Received <day> <Month>, <year>; | Revised <day> <Month>, <year>; | Accepted <day> <Month>, <year>

DOI: xxx/xxxx

ARTICLE TYPE

SwISS: A Scalable Markov chain Monte Carlo Divide-and-Conquer Strategy

Callum Vyner | Christopher Nemeth* | Chris Sherlock

[1]Department of Mathematics and Statistics, Lancaster University, Lancashire, U.K.

Correspondence
*Christopher Nemeth, Postgraduate Statistics Centre, Lancaster University, Lancaster, LA1 4YF, UK. Email: c.nemeth@lancaster.ac.uk

Present Address
Present address

Abstract

Divide-and-conquer strategies for Monte Carlo algorithms are an increasingly popular approach to making Bayesian inference scalable to large data sets. In its simplest form, the data are partitioned across multiple computing cores and a separate Markov chain Monte Carlo algorithm on each core targets the associated partial posterior distribution, which we refer to as a sub-posterior, that is the posterior given only the data from the segment of the partition associated with that core. Divide-and-conquer techniques reduce computational, memory and disk bottle necks, but make it difficult to recombine the sub-posterior samples. We propose SwISS: Sub-posteriors with Inflation, Scaling and Shifting; a new approach for recombining the sub-posterior samples which is simple to apply, scales to high-dimensional parameter spaces and accurately approximates the original posterior distribution through affine transformations of the sub-posterior samples. We prove that our transformation is asymptotically optimal across a natural set of affine transformations and illustrate the efficacy of SwISS against competing algorithms on synthetic and real-world data sets.

KEYWORDS:
Markov chain Monte Carlo; divide-and-conquer; parallel MCMC; big data

1 | INTRODUCTION

Markov chain Monte Carlo (MCMC) algorithms are widely used within Bayesian modelling to sample from the often intractable posterior distribution. These techniques are widely applicable and only require point-wise evaluation of the posterior density. One of the potential drawbacks of MCMC algorithms is their lack of scalability. The computational cost of MCMC is typically linear in the amount of data and can be prohibitive for large data sets, both in computational cost and storage.

In settings with large data sets, or where the model is computationally expensive, evaluating the posterior at every iteration of the MCMC algorithm may be infeasible. Strategies to overcome this include data subsampling (1; 2; 10; 18), where only a subset of the data is used at each MCMC iteration, or delayed acceptance (12; 15), where the Metropolis–Hastings accept-reject probability is replaced with a cheaper approximation to the true posterior and the full data posterior is evaluated less frequently.

In situations where it is possible to easily parallelise computation in a MapReduce framework, or through cloud-computing infrastructure such as Amazon Web Services, then statistical modelling becomes easily scalable to large data sets. However, applying this approach in practice using algorithms such as MCMC, which are designed to work in serial rather than parallel, is challenging. In this paper, we consider the divide-and-conquer strategy to circumvent the computational bottleneck of MCMC, where the data are partitioned into batches, and each batch is stored on a separate computer core. MCMC is then applied independently on each data batch and posterior samples from each computer are combined to form an accurate approximation of the full posterior, *i.e.*, the posterior that would have been obtained using the full data set.

[0] **Abbreviations:** MCMC, Markov chain Monte Carlo; SwISS, sub-posteriors with inflation, scaling and shifting

The main challenge with divide-and-conquer approaches for MCMC lies in the merge step. A range of approaches has been considered in the literature such as: the use of weighted averages of the batch samples (14); kernel density estimation (11); Gaussian process approximations (9); finding the Wasserstein barycenter of different measures (16), the geometric median of batch samples (8), as well as using a post-MCMC importance sample (5), to name a few.

One of the most popular algorithms in the literature is the consensus Monte Carlo algorithm (14), which approximates the full posterior using a weighted average of sub-posterior samples. The consensus approach is computationally cheap to apply, does not require tuning and scales well to high-dimensional parameter spaces. It is also analytically exact in the case of Gaussian sub-posteriors, but can produce poor approximations when the sub-posteriors are non-Gaussian (see Section 4.4).

In this paper we propose SwISS, an algorithm that is as fast as the consensus algorithm, is exact in the Gaussian case, does not require tuning, and which scales well to high-dimensional posterior distributions. However, in the case of non-Gaussian sub-posteriors, it can produce more accurate posterior approximations than the consensus algorithm. Unlike the consensus approach, SwISS does not merge samples but instead applies a transformation to the posterior samples that are generated from a stochastic approximation of the full posterior. As in (5), we refer to this stochastic approximation as the *inflated sub-posterior*, which is the posterior density, conditional on a subset of the data, raised to a positive power. Inflating the sub-posterior in this manner has the effect of approximately preserving the shape of the posterior density conditional on the full data. Affine transformations (shift and re-scale) are applied to each batch of sub-posterior samples to form an approximate sample from the full posterior. This is a generalisation of the algorithm of (19) which simply shifts each sub-posterior, with no further correction and hence performs poorly when the sub-posterior variances differ substantially. There are many different affine transformations that produce a sample from the true posterior when the sub-posteriors are Gaussian; we provide theoretical support for our particular choice, showing that, in a natural sense, it is optimal amongst the set of transformations that are exact in the Gaussian case.

The paper is organised as follows: Section 2 provides an introduction to divide-and-conquer MCMC, covering the notation for posterior and sub-posterior densities. In Section 3 we introduce our proposed algorithm, SwISS, and provide supporting theoretical results and pseudo-code for implementation. Section 4 covers the numerical performance of SwISS and is compared against other popular divide-and-conquer algorithms from the literature. Finally, Section 5 gives a summary of the contributions from the paper.

2 | PRELIMINARIES

Let $f(\mathbf{y}|\boldsymbol{\theta})$ be the likelihood for a statistical model, parameterised by $\boldsymbol{\theta} \in \mathbb{R}^d$, for a data set $\mathbf{y} = \{y_1, y_2, \dots, y_n\}$ of length n. Let $\pi_0(\boldsymbol{\theta})$ denote the prior density for the parameter vector $\boldsymbol{\theta}$, then our posterior density is, up to a constant of proportionality,

$$\pi(\boldsymbol{\theta}|\mathbf{y}) \propto \pi_0(\boldsymbol{\theta}) f(\mathbf{y}|\boldsymbol{\theta}). \tag{1}$$

We assume that $\mathbf{y}$ can be partitioned into B batches, $\mathbf{y}_1, \dots, \mathbf{y}_B$, such that the likelihood for the full data is the product of the likelihoods for the individual batches, *i.e.*, $f(\mathbf{y}|\boldsymbol{\theta}) = \prod_{b=1}^{B} f_b(\mathbf{y}_b|\boldsymbol{\theta})$. This is the case, for example, when the individual data points are independent. The posterior density for $\boldsymbol{\theta}$ given $\mathbf{y}$ is, up to a constant of proportionality,

$$\pi(\boldsymbol{\theta}|\mathbf{y}) \propto \pi_0(\boldsymbol{\theta}) \prod_{b=1}^{B} f_b(\mathbf{y}_b|\boldsymbol{\theta}). \tag{2}$$

In the literature, there are generally two approaches to applying MCMC on batches of data. In the first approach, MCMC is applied to target a *sub-posterior* density for each batch b, of the form

$$\pi_b(\boldsymbol{\theta}|\mathbf{y}_b) \propto \pi_0(\boldsymbol{\theta})^{\frac{1}{B}} f(\mathbf{y}_b|\boldsymbol{\theta}), \tag{3}$$

where $b = 1, \dots, B$, such that $\prod_{b=1}^{B} \pi_b(\boldsymbol{\theta}|\mathbf{y}_b) = \pi(\boldsymbol{\theta}|\mathbf{y})$ as defined in (2).

If we assume that there are J sub-posterior samples from each of the B batches, which we define as $(\boldsymbol{\theta}_b^{(j)}; j \in \{1, \dots, J\}, b \in \{1, \dots, B\})$, then the consensus Monte Carlo algorithm (14) gives a simple strategy for approximating the full posterior (2) through a weighted average of the sub-posterior samples,

$$\boldsymbol{\theta}^{(j)} = \left(\sum_{b=1}^{B} \mathbf{w}_b\right)^{-1} \sum_{b=1}^{B} \mathbf{w}_b \boldsymbol{\theta}_b^{(j)},$$

where the weights are typically chosen to be $\mathbf{w}_b = \mathrm{Var}\left[\boldsymbol{\theta}|\mathbf{y}_b\right]^{-1}$. If each $\pi_b(\boldsymbol{\theta}|\mathbf{y}_b)$ is Gaussian, then the consensus algorithm produces exact samples from the full posterior.

A second approach applies MCMC to each *inflated sub-posterior*, where the target density for batch $b = 1, \dots, B$ is

$$\pi_b^B(\boldsymbol{\theta}|\mathbf{y}_b) \propto \pi_0(\boldsymbol{\theta}) f(\mathbf{y}_b|\boldsymbol{\theta})^B. \tag{4}$$

This is a stochastic (across partitions of the data) approximation to the full posterior $\pi(\boldsymbol{\theta}|\mathbf{y}) \approx \pi_b^B(\boldsymbol{\theta}|\mathbf{y}_b)$, and hence individual samples from it, in a sense, are already on the same scale as samples from the full posterior.

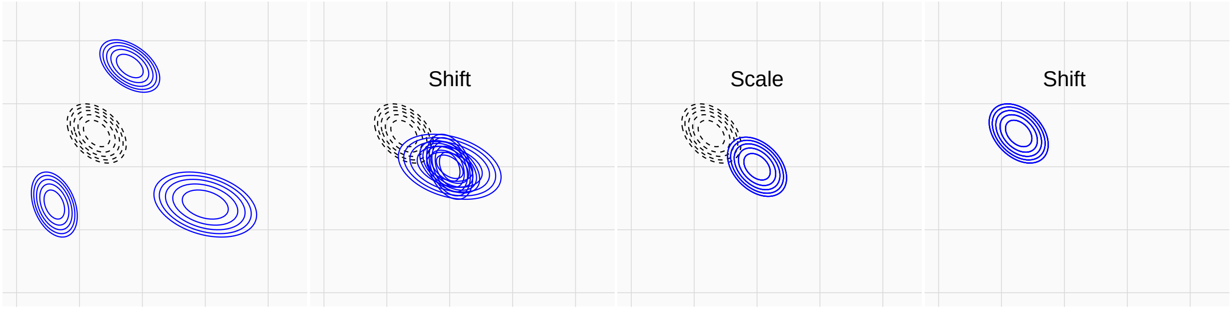

FIGURE 1 A visual representation of the SwISS algorithm applied to a two-dimensional Gaussian with three sub-posteriors (blue) approximating the full posterior (black). First, the batch samples are shifted by their respective means ($\boldsymbol{\theta}_\mathsf{b} - \boldsymbol{\mu}_\mathsf{b}$), then scaled by the matrix $\mathbf{A}_\mathsf{b}$ before finally being shifted by the global mean $\boldsymbol{\mu}$.

If we assume that the data are partitioned equally across batches, then in the limit, as the amount of data in each batch $\mathsf{n}_* = \mathsf{n}/\mathsf{B}$ approaches infinity, the likelihood will typically dominate the prior, so that by the Bernstein von Mises theorem, the b^{th} inflated sub-posterior is $\boldsymbol{\theta}_\mathsf{b} \sim \mathcal{N}(\hat{\boldsymbol{\theta}}_\mathsf{b}, \mathsf{I}_{\mathsf{O},\mathsf{b}}^{-1}(\hat{\boldsymbol{\theta}}_\mathsf{b}))$, where approximately, $\hat{\boldsymbol{\theta}}_\mathsf{b} \sim \mathcal{N}(\boldsymbol{\theta}_0, \mathsf{B}\mathsf{I}_\mathsf{E}^{-1}(\boldsymbol{\theta}_0))$ and where I_E and $\mathsf{I}_{\mathsf{O},\mathsf{b}}$ are the full-data expected information and the observed information from the inflated likelihood for batch b, respectively, and $\boldsymbol{\theta}_0$ is the true parameter value. Hence, the difference between the expectations of the inflated sub-posteriors are $\mathcal{O}(\mathsf{n}_*^{-1/2})$ and, since $\lim_{\mathsf{n}\to\infty} ||\mathsf{I}_\mathsf{E}^{-1}\mathsf{I}_{\mathsf{O},\mathsf{b}}|| = 1 + \mathcal{O}(\mathsf{n}_*^{-1/2})$, the ratio of the variances of the sub-posteriors is $1 + \mathcal{O}(\mathsf{n}_*^{-1/2})$. However, in practice, both the location and scale of the inflated sub-posteriors can vary considerably if the partitioned data sets are imbalanced (see examples in Section 4). Our proposed algorithm, SwISS, provides a correction for the discrepancy in the variance and location of the sub-posterior approximations.

3 | SWISS ALGORITHM

Suppose that we have applied independent Monte Carlo algorithms, such as MCMC, in parallel to sample from the inflated sub-posteriors (4), and denote the j^{th} (of J) sample from the b^{th} (of B) batch by $\theta_\mathsf{b}^{(\mathsf{j})}$. The SwISS algorithm transforms each sample from the inflated sub-posterior into a sample from an approximation to the full posterior (2) using a batch-specific affine transformation. In the case of Gaussian sub-posteriors, as we show below, these affine transformations produce a set of samples from the correct Gaussian full posterior, and SwISS is exact in this setting. In general, sub-posteriors are non-Gaussian; however under standard regularity conditions and the Bernstein-von Mises theorem (7), as $\mathsf{n}^* = \mathsf{n}/\mathsf{B}$ approaches infinity the sub-posteriors will be approximately Gaussian and SwISS can be expected to produce samples from an approximation to the full posterior.

Firstly, let us suppose that each inflated sub-posterior is Gaussian with expectation $\boldsymbol{\mu}_\mathsf{b}$ and invertible variance matrix $\mathbf{V}_\mathsf{b}$, so that the full posterior is $\boldsymbol{\theta}|\mathbf{y} \sim \mathcal{N}_\mathsf{d}(\boldsymbol{\mu}, \mathbf{V})$ where,

$$\mathbf{V} = \left(\frac{1}{\mathsf{B}}\sum_{\mathsf{b}=1}^{\mathsf{B}} \mathbf{V}_\mathsf{b}^{-1}\right)^{-1} \text{ and } \boldsymbol{\mu} = \mathbf{V}\frac{1}{\mathsf{B}}\sum_{\mathsf{b}=1}^{\mathsf{B}} \mathbf{V}_\mathsf{b}^{-1}\boldsymbol{\mu}_\mathsf{b} \tag{5}$$

are the variance and mean of the full posterior.

Since it is invertible, $\mathbf{V}_\mathsf{b}$ is a positive-definite matrix and therefore it has a $\mathsf{d} \times \mathsf{d}$, invertible square root, $\mathbf{M}_\mathsf{b}$; *i.e.* $\mathbf{V}_\mathsf{b} = \mathbf{M}_\mathsf{b}\mathbf{M}_\mathsf{b}^\top$. Similarly, the full posterior variance $\mathbf{V}$ has a square root, $\mathbf{M}$, so that $\mathbf{V} = \mathbf{M}\mathbf{M}^\top$. Let samples from the b^{th} inflated sub-posterior, $\boldsymbol{\theta}_\mathsf{b}^{(1:\mathsf{J})}$, be (marginally) realisations from the random variable $\boldsymbol{\theta}_\mathsf{b} \sim \pi_\mathsf{b}^\mathsf{B}$ and define the transformed random variable:

$$\boldsymbol{\vartheta}_b := \mathbf{A}_b\left(\boldsymbol{\theta}_b - \boldsymbol{\mu}_b\right) + \boldsymbol{\mu}. \tag{6}$$

where $\mathbf{A}_\mathsf{b}$ is any matrix satisfying $\mathbf{A}_\mathsf{b}\mathbf{V}_\mathsf{b}\mathbf{A}_\mathsf{b}^\top = \mathbf{V}$; for example, $\mathbf{A}_\mathsf{b} = \mathbf{M}\mathbf{M}_\mathsf{b}^{-1}$.

Clearly, $\mathbb{E}\left[\boldsymbol{\vartheta}_\mathsf{b}\right] = \boldsymbol{\mu}$ and $\mathsf{Var}\left[\boldsymbol{\vartheta}_\mathsf{b}\right] = \mathbf{V}$. Furthermore, an affine transformation of a Gaussian random variable is Gaussian, and hence $\boldsymbol{\vartheta} \sim \mathcal{N}_\mathsf{d}(\boldsymbol{\mu}, \mathbf{V})$. Applying the same transformation to individual samples from the b^{th} batch, therefore provides a sample from the full posterior. As discussed above, even when the sub-posteriors are not Gaussian, we can still apply the same scaling and shifting to any sub-posterior samples and produce samples from an approximation to the full posterior.

3.1 | Choice of Matrix Square Roots

Matrix square roots are not unique; *e.g.* for a diagonal matrix, each element of the diagonal square root could be negated; methods for finding a square root of a positive-definite matrix include the Cholesky decomposition, or the simple asymmetric square root arising from the spectral decomposition. Moreover, for square roots $\mathbf{M}$ and $\mathbf{M}_b$, $\mathbf{A}_b$ need not be simply $\mathbf{M}\mathbf{M}_b^{-1}$, and indeed, this is not always the most sensible choice.

For now, let $\mathbf{M}$ be any $d \times d$ square root of $\mathbf{V}$ and let

$$\widetilde{\mathbf{V}}_b := \mathbf{M}^{-1}\mathbf{V}_b\left(\mathbf{M}^{-1}\right)^\top \quad \text{and} \quad \mathbf{A}_b = \mathbf{M}\widetilde{\mathbf{M}}_b^{-1}\mathbf{M}^{-1},$$

where $\widetilde{\mathbf{M}}_b$ is any $d \times d$ square root of $\widetilde{\mathbf{V}}_b$. Then, $\text{Var}\left[\mathbf{M}^{-1}\boldsymbol{\theta}_b\right] = \widetilde{\mathbf{V}}_b$, so $\text{Var}\left[\widetilde{\mathbf{M}}_b^{-1}\mathbf{M}^{-1}\boldsymbol{\theta}_b\right] = \mathbf{I}$ and hence $\text{Var}\left[\mathbf{A}_b\boldsymbol{\theta}_b\right] = \mathbf{M}\mathbf{M}^\top = \mathbf{V}$.

If $\mathbf{V}_b = \mathbf{V}$ for all $b = 1, \ldots, B$, then $\widetilde{\mathbf{V}}_b = \mathbf{I}$, and provided $\widetilde{\mathbf{M}}_b$ is chosen to be $\mathbf{I}$, $\mathbf{A}_b$ becomes the identity transformation. To be clear, though, if some diagonal elements of $\widetilde{\mathbf{M}}_b$ had been chosen to be -1 rather than 1 then $\mathbf{A}_b$ would not be the identity and, unless the initial distribution of points $\boldsymbol{\theta}_b^{(1:J)}$ was elliptically symmetric, the transformation in (6) would not then lead to a set of points that represented the true posterior at all.

Applying the same logic as above, the transformation $\widetilde{\mathbf{M}}_b$ should be the square root of $\widetilde{\mathbf{V}}_b$ that moves the individual points $\boldsymbol{\theta}_b^{(j)}$ as little as possible. With this in mind, we define a natural measure of the distance moved by J points, $\boldsymbol{\theta}^{(1:J)}$, to which a linear transformation $\mathbf{A}$ is applied, as:

$$D\left(\mathbf{A}; \boldsymbol{\theta}^{(1:J)}\right) := \frac{1}{J}\sum_{j=1}^{J}\left\|\boldsymbol{\theta}^{(j)} - \mathbf{A}\boldsymbol{\theta}^{(j)}\right\|^2, \tag{7}$$

where $||.||^2$ denotes Euclidean distance. We wish to find the linear transformation $\widetilde{\mathbf{M}}_b$ that minimises $D(\widetilde{\mathbf{M}}_b^{-1}; \boldsymbol{\theta}^{(1:J)})$ subject to the constraint that $\widetilde{\mathbf{M}}_b\widetilde{\mathbf{M}}_b^\top = \widetilde{\mathbf{V}}_b$. In Section 3.2 we show that, provided the points have expectation zero, as $J \uparrow \infty$, the best choice of $\widetilde{\mathbf{M}}_b$ is the positive-definite, symmetric square root of $\widetilde{\mathbf{V}}_b$; this is the square root used by SwISS. The choice of square root, $\mathbf{M}$, of $\mathbf{V}$ is less important, since within the linear transformation $\mathbf{A}_b$, the initial transformation by $\mathbf{M}^{-1}$ is later inverted; however, with the general motivation of preventing excess movement, SwISS sets $\mathbf{M}$ to be the positive-definite, symmetric square root of $\mathbf{V}$. Finally, the averaged re-centring algorithm of (19) can be viewed as a special case of SwISS where $\mathbf{A}_b = \mathbf{I}$.

3.2 | The Positive-Definite, Symmetric Square Root and its Optimality

We first define the positive-definite symmetric square root of a positive-definite matrix and detail the sense in which it is optimal with respect to the distance measure D (7).

Let $\mathbf{V}$ be a positive-definite matrix and let its spectral decomposition be

$$\mathbf{V} = \mathbf{U}\boldsymbol{\Lambda}\boldsymbol{\Lambda}\mathbf{U}^\top. \tag{8}$$

where Λ is a diagonal matrix with entries equal to the positive square roots of the eigenvalues of $\mathbf{V}$, and $\mathbf{U}$ is a unitary matrix (*i.e.* the columns of $\mathbf{U}$ are the orthonormal right eigenvectors of $\mathbf{V}$, so $\mathbf{U}\mathbf{U}^\top = \mathbf{I} = \mathbf{U}^\top\mathbf{U}$) and so

$$\mathbf{V}^{1/2} := \mathbf{M} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top. \tag{9}$$

The natural interpretation of $\mathbf{M}$ is as a simple scaling transformation with different scalings applied along each of the eigenvectors of $\mathbf{V}$.

As explained previously, we require a matrix $\mathbf{A}_b$ such that the transformation (6) leads to a sample with a variance of $\mathbf{V}$; however, when inflated sub-posteriors are non-Gaussian, we need a transformation that preserves the shape and orientation of the inflated sub-posterior as much as possible. Theorem 1 shows that for large J, $\mathbf{M}^{-1}$ is not likely to cause more than the minimum discrepancy, given the constraints.

Theorem 1. Let $\boldsymbol{\theta}^{(j)} \in \mathbb{R}^d$ $(j = 1, \ldots, J)$ be a set of independent and identically distributed realisations of a random variable $\boldsymbol{\theta}$ with $\mathbb{E}[\boldsymbol{\theta}] = 0$ and $\text{Var}[\boldsymbol{\theta}] = \mathbf{V}$. Let $\mathbf{V}$ have a spectral decomposition as in (8) and let $\mathbf{M} = \mathbf{U}\Lambda\mathbf{U}^\top$. Let $\mathbf{A}$ be any other $d \times d$ matrix such that $\text{Var}[\mathbf{A}\boldsymbol{\theta}] = \mathbf{I}_d$. Then

$$\mathbb{P}\left(\lim_{J\to\infty}\left[D\left(\mathbf{M}^{-1}; \boldsymbol{\theta}^{(1:J)}\right) - D\left(\mathbf{A}; \boldsymbol{\theta}^{(1:J)}\right)\right] > 0\right) = 0,$$

where $D(\cdot;\cdot)$ is as defined in (7).

Theorem 1 relies upon the following two results.

Proposition 1. Let $Z_i \in \mathbb{R}^d$ $(i = 1, \ldots, n)$ be an independent and identically distributed sequence of random variables with $\mathbb{E}[Z] = 0$ and $\text{Var}[Z] = I_d$, and let B be any $d \times d$ matrix. Then, as $n \to \infty$, $D(B; Z_{1:n}) \to d + \text{trace}\left(BB^\mathsf{T}\right) - 2\text{trace}(B)$, almost surely.

Proof.

$$\begin{aligned}\mathsf{D}(\mathsf{B};\mathsf{Z}_{1:n}) = \frac{1}{n}\sum_{i=1}^{n}||(\mathsf{B}-\mathsf{I})\mathsf{Z}||^2 \to \mathbb{E}\left[||(\mathsf{B}-\mathsf{I})\mathsf{Z}||^2\right] &= \mathbb{E}\left[\sum_{i,j,k=1}^{d}\mathsf{Z}_i(\mathsf{B}-\mathsf{I})^{\mathsf{T}}_{i,j}(\mathsf{B}-\mathsf{I})_{j,k}\mathsf{Z}_k\right] \\ &= \sum_{i,j=1}^{d}(\mathsf{B}-\mathsf{I})^{\mathsf{T}}_{i,j}(\mathsf{B}-\mathsf{I})_{j,i} = \operatorname{trace}\left((\mathsf{B}-\mathsf{I})^{\mathsf{T}}(\mathsf{B}-\mathsf{I})\right) \\ &= \operatorname{trace}\left(\mathsf{B}^{\mathsf{T}}\mathsf{B}\right) + \operatorname{trace}(\mathsf{I}) - 2\operatorname{trace}(\mathsf{B}),\end{aligned}$$

where the convergence is almost sure. But $\operatorname{trace}\left(\mathsf{B}^{\mathsf{T}}\mathsf{B}\right) = \operatorname{trace}\left(\mathsf{B}\mathsf{B}^{\mathsf{T}}\right)$, giving the required result. □

Lemma 1. Let V, Λ, and U be as defined in Theorem 1, and let and $\lambda_i = \Lambda_{i,i}$ $(i = 1,\dots,d)$. Then

$$\sup_{M:MM^T=V}\operatorname{trace}(M) = \sum_{i=1}^{d}\lambda_i.$$

The supremum is achieved when $\mathsf{M} = \mathsf{U}\Lambda\mathsf{U}^{\mathsf{T}}$.

Proof. The rows of U form an orthonormal basis $\mathsf{e}_1,\dots,\mathsf{e}_d$, with $\mathsf{e}_i^{\mathsf{T}}\mathsf{V}\mathsf{e}_i = \lambda_i^2$ $(i = 1,\dots,d)$. For any matrix M with $\mathsf{M}\mathsf{M}^{\mathsf{T}} = \mathsf{V}$,

$$\lambda_i^2 = e_i^T MM^T e_i = ||M^T e_i||^2 = ||f_i||^2,$$

where $\mathsf{f}_i = \mathsf{M}^{\mathsf{T}}\mathsf{e}_i$. Next, recall that for any unitary matrix, U, and square matrix M, $\operatorname{trace}\left(\mathsf{U}\mathsf{M}\mathsf{U}^{\mathsf{T}}\right) = \operatorname{trace}(\mathsf{M})$. Thus, using the Cauchy-Schwarz inequality, and since U^{T} is also unitary,

$$\begin{aligned}\operatorname{trace}(\mathsf{M}) = \operatorname{trace}\left(\mathsf{U}^{\mathsf{T}}\mathsf{M}\mathsf{U}\right) &= \sum_{i,j,k=1}^{d}(\mathsf{e}_i)_j(\mathsf{e}_i)_k\mathsf{M}_{j,k} \\ &= \sum_{i=1}^{d}\mathsf{e}_i^{\mathsf{T}}\mathsf{M}\mathsf{e}_i = \sum_{i=1}^{d}\mathsf{f}_i^{\mathsf{T}}\mathsf{e}_i \le \sum_{i=1}^{d}||\mathsf{f}_i||\,||\mathsf{e}_i|| = \sum_{i=1}^{d}\lambda_i.\end{aligned}$$

The final part of the Lemma follows as $\operatorname{trace}\left(\mathsf{U}\Lambda\mathsf{U}^{\mathsf{T}}\right) = \operatorname{trace}(\Lambda) = \sum_{i=1}^{d}\lambda_i$. □

To prove Theorem 1, let $\mathsf{Z}_i = \mathsf{A}\mathsf{X}_i$, so $\mathbb{E}[\mathsf{Z}_i] = 0$ and $\operatorname{Var}[\mathsf{Z}_i] = \mathsf{I}_d$, and let $\mathsf{B} = \mathsf{A}^{-1}$ so $\mathsf{B}\mathsf{B}^{\mathsf{T}} = \mathsf{V}$. Since $\mathsf{D}(\mathsf{A};\mathsf{X}_{1:n}) = \mathsf{D}(\mathsf{B};\mathsf{Z}_{1:n})$, by Proposition 1, and then from Lemma 1, we have almost surely,

$$\begin{aligned}\mathsf{D}(\mathsf{A};\mathsf{X}_{1:n}) - \mathsf{D}(\mathsf{M}^{-1};\mathsf{X}_{1:n}) &= \mathsf{D}(\mathsf{B};\mathsf{Z}_{1:n}) - \mathsf{D}(\mathsf{M};\mathsf{Z}_{1:n}) \\ &\to 2\operatorname{trace}(\mathsf{M}) - 2\operatorname{trace}(\mathsf{B}) \ge 0. \quad \square\end{aligned}$$

The affine transformation (6) of SwISS is easy to apply to each batch of inflated sub-posterior samples, making the algorithm as fast and as simple to use as the consensus algorithm, with the guarantee of exactness in the Gaussian case. A visual representation of SwISS is given in Figure 1 and pseudo-code for implementing the algorithm is given in Algorithm 1.

4 | EXPERIMENTS

In this section we test the accuracy of the SwISS algorithm to merge batch posterior samples drawn from a variety of posterior distributions. We consider various complex posterior geometries to highlight the difference between affine transformations of posterior samples (*i.e.* SwISS) and averaging posterior samples (*i.e.* Consensus Monte Carlo). We also investigate the efficiency of alternative merging algorithms on popular statistical models with simulated and real data. We compare the SwISS algorithm against the following popular competing algorithms from the literature:

- **Consensus Monte Carlo** (Cons) algorithm (14), as described in Section 2.
- **Semiparametric density estimation** (SKDE)[1] from (11), where sub-posteriors are approximated semi-parametrically as described in (6).
- **Average re-centring** (AR) algorithm from (19), which is a special case of SwISS where $\mathbf{A}_b = \mathbf{I}$.
- **Gaussian Barycenter** (GB) algorithm (17), assuming a Gaussian approximation for each inflated sub-posterior, the barycenter is the geometric center of the inflated sub-posterior distributions.

[1] Implemented using the *parallelMCMCcombine* R package

Algorithm 1 SwISS Algorithm; here SPSQ$(\mathbf{V})$ denotes the symmetric positive-definite square root of the matrix $\mathbf{V}$ as described through (8) and (9).

```
Require: ${\theta_b^j}_{j=1}^J$ - J Monte Carlo samples from each of the B inflated posteriors

  Calculate the mean and variance for each of the inflated posteriors
  for b $\in \{1,\ldots,B\}$ do
      $\mu_b \leftarrow \text{mean}[\theta_b^{(1:J)}]$ and $\mathbf{V}_b \leftarrow \text{var}[\theta_b^{(1:J)}]$
  end for

  Set the global mean $\mu$ and variance $\mathbf{V}$ and calculate the matrix square root,
  $\mathbf{V} = \left(\frac{1}{B}\sum_{b=1}^B \mathbf{V}_b^{-1}\right)^{-1}$, $\mu = \mathbf{V}\frac{1}{B}\sum_{b=1}^B \mathbf{V}_b^{-1}\mu_b$, and $\mathbf{M} \leftarrow \text{SPSQ}(\mathbf{V})$

  Apply the affine transformation to the inflated posterior samples
  for b $\in \{1,\ldots,B\}$ do
      $\widetilde{\mathbf{V}}_b \leftarrow \mathbf{M}^{-1}\mathbf{V}_b\mathbf{M}^{-1}$
      $\widetilde{\mathbf{M}}_b \leftarrow \text{SPSQ}(\widetilde{\mathbf{V}}_b)$
      $\mathbf{A}_b \leftarrow \mathbf{M}\widetilde{\mathbf{M}}_b^{-1}\mathbf{M}^{-1}$
      Set $\vartheta_b^{1:J} \leftarrow \mathbf{A}_b(\theta_b - \mu_b) + \mu$
  end for

  Concatenate the transformed samples $\vartheta_b^{1:J}$ to give a Monte Carlo approximation of the full posterior distribution $\pi(\theta|\mathbf{y})$
  return $\left\{\vartheta_1^{(1:J)},\ldots,\vartheta_B^{(1:J)}\right\}$
```

We assess the accuracy of the above algorithms to combine batch posterior samples to form an approximation of the full posterior, comparing the merged approximations against the full posterior, which is generated by sampling (in serial) from the posterior conditional on the full data set. Accuracy of estimation of the posterior of the d-dimensional parameter, θ, is assessed with the following discrepancy measures:

- **Mahalanobis distance** (Mah):
$$D_{\text{Mah}} := \sqrt{(\mu_a - \mu_f)^\top \mathbf{V}_f^{-1}(\mu_a - \mu_f)},$$
where $\mathbf{V}_f$ and μ_f are the variance and mean estimates of posterior samples taken from the full data posterior using an MCMC algorithm. For a given posterior approximation algorithm, *e.g.* SwISS, μ_a denotes the estimated mean.
- **Mean absolute skew deviation** (Skew):
$$\eta := \frac{1}{d}\sum_{i=1}^d |\hat{\gamma}_i^a - \hat{\gamma}_i^f|,$$
where $\gamma_i = \mathbb{E}[\{(\theta_i - \mu_i)/\mathbf{V}_{ii}^{1/2}\}^3]$; *i.e.* η is the sum over components of the third standardised moments.
- **Integrated absolute distance** (IAD):
$$D_{\text{IAD}} := \frac{1}{2d}\sum_{j=1}^d \int |\hat{\pi}_j^a(\theta_j) - \hat{\pi}_j^f(\theta_j)| d\theta_j \in [0,1],$$
the average of the integrated absolute differences between two kernel density estimates of the marginal posteriors for each component, j, of θ: $\hat{\pi}_j^f$, obtained from samples from the true posterior, and $\hat{\pi}_j^a$ using one of the approximate merging algorithms (4).

4.1 | Scalability with parameter dimension

Typically, divide-and-conquer methods are advertised for use with tall data, *i.e.* a large number of observations and up to a moderate number of parameters. Here, we test the accuracy and computational speed of the merging algorithms as the number of parameters grows.

Let $\theta|\mathbf{y}_b \sim \mathcal{N}_d(\mu_b, V_b)$ for $b \in \{1,\ldots,B=10\}$, where d is the dimension of the parameter space and let $\mu_b \sim \mathcal{N}_d(0, I_d)$ and $V_b \sim \mathcal{W}^{-1}(5d, I_d)$. Each sub-posterior is Gaussian, with expectation and variance drawn respectively from Gaussian and inverse-Wishart distributions. For each experiment $J = 5,000$ samples were drawn from each sub-posterior and inflated sub-posterior. Using this model, the full data posterior is tractable:

$$\theta|\mathbf{y} \sim \mathcal{N}_d\left(V\sum_{b=1}^B V_b^{-1}\mu_b, V\right),$$

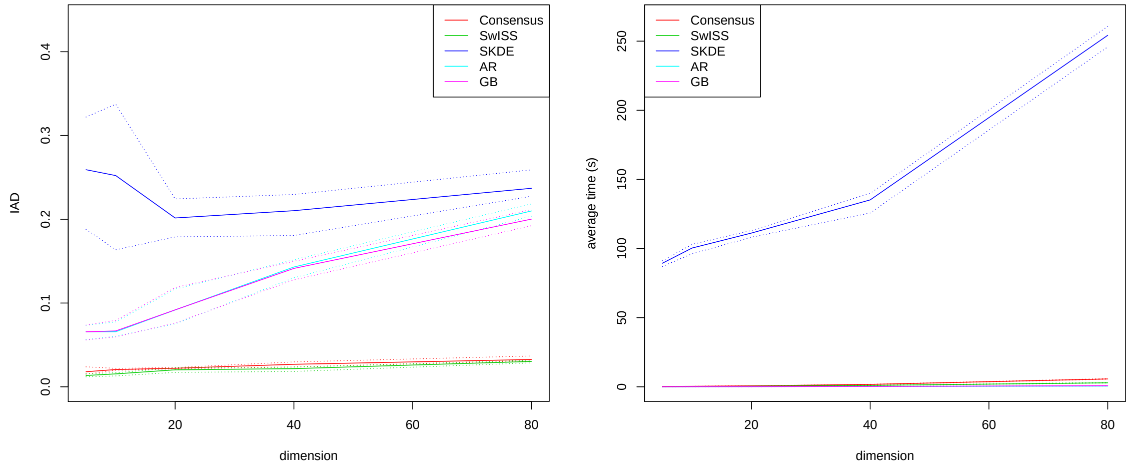

FIGURE 2 The left plot shows integrated absolute distance for each method for a different number of parameters. The right plot shows the mean time each combination method took to obtain the samples from an approximate posterior.

where $V^{-1} = \sum_{b=1}^{B} V_b^{-1}$. The following set of dimensions were used: $d \in \{5, 10, 20, 40, 80\}$.

Figure 2 shows that both the consensus Monte Carlo algorithm and SwISS perform well with increasing dimension (as measured by integrated absolute distance) and are both computationally efficient. The semi-parametric KDE approach, Gaussian barycenter and average re-centering approaches display reduced accuracy (as measured by integrated absolute distance). Only SwISS and consensus are robust to increasing the dimension of the parameter space. In terms of the computational cost required to merge the posterior samples, all approaches are generally fast, with the exception of the semi-parametric KDE approach.

4.2 | Linear Mixed Effects Model

A natural way to extend the simple linear model is to introduce both fixed and random effects. This extension can be particularly useful when data exhibit a hierarchical dependency structure, for example, to cluster student test scores based on classroom. Let $y_{i,j} \in \mathcal{Y} \subseteq \mathbb{R}$ (for $i, j = 1, \dots, n_j$, and $j = 1, \dots, n$) be the response variable, where n_j is the number of observations for group j. The fixed and random effects are $\mathbf{x}_{i,j} \in \mathcal{X} \subseteq \mathbb{R}^p$ and $\mathbf{z}_j \in \mathcal{Z} \subseteq \mathbb{R}^r$, respectively, and are related to the response variable by

$$y_{i,j}|\boldsymbol{\beta}, \boldsymbol{\alpha}_j, 1 \sim \text{Logistic}\ (\mathbf{x}_{i,j}\boldsymbol{\beta} + \mathbf{z}_j\boldsymbol{\alpha}_j, 1), \quad \boldsymbol{\alpha}_i \sim \mathcal{N}_r(0, \Sigma),$$

where $\boldsymbol{\beta} \in \mathbb{R}^p$ and $\boldsymbol{\alpha}_i \in \mathbb{R}^r$ are the fixed and random effect model coefficients and the $\text{Logistic}(a, 1)$ distribution has a cumulative distribution function of $1/(1 + \exp[-(x - a)])$. Our parameters of interest are then $\boldsymbol{\theta} = (\boldsymbol{\beta}, \boldsymbol{\alpha}, \Sigma)$, where Σ represents the variance of the random effects. We assume an inverse-Wishart distribution for the prior of Σ, $\Sigma \sim \mathcal{W}^{-1}(\nu, S)$, with $\nu = 5$ and $S = 5I_r$, and *a priori* we assumed $\boldsymbol{\beta} \sim \mathcal{N}_p(0, 1000I_p)$.

We simulated a dataset that contains $200,000$ observations. We set the number of groups $n = 2000$ and the number observations for each group $n_j = 100$, for $j = 1, \dots, n$. The number of parameters for the fixed effects were set to $p = 10$, with $\beta_{0,i} = (-1)^{(i-1)}$. The number of parameters for each random effect was set to be $r = 2$, and we set

$$\Sigma_0 = \begin{bmatrix} 1 & 0.2 \\ 0.2 & 1 \end{bmatrix},$$

then α_i were simulated independently from a $\mathcal{N}_r(0, \Sigma_0)$ distribution. We included an intercept term, that is $x_{i,j,1} = 1$ for all i, j, otherwise $\mathbf{x}_{i,j}$ and z_j were simulated from independent $\text{Bernoulli}(0.5)$ distributions.

The data were randomly partitioned into $B = 10$ batches by group, so that each group only belonged to one batch. This was necessary since divide and conquer methods assume independence between the batches. With a Gaussian observation model for the $y_{i,j}$, marginalisation over all of the random effects would be tractable. The logistic observation model necessitates the use of a sampling scheme such as MCMC.

We used the STAN software to sample from the full posterior and sub-posteriors generating $J = 5,000$ MCMC samples after an initial 1,000 sample burn in. Table 1 gives the discrepancy measures for each of the merging algorithms, averaged over 10 random partitions of the data. The

TABLE 1 Discrepancy measures for the linear mixed effects model on the simulated dataset. These measures were averaged over 5 random partitions of the data. Estimated standard errors are given in brackets.

Algorithm	Mah	Skew	IAD
SwISS	0.69 (0.14)	**0.02** (<0.01)	0.06 (0.01)
Consensus	**0.39** (0.13)	0.03 (0.01)	**0.04** (0.01)
Average Re-centring	2.20 (0.31)	**0.02** (<0.01)	0.13 (0.01)
Semi-parametric KDE	**0.39** (0.08)	0.04 (<0.01)	**0.04** (0.01)
Gaussian Barycenter	2.19 (0.31)	0.05 (0.01)	0.13 (0.01)

TABLE 2 Discrepancy measures for the logistic regression model with simulated and Hepmass data sets. Each metric was averaged over 5 runs. Estimated standard error are given in brackets.

	Simulated data			Hepmass data		
Algorithm	Mah	Skew	IAD	Mah	Skew	IAD
SwISS	**0.46** (0.30)	**0.04** (0.01)	**0.05** (0.03)	0.56 (0.05)	**0.03** (<0.01)	0.03 (<0.01)
Consensus	0.48 (0.35)	0.05 (0.01)	0.06 (0.03)	**0.35** (0.05)	**0.03** (<0.01)	0.03 (<0.01)
Average Re-centring	5.46 (3.92)	0.13 (0.07)	0.20 (0.03)	0.47 (0.04)	**0.03**(<0.01)	**0.02** (<0.01)
Semi-parametric KDE	1.25 (1.11)	0.76 (0.33)	0.12 (0.06)	0.36 (0.05)	**0.03**(<0.01)	0.03 (<0.01)
Gaussian Barycenter	5.42 (3.75)	**0.04** (0.01)	0.20 (0.01)	0.46 (0.05)	**0.03**(<0.01)	**0.02** (<0.01)

results show that all algorithms perform well, with the exception of AR and the Gaussian barycenter, both on the Mahalanobis metric. The SwISS and Consensus algorithms are robust across the range of metrics.

4.3 | Logistic Regression Model

Logistic regression is a popular technique for modelling binary data, *i.e.* $\mathrm{y_i} \in \{0,1\}$. Features $\mathbf{x}_\mathrm{i} \in \mathbb{R}^\mathrm{d}$, also known as covariates, that can indicate the classification outcome are mapped onto the binary observations using a logit transformation, where the outcome probability $\mathbb{P}(\mathrm{y_i} = 1) = \exp(\mathbf{x}_\mathrm{i}^\top \boldsymbol{\theta}) / \left(1 + \exp(\mathbf{x}_\mathrm{i}^\top \boldsymbol{\theta})\right)$, is the success probability of a Bernoulli random variable. Our parameter of interest $\boldsymbol{\theta} \in \mathbb{R}^\mathrm{d}$ is the vector of coefficients.

We consider two data sets, the first is a synthetic data set which is designed to simulate a scenario with rare but highly informative features. This data set is similar to the one given in (14). We simulate $\mathrm{N} = 100,000$ data points with $\mathrm{d} = 5$ binary features with relative frequencies of $\mathbf{x}_\mathrm{i} = 1$ being $(1, 0.02, 0.03, 0.05, 0.001)$ and the corresponding true parameter values are $\boldsymbol{\theta}_0 = (-3, 1.2, -0.5, 0.8, 3)$. Due to the rarely occurring final feature, this can lead to largely differing variances across the sub-posteriors. For our experiments, we spilt the data equally across $\mathrm{B} = 25$ batches.

We also consider a real-world data set; the Hepmass data set[2] from high-energy particle physics where the response is an indicator for whether a signal was indicative of an exotic particle being present as opposed to background noise. The data set contains 27 real features which we augmented with an intercept term to give $\mathrm{d} = 28$ parameters. The full data set contains 10.5 million responses, in this experiment we considered the first $\mathrm{N} = 100,000$ and split the data across $\mathrm{B} = 20$ batches.

In each of the our experiments, the data were repeatedly partitioned $\mathrm{n_{runs}} = 5$ times with a Monte Carlo average of the discrepancy metrics given in Table 2 . The STAN (3) software, which implements an automatically-tuned version of Hamiltonian Monte Carlo sampling, was used as the MCMC sampler and applied to the full posterior and sub-posteriors for each experiment. Each sampler drew $\mathrm{J} = 10,000$ samples after a burn in of $1,000$ iterations.

The results in Table 2 show that SwISS and the Consensus algorithm outperform all of the others on the simulated data, whereas for the real example all of the methods work well. The AR algorithm performs especially poorly on the synthetic data example as the variance of the sub-posteriors varies across subposteriors, and the AR algorithm does not correct for this when the sub-posterior samples are merged. The similarity of eprformance on the Hepmass data could be due to the sub-posteriors all being close to Gaussian. We would expect both Consensus and SwISS to work well in this setting as they are exact for Gaussian sub-posteriors, and much faster to apply than nonparametric methods such as SKDE.

[2] http://archive.ics.uci.edu/ml/datasets/HEPMASS

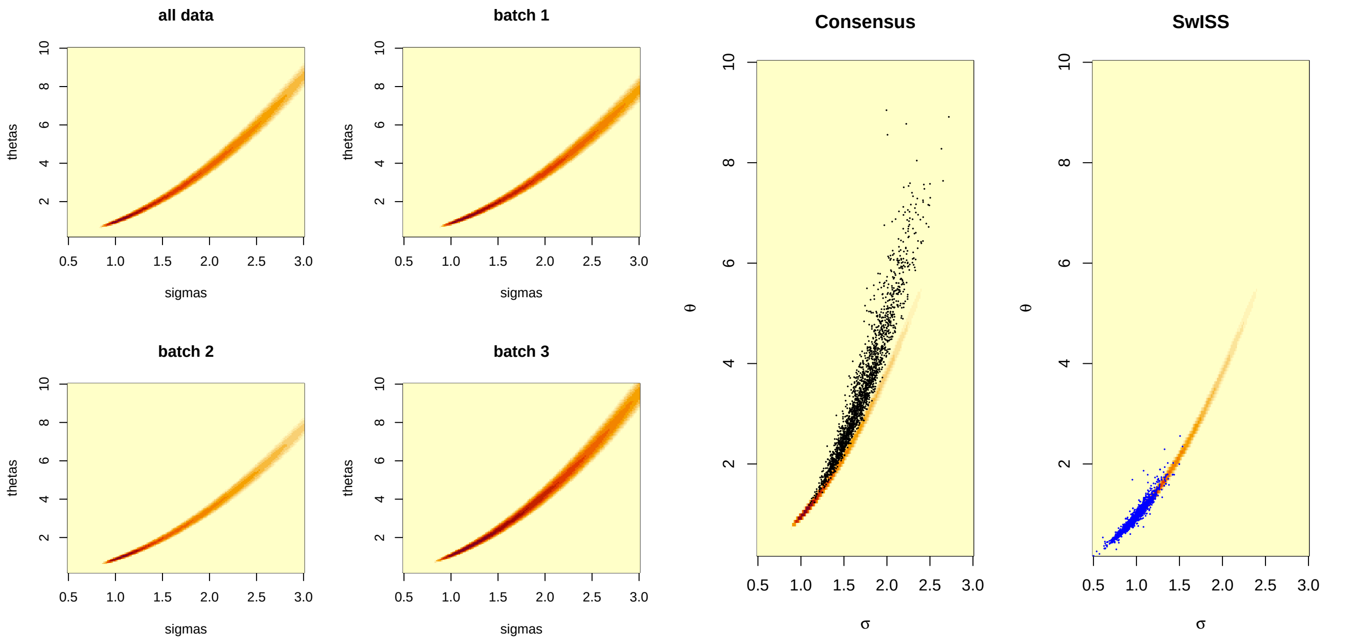

FIGURE 3 Left-hand plots: Four plots which show the two-dimensional full data posterior and three of the batch posteriors. Right-hand plots: A comparison is given between the consensus Monte Carlo algorithm (left) with black points and the SwISS algorithm (right) with blue points, compared against the true full posterior (orange points).

4.4 | Ornstein-Uhlenbeck Processes

Consider an Ornstein-Uhlenbeck (OU) process centered at zero, which satisfies the stochastic differential equation (SDE)

$$dX_t = -\frac{1}{2}\theta dt + \sigma dW_t. \tag{10}$$

This has stationary distribution $\mathrm{N}(0, \sigma^2/\theta)$. If we assume that the process is started from stationarity, then for $\mathrm{s} \le \mathrm{t}$, $\mathrm{Cov}[\mathrm{X_s}, \mathrm{X_t}] = \frac{\sigma^2}{\theta} \exp[-\theta(\mathrm{t} - \mathrm{s})/2]$. Suppose we observe $\mathrm{X}_0, \mathrm{X}_\Delta, \mathrm{X}_{2}\Delta, \ldots, \mathrm{X}_\mathrm{M}\Delta$ for a very large M. If $\Delta\theta >> 1$ then consecutive observations are almost uncorrelated and the likelihood is close to that of a set of iid draws from the stationary distribution. Thus σ^2/θ is well identified, but the individual parameters are not. The likelihood is,

$$L(\theta, \sigma; x_0, \ldots, x_{M\Delta}) = N(x_0; 0, \sigma^2/\theta) \prod_{m=1}^{M} N(x_m; x_{m-1} \exp(-\theta\Delta/2), \sigma^2(1 - \exp(-\theta\Delta))/\theta) \tag{11}$$

If we observe $\mathrm{B} = 10$ realisations of the same OU process, $\mathrm{x}_0^{(\mathrm{b})}, \mathrm{x}_1^{(\mathrm{b})}, \ldots, \mathrm{x}_{\mathrm{M}\Delta}^{(\mathrm{b})}$, for $\mathrm{b} = 1, 2, \ldots, \mathrm{B}$, with $\theta = 1, \sigma = 1, \Delta = 6$ and $\mathrm{M} = 365$, then then the likelihood will be

$$\prod_{b=1}^{B} L(\theta, \sigma; x_0^{(b)}, x_1^{(b)}, \ldots, x_{M\Delta}^{(b)}). \tag{12}$$

This example is interesting because the individual batches are centered at different points along a quadratic curve (see Figure 3), so averaging them, as the consensus algorithm will do, does not give a value on that same curve. Since the curve is convex, the average will give a value above the curve. This is illustrated in Figure 3 and as we can see, if we try to estimate the stationary variance σ^2/θ from the consensus samples for (σ, θ), it will be an overestimate.

We can see from Figure 3 that the mean of the batch means will lead to an overestimate of the mean for σ^2/θ but this effect is significantly smaller for the SwISS algorithm than for the consensus algorithm as the batch means are more concentrated round the true curve. Figure 3 (left) also shows a scatter plot of the batch and full data posteriors to highlight the areas of highest posterior concentration. The scatter plots are slightly misleading as for the full posterior there is a long right tail which leads to some large σ and θ values. However, the 95% upper quantiles for σ and θ are 1.73 and 2.98, respectively, and the SwISS algorithm accurately captures this area of highest posterior mass.

5 | CONCLUSIONS

We have introduced a new method to merge posterior samples generated in parallel on independent batches of data. Our algorithm, SwISS, is fast, scalable to high-dimensional settings, and accurate on a variety of test cases. The SwISS algorithm, like the consensus Monte Carlo algorithm, is simple to apply and competitive against popular alternative divide-and-conquer algorithms. SwISS also has the advantage that it does not require hyper-parameter tuning and is faster to apply than many of the alternative divide-and-conquer algorithms given in the literature. We have provided theoretical support for our choice of affine transformations and shown that SwISS is exact in the case of merging inflated Gaussian sub-posteriors. Code to recreate this work is available through the Github link: https://github.com/CJohnVyner/SwISS

5.1 | Acknowledgements

The authors gratefully acknowledge the support of the UK Engineering and Physical Sciences Research Council grants EP/S00159X/1, EP/V022636/1 and EP/P033075/1.

References

Baker, J., Fearnhead, P., Fox, E. B. and Nemeth, C. (2019). Control variates for stochastic gradient MCMC. Statistics and Computing, 29(3), 599-615.

Baker, J., Fearnhead, P., Fox, E. B. and Nemeth, C. (2019). sgmcmc: An R Package for Stochastic Gradient Markov Chain Monte Carlo. Journal of Statistical Software, 91, 1-27.

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., ... and Riddell, A. (2017). Stan: A probabilistic programming language. Journal of statistical software, 76(1).

Chan, Ryan S., Pollock, M., Johansen, A.M. and Roberts, G.O. (2021). Divide-and-Conquer Monte Carlo Fusion. arXiv preprint arXiv:2110.07265.

Entezari, R., Craiu, R. V. and Rosenthal, J. S. (2018). Likelihood inflating sampling algorithm. Canadian Journal of Statistics, 46(1), 147-175.

Hjort, N. L. and Glad, I. K. (1995). Nonparametric density estimation with a parametric start. The Annals of Statistics, 882-904.

Le Cam, L. M. and Yang, G. L. (2000). Asymptotics in statistics: some basic concepts. Springer Science and Business Media.

Minsker, S., Srivastava, S., Lin, L. and Dunson, D. (2014, June). Scalable and robust Bayesian inference via the median posterior. In International conference on machine learning (pp. 1656-1664).

Nemeth, C. and Sherlock, C. (2018). Merging MCMC subposteriors through Gaussian-process approximations. Bayesian Analysis, 13(2), 507-530.

Nemeth, C. and Fearnhead, P. (2021). Stochastic gradient Markov chain Monte Carlo. Journal of the American Statistical Association, 116(533), 433-450.

Neiswanger, W., Wang, C. and Xing, E. (2014). Asymptotically exact, embarrassingly parallel MCMC. Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence, 623-632.

Quiroz, M., Tran, M. N., Villani, M. and Kohn, R. (2018). Speeding up MCMC by delayed acceptance and data subsampling. Journal of Computational and Graphical Statistics, 27(1), 12-22.

Quiroz, M., Kohn, R., Villani, M. and Tran, M. N. (2018). Speeding up MCMC by efficient data subsampling. Journal of the American Statistical Association.

Scott, S. L., Blocker, A. W., Bonassi, F. V., Chipman, H. A., George, E. I. and McCulloch, R. E. (2016). Bayes and big data: The consensus Monte Carlo algorithm. International Journal of Management Science and Engineering Management, 11(2), 78-88.

Sherlock, C., Golightly, A. and Henderson, D. A. (2017). Adaptive, delayed-acceptance MCMC for targets with expensive likelihoods. Journal of Computational and Graphical Statistics, 26(2), 434-444.

Srivastava, S., Cevher, V., Dinh, Q. and Dunson, D. (2015, February). WASP: Scalable Bayes via barycenters of subset posteriors. In Artificial Intelligence and Statistics (pp. 912-920).

Srivastava, S., Li, C. and Dunson, D. B. (2018). Scalable Bayes via barycenter in Wasserstein space. The Journal of Machine Learning Research, 19(1), 312-346.

Welling, M. and Teh, Y. W. (2011). Bayesian learning via stochastic gradient Langevin dynamics. In Proceedings of the 28th international conference on machine learning (ICML-11) (pp. 681-688).

Wu, C. and Robert, C. P. (2017). Average of recentered parallel MCMC for big data. arXiv preprint arXiv:1706.04780.

How to cite this article: C. Vyner, C. Nemeth, and C. Sherlock (2022), SwISS: A Scalable Markov chain Monte Carlo Divide-and-Conquer Strategy, *?*, *???;??:?–?*.