. Rosenthal, *Coupling and ergodicity of adaptive Markov chain Monte Carlo algorithms*, J. Appl. Probab., 44 (2007), pp. 458–475.
[41] G. Roberts and J. Rosenthal, *Examples of adaptive MCMC*, J. Comput. Graph. Statist., 18 (2009), pp. 349–367.
[42] P. Russell, *Parallel MCMC Methods and Their Applications in Inverse Problems*, Ph.D. thesis, School of Mathematics, University of Manchester, 2017.
[43] J. Sexton and D. Weingarten, *Hamiltonian evolution for the hybrid Monte Carlo algorithm*, Nuclear Phys. B, 380 (1992), pp. 665–677.
[44] J. Sirén, P. Marttinen, and J. Corander, *Reconstructing population histories from single nucleotide polymorphism data*, Mol. Biol. Evol., 28 (2011), pp. 673–683.
[45] A. Solonen, P. Ollinaho, M. Laine, H. Haario, J. Tamminen, and H. Järvinen, *Efficient MCMC for climate model parameter estimation: Parallel adaptive chains and early rejection*, Bayesian Anal., 7 (2012), pp. 715–736.

Downloaded 11/28/19 to 90.194.4.112. Redistribution subject to CCBY license

© 2019 SIAM and ASA. Published by SIAM and ASA under the terms of the Creative Commons 4.0 license