Stat Comput (2018) 28:277–290
https://doi.org/10.1007/s11222-017-9730-1

Ensemble preconditioning for Markov chain Monte Carlo simulation

Benedict Leimkuhler[1] · **Charles Matthews**[2] · **Jonathan Weare**[3]

Received: 15 September 2016 / Accepted: 23 January 2017 / Published online: 27 February 2017

Abstract We describe parallel Markov chain Monte Carlo methods that propagate a collective ensemble of paths, with local covariance information calculated from neighbouring replicas. The use of collective dynamics eliminates multiplicative noise and stabilizes the dynamics, thus providing a practical approach to difficult anisotropic sampling problems in high dimensions. Numerical experiments with model problems demonstrate that dramatic potential speedups, compared to various alternative schemes, are attainable.

Keywords Stochastic sampling · Markov chain Monte Carlo · MCMC · Computational statistics · Machine learning · BFGS · Langevin methods · Brownian dynamics

BL is supported by EPSRC Grants EP/K035912/1 (ExTASY) and EP/N510129/1 (The Alan Turing Institute).
JQW and CM are supported by the Advanced Scientific Computing Research Program within the DOE Office of Science through award DE-SC0014205 as well as through a contract from Argonne, a U.S. Department of Energy Office of Science Laboratory.

✉ Charles Matthews
c.matthews@uchicago.edu

Benedict Leimkuhler
b.leimkuhler@ed.ac.uk

Jonathan Weare
weare@uchicago.edu

[1] School of Mathematics, University of Edinburgh, Edinburgh EH93FD, UK

[2] Department of Statistics, University of Chicago, Chicago, IL 60615, USA

[3] Department of Statistics and James Franck Institute, University of Chicago, Chicago, IL 60615, USA

1 Introduction

A popular family of methods for Bayesian parameterization in data analytics are derived as Markov chain Monte Carlo (MCMC) methods, including Hamiltonian (or hybrid) Monte Carlo (HMC)(Duane et al. 1987; Neal 2011; Monnahan et al. 2016), or the Metropolis adjusted Langevin algorithm (MALA)(Rossky et al. 1978; Bou-Rabee and Vanden-Eijnden 2010; Roberts and Tweedie 1996). These methods involve proposals that are based on approximating a continuous-time (stochastic) dynamics that exactly preserves the target (posterior) density π, followed by an accept/reject step to correct for approximation errors.

Efficient parameterization of the stochastic differential equations used in these procedures has the potential to greatly accelerate their convergence, particularly when the target density is poorly scaled, i.e. when the Hessian matrix of the logarithm of the density has a large condition number (an example is given in "Appendix 1"). In precise analogy with well-established strategies in optimization (see e.g. Sun and Yuan 2006), the solution to conditioning problems in the sampling context is to find a well-chosen change of variables (preconditioning) for the system, such that the natural scales of the transformed system are roughly commensurate.

In this article, we discuss an approach to dynamic preconditioning based on simultaneously evolving an ensemble of parallel MCMC simulations, each of which is referred to as a "walker" or "particle". As we will show, the walkers provide information that can greatly improve the efficiency of MCMC methods. There is a long history of using multiple parallel simulations to improve MCMC calculations (see e.g. (Gilks et al. 1994; ter Braak 2006; Goodman and Weare 2010; Andrés Christen and Fox 2010; Jasra et al. 2007; Cappé et al. 2004; Iba 2001; Hairer and Weare 2014; Hammersley and Morton 1954; Liu 2002; Rosenbluth and Rosenbluth

1955)). Many of these methods rely on occasional duplication or removal of walkers and reweighting of samples to speed sampling of densities with multiple modes or to compute tail averages. The schemes proposed in this article are more similar to methods introduced in (Gilks et al. 1994; ter Braak 2006; Andrés Christen and Fox 2010; Goodman and Weare 2010) that address conditioning issues using walker proposal moves informed by the positions of other walkers in the ensemble. These methods are not designed to directly address multimodality and do not involve any reweighting of samples. Our approach differs in that proposal moves are derived from time discretization of an SDE whose solutions exactly preserve π (or more precisely the joint density of an ensemble of independent random variables drawn from π). This results in ensemble MCMC schemes that converge rapidly on poorly conditioned distributions even in relatively high-dimensional sample spaces and when the details of the conditioning problems depend on position in sample space.

Our starting point is the discrete approximation of a system of SDEs for a state vector $x \in \mathcal{D} \subset \mathbb{R}^D$,

$$\dot{x} = (J(x) + S(x))\nabla \log \pi(x) + \operatorname{div}(J(x) + S(x)) + \sqrt{2S(x)}\,\eta(t) \tag{1}$$

where $J(x)$ and $S(x)$ are skew-symmetric and symmetric positive semi-definite $D \times D$ matrices, respectively, with $\eta(t)$ representing a vector of independent Gaussian white noise components. In our sampling schemes, each walker generates a discrete-time approximation of (1) with its own particular choice of J which corresponds to a notion of the localized and regularized sample covariance matrix across the ensemble of walkers and incorporates information about the target density π into the evolution of each walker.

Many existing sampling methods can be characterized as time discretizations of (1) (Ma et al. 2015). The matrix S is sometimes referred to as a mass matrix (though we reserve that term for a different matrix) and is often chosen to be diagonal. More general modifications of S (with $J = 0$) to improve convergence have been considered in the Monte Carlo literature, dating at least to (Bennett 1975). This idea has been the focus of renewed attention in statistics, and several recent approaches concerning this or related ideas have been proposed (Martin et al. 2012; Girolami and Calderhead 2011a). Though modification of S appears to be much more common in practice, several authors have considered the effect that the choice of J and S has on the ergodic properties of the solution to (1) from a more theoretical perspective (see e.g. (Rey-Bellet and Spiliopoulos 2015; Duncan et al. 2016; Hwang et al. 2005, 1993)). In this paper, we are concerned with motivating and presenting a particular choice of S and J based on the ensemble framework mentioned above and yielding practical and efficient sampling schemes. We demonstrate that the choice of J and S has important ramifications for the stability of the discretization scheme as well as for the overall sampling efficiency. This interplay will be explored in future work

2 Preconditioning strategies for sampling

As in any MCMC scheme, the goal is to estimate the average $\mathrm{E}[f] = \int f(x)\pi(x)\mathrm{d}x$ by a trajectory average of the form

$$\overline{f}_N = \frac{1}{N}\sum_{n=0}^{N-1} f(x^{(n)}),$$

for large N. In many cases, we can expect the error in an MCMC scheme to satisfy a central limit theorem: $\sqrt{N}\left(\overline{f}_N - \mathrm{E}[f]\right) \xrightarrow{dist} N(0, \tau\sigma^2)$, where σ^2 is the variance of f under π (and is independent of the particular MCMC scheme), the τ the integrated autocorrelation time (IAT) which is often used to quantify the efficiency of an MCMC approach (see "Appendix 1").

To emphasize an analogy with optimization, for the moment assume that $J = 0$. The steepest descent algorithm of optimization corresponds to an Euler–Maruyama discretization of the so-called overdamped Langevin (or Brownian) dynamics (Milstein and Tretyakov 2004; Pavliotis 2014),

$$x^{(n+1)} = x^{(n)} + \delta t\, \nabla \log(\pi(x^{(n)})) + \sqrt{2\delta t}\,\mathrm{R}^{(n)} \tag{2}$$

where $\mathrm{R} \sim N(0, I)$. Discretization introduces an $O(\delta t)$ error in the sampled invariant distribution so a Metropolis–Hastings accept/reject step may be incorporated in order to recover the correct statistics (see the MALA algorithm (Rossky et al. 1978)) when time discretization error dominates sampling error. Reducing δt gives a more accurate approximation of the evolution of the dynamics and boosts the acceptance rate.

When π is Gaussian with covariance Σ, one can easily show that the cost to achieve a fixed accuracy depends on the condition number $\kappa = \lambda_{\max}/\lambda_{\min}$ where $\lambda_{\max}$ and $\lambda_{\min}$ are the largest and smallest eigenvalues of Σ. Indeed, one finds that the worst-case IAT τ for the scheme in (2) over observables of the form $v^{\mathrm{T}}x$ is $\tau = \kappa - 1$ (see "Appendix 1"). In this formula, the eigenvalue λ_{min} arises due to the discretization stability constraint on the stepsize parameter δt and $\lambda_{\max}$ appears because the direction of the corresponding eigenvector is slowest to relax for the continuous-time process. The presence of $\lambda_{\min}$ in this formula indicates that analysis of the continuous-time scheme (1) (i.e. neglect of the discretization stability constraint) can be misleading when considering the effects of poor conditioning on sampling efficiency. Since

the central limit theorem suggests that the error after N steps of the scheme is roughly proportional to $\sqrt{\tau/N}$, the cost to achieve a fixed accuracy is again roughly proportional to κ.

Continuing to use $J=0$, taking $S(x)=-(\nabla^2 \log(\pi(x)))^{-1}$ in (1) and discretizing with timestep $\delta t > 0$, we obtain a stochastic analogue of Newton's method:

$$x^{(n+1)} = x^{(n)} + \delta t\, S(x^{(n)})\nabla \log(\pi(x^{(n)})) + \delta t \operatorname{div}\left(S(x^{(n)})\right) + \sqrt{2\delta t S(x^{(n)})}\,\mathrm{R}^{(n)}. \quad (3)$$

Schemes of a similar form though neglecting the $\operatorname{div}(S)$ term (and therefore requiring Metropolization) have been explored recently in (Martin et al. 2012). Metropolization may also be used to correct the $O(\delta t)$ sampling bias introduced by the discretization. It can be shown that the scheme is affine invariant in the sense that when it is applied to sampling $\pi_{A,v}$ it generates a sequence of samples $y^{(n)}$ so that $x^{(n)} = Ay^{(n)} + v$ has exactly the same distribution as the sequence of samples generated by the method when applied to π (see Goodman and Weare 2010 for a detailed discussion of the role of affine invariance in the design of MCMC methods for poorly conditioned problems). We therefore expect that when this method can be applied (e.g. when the Hessian is positive definite), it should be effective on poorly scaled problems. This affine invariance property is shared by the deterministic Newton's method (obtained from (3) by dropping the noise and matrix divergence terms) and is responsible for its good performance when applied to optimizing poorly scaled functions (e.g. when the condition number of the Hessian is large). We stress that the key to the usefulness of either the deterministic or stochastic Newton's method is that one does not need to make an explicit choice of the matrix A or the vector v. As the performance is independent of the choice of A and v, we can assume that A or v is chosen to improve the conditioning of the problem.

Due to the presence of the divergence term in the continuous dynamics, discretization will require evaluation of first-, second- and third-order derivatives of $\log(\pi(x))$, making it prohibitively expensive for many models. To avoid this difficulty, one can estimate the divergence term using an extra evaluation of the Hessian (see "Appendix 6"), or omit the divergence term and rely on a Metropolization step to ensure correct sampling. Regardless of how this term is handled, the system (3), unlike (2), is based on multiplicative noise (where the magnitude of the noise process depends upon the state of the system) which is known to introduce complexity (and reduce accuracy) in numerical discretization (Milstein and Tretyakov 2004).

More fundamentally, complex sampling problems will exhibit regions of substantial probability where the Hessian fails to be positive definite. A simple (and often more robust alternative) is $S = \Sigma$, where Σ is the covariance matrix of π (even when π is not Gaussian) and is positive definite. It can be shown that the iteration in (3) is again affine invariant. The resulting scheme, which can be regarded as a simple quasi-Newton type approach, is closely related to adaptive MCMC approaches (Roberts and Rosenthal 2007; Haario et al. 2001). On the other hand, because this choice of S does not depend on position, the scheme can be expected to perform poorly on problems for which the conditioning is dramatically different in different regions of space (e.g. the Hessian has high condition number and its eigenvectors are strongly position dependent), see Fig. 1. These observations suggest a choice of S corresponding to a notion of local covariance.

While a notion of local covariance will be central to the schemes we eventually introduce, we choose to incorporate that information not through S in (1), but through the skew-symmetric matrix J in that equation. In the remainder of this section, we discuss how the choices of S described so far, and the corresponding properties of (3), have analogues in choices of J and a family of so-called underdamped Langevin schemes that we next introduce Pavliotis (2014), Leimkuhler and Matthews (2015).

A popular way to obtain an MCMC scheme with decreased IAT relative to the overdamped scheme in (2) is to introduce "inertia". We extend the space by writing our state $x = (q, p)^{\mathrm{T}} \in \mathcal{D} \times \mathbb{R}^D \subset \mathbb{R}^{2D}$, with the target distribution

$$\hat{\pi}(x) = \hat{\pi}(q, p) = \pi(q)\varphi(p), \quad \int \hat{\pi}(q, p)\mathrm{d}p = \pi(q). \quad (4)$$

The distribution of interest $\pi(q)$ is recovered from $\hat{\pi}(q, p)$ as the marginal distribution of the position vector q. For the distribution $\varphi(p)$ we will follow common practice and use $\varphi(p) \propto \exp(-\|p\|^2/2)$. With this extension of the space, we recover the standard underdamped form of Langevin dynamics using

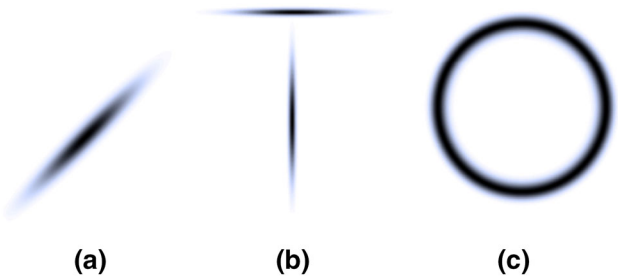

Fig. 1 We plot three examples of posterior distribution functions that might be described as poorly scaled. The distribution **a** has a scaling that can be removed through a linear change of variables, whereas useful scaling information in distributions **b** and **c** depends on the location in space. Proposals can benefit from taking into account local scaling behaviour over the global covariance information (the condition number of their covariance matrices in both **b** and **c** are unity)

$$J = \begin{bmatrix} 0 & -I_D \\ I_D & 0 \end{bmatrix}, \qquad S = \begin{bmatrix} 0 & 0 \\ 0 & \gamma I_D \end{bmatrix} \tag{5}$$

in equation (1), where I_D is the $D \times D$ identity matrix and γ is a positive constant (Milstein and Tretyakov 2004). Recent work Dalalyan (2016), Durmus and Moulines (2016), especially in connection with molecular dynamics (Leimkuhler et al. 2015), has examined efficient ways to discretize Langevin dynamics while minimizing the error in sampling $\pi(q)$.

To incorporate information such as the Hessian matrix or the covariance matrix (or local covariance matrices) in the underdamped Langevin scheme, we focus on choices of J and S as follows:

$$J(x) = \begin{bmatrix} 0 & -B(q) \\ B(q)^{\mathrm{T}} & 0 \end{bmatrix}, \qquad S = \begin{bmatrix} 0 & 0 \\ 0 & \gamma I_D \end{bmatrix},$$

where $B(q)B^T(q)$ is a symmetric positive definite matrix, resulting in the system

$$\begin{aligned} \dot{q} &= B(q)p, \\ \dot{p} &= B(q)^{\mathrm{T}}\nabla \log(\pi(q)) + \mathrm{div}(B(q)^{\mathrm{T}}) - \gamma p + \sqrt{2\gamma}\eta(t). \end{aligned} \tag{6}$$

Discretization of the stochastic system may be derived by mimicking the BAOAB scheme (Leimkuhler et al. 2015). Given a stepsize $\delta t > 0$, define $\alpha = \exp(-\gamma \delta t)$ and approximate the step from t_n to $t_{n+1} = t_n + \delta t$ by the formulas

$$p^{(n+1/2)} = p^{(n)} + \frac{\delta t}{2} F(q^{(n)}), \tag{7a}$$

$$q^{(n+1/2)} = q^{(n)} + \frac{\delta t}{2} B\left(q^{(n+1/2)}\right) p^{(n+1/2)} \tag{7b}$$

$$\begin{aligned} \hat{p}^{(n+1/2)} = \alpha p^{(n+1/2)} &+ \frac{(\alpha+1)\delta t}{2} \mathrm{div}\left(B\left(q^{(n+1/2)}\right)^{\mathrm{T}}\right) \\ &+ \sqrt{1-\alpha^2}\mathrm{R}^{(n)} \end{aligned} \tag{7c}$$

$$q^{(n+1)} = q^{(n+1/2)} + \frac{\delta t}{2} B\left(q^{(n+1/2)}\right) \hat{p}^{(n+1/2)} \tag{7d}$$

$$p^{(n+1)} = \hat{p}^{(n+1/2)} + \frac{\delta t}{2} F(q^{(n+1)}) \tag{7e}$$

where $\mathrm{R} \sim N(0, I_D)$ and $F(q) = B(q)^{\mathrm{T}}\nabla \log \pi(q)$, with an implicit equation in (7b). The choice of matrix BB^{T} introduced in the next section will be a sum of the identity and a small (relative to the dimension D) number of rank 1 matrices, alleviating storage demands and reducing the cost of all calculations involving B to linear in D. As described in "Appendix 2", schemes of the form in (7) can also be used to generate proposals in a Metropolis–Hastings framework to strictly enforce a condition that, like detailed balance, guarantees that π is exactly preserved.

Suppose that, when applied to sampling the density $\pi_{A,v}$, an underdamped Langevin scheme of the form in (7) generates a sequence $(q^{(n)}, p^{(n)})$. The scheme will be referred to as affine invariant if the transformed sequence $(Aq^{(n)} + v, p^{(n)})$ has the same distribution as the sequence generated by the method when applied to sample π. As for (3) one can demonstrate that the choices $B(q)B^{\mathrm{T}}(q) = -(\nabla^2 \log(\pi(q)))^{-1}$ and $B(q)B^{\mathrm{T}}(q) = \Sigma$, yield affine invariant sampling schemes (see "Appendix 5" for details). Recall that the choice of $S(x) = -(\nabla^2 \log(\pi(x)))^{-1}$ in (3) also gave an affine invariant scheme, but that there the S matrix appears multiplying the noise (making it multiplicative).

Before proceeding to the important issue of selecting a practically useful choice of B, we observe the following important properties of our formulation: (i) the stochastic dynamical system (6) exactly preserves the target distribution (see Ma et al. 2015) and thus, if discretization error is well controlled, Metropolis correction is not necessarily needed for the computation, and (ii) the formulation, with appropriate choice of B, is affine invariant, even under discretization (see "Appendix 5"), a property which ensures the stability of the method under change of coordinates. By contrast, we emphasize that schemes that modify S (instead of J) in (5) or that are based on a q-dependent normal distribution φ in (4) (e.g. within HMC as in (Girolami and Calderhead 2011a)), cannot be made affine invariant in the same sense, though they can be made to satisfy an alternative notion of affine invariance (see "Appendix 5").

With the general stochastic quasi-Newton form in (7) as a template, one may consider many possible choices of B. Just as in optimization, in MCMC the question is not whether one should precondition, but rather how can one precondition in an affordable and effective way. Unfortunately, practical and effective quasi-Newton approaches for optimization do not have direct analogues in the sampling context, leaving a substantial gap between un-preconditioned methods and often impractical preconditioning approaches. In the next section, we suggest an alternative strategy to fill this gap: using multiple copies of a simulation to incorporate local scaling information in the B matrix in (7).

3 Ensemble quasi-Newton (EQN) schemes

We next describe an efficient MCMC approach in which information from an ensemble of walkers provides an estimate of a modified local covariance matrix. We consider a system of L walkers (independent copies evolving under the same dynamics) with state $x_i = (q_i, p_i)^{\mathrm{T}}$, where subscripts now indicate the walker index. Each walker has position q_i and momentum p_i for $i = 1, \cdots, L$, and we define the vectors $Q = (q_1, q_2, \ldots, q_L)^{\mathrm{T}} \in \mathcal{D}^L$ and $P =$

$(p_1, p_2, \ldots, p_L)^{\mathrm{T}} \in \mathbb{R}^{DL}$. We seek to sample the product measure $\bar{\pi}$ whose marginals give copies of the distribution of interest π:

$$\bar{\pi}(Q, P) = \prod_{i=1}^{L} \hat{\pi}(q_i, p_i), \qquad \int \bar{\pi}(Q, P) \mathrm{d}P = \prod_{i=1}^{L} \pi(q_i).$$

A simple strategy is for each walker to sample $\bar{\pi}$ by evolving each x_i independently using an equation such as (2) or (5). Such a method scales perfectly in parallel when initial conditions are drawn from the target distribution, but no use is made of the local observed geometry or inter-walker information. Alternatively we may use the dynamics (6) to introduce walker information through the $B(q)$ preconditioning matrix in order to scale the dynamics based upon information from the other walkers. This preconditioning enters into the dynamics but not the invariant distribution which remains $\bar{\pi}$. A popular alternative preconditioning strategy is to modify the mass matrix, i.e. the covariance of the Gaussian distribution φ in (4) (see e.g. Girolami and Calderhead 2011a or "Appendix 5"). In our context of ensemble-based schemes, this strategy would introduce substantial (and costly) communication between walkers at each evolution step.

Using L walkers, the global state $x = (Q, P)$ consists of $2DL$ total variables and $B(Q)$ is a $DL \times DL$ matrix. We will use $B(Q) = \mathrm{diag}(B_1(Q), B_2(Q), \ldots, B_L(Q))$ with each $B_i(Q) \in \mathbb{R}^{D \times D}$ so that the position and momentum (q_i, p_i) of walker i evolve according to (7) with $B(q)$ replaced by $B_i(Q)$. Note that the divergence and gradient terms in the equation for each walker are taken with respect to the q_i variable.

Within this quasi-Newton framework, there are many potential choices for the B_i matrix, with $B_i = I_D$ reducing to the simulation of L independent copies of underdamped Langevin dynamics. Before exploring the possibilities, we remark that, in order to exploit parallelism, we will divide our L walkers into several groups of equal size in an approach similar to the emcee package (Foreman-Mackey et al. 2013). Walkers in the same group $g(i)$ as walker i will *not* appear in B_i so that the walkers in any single group can be advanced in parallel independently. The fact that B_i is independent of walkers in the same group as walker i is vital when we introduce the Metropolis step to exactly preserve the target distribution (see "Appendix 2").

We set $Q_{[i]} = \{q_j \mid g(j) \neq g(i)\}$ and let K be the common size of these sets. For example, if we have 16 cores available we may wish to use ten groups of 16 walkers (so $L = 160$ and $K = 144$). If walker j is designated as belonging to group 1, it evolves under the dynamics given in equation (7) but the set $Q_{[j]}$ only includes walkers in groups $2, \ldots, 10$. We may then iterate over the groups of walkers sequentially, moving all the walkers in a particular group in parallel with the others.

One choice for the preconditioning matrix (not yet the one we employ) is to use the sample covariance of the ensemble

$$B_i(Q) = \sqrt{\mathrm{cov}(Q_{[i]})}, \tag{8}$$

where the square root of a matrix is taken in the Cholesky sense. Note that $\mathrm{div}(B_i(Q)^{\mathrm{T}}) \equiv 0$, simplifying the Metropolization of the scheme. In order for $B_i(Q)$ to be positive definite, we need at least D linearly independent walker positions, which at minimum requires that $L > D$.

With the choice of B_i in (8), the ensemble scheme applied to the density

$$\bar{\pi}_{A,v}(Q, P) = \prod \hat{\pi}_{A,v}(q_i, p_i) = \prod \hat{\pi}(Aq_i + v, p), \tag{9}$$

for some invertible matrix A and vector v, generates a sequence of vectors $(q_1^{(n)}, \ldots, q_L^{(n)}, p_1^{(n)}, \ldots, p_L^{(n)})$ with the property that the transformed sequence $(Aq_1^{(n)} + v, \ldots, Aq_L^{(n)} + v, p_1^{(n)}, \ldots, p_L^{(n)})$ has exactly the same distribution as the sequence generated by the ensemble scheme applied to $\bar{\pi}$ (see "Appendix 5"). Just as choosing B as the square root of global covariance of π in (6) yields an affine invariant scheme, choosing the B_i as the square root of the ensemble covariance yields an affine invariant ensemble scheme. This affine invariance property suggests that ensemble schemes with B_i chosen as in (8) should perform well when the covariance of π has a large condition number. A related choice in the context of an overdamped formulation appears in Greengard (2015) and is shown to be affine invariant. An ensemble version of the HMC scheme using a mass matrix inspired by the BFGS optimization scheme appears in Zhang and Sutton (2011) though the relationship between that mass matrix and an approximation of the Hessian of $\log(\pi)$ or its inverse seems unclear because the method does not evaluate the derivative of $\log(\pi)$ at nearby points.

Using (8) in our ensemble schemes is problematic for several reasons. For high-dimensional problems, the requirement that $L > D$ may render the memory demands of the methods prohibitive. This problem can be easily remedied by only approximating and rescaling in the space spanned by the eigenvectors corresponding to the largest eigenvalues of the ensemble covariance matrix. While such a scheme can be implemented in a reasonably efficient manner, we find that simply blending the sample covariance matrix with the identity via the choice

$$B_i(Q) = \sqrt{I_D + \eta\, \mathrm{cov}(Q_{[i]})}, \tag{10}$$

for some fixed parameter $\eta \geq 0$ is just as effective and much simpler. There are several other ways to combine the identity and sample covariance matrix (e.g. a convex combination),

but our choice in (10) means that we do not need to additionally scale the stepsize with η, as for modest η the slowest motions of the system are not dramatically altered. The combination with the identity allows $L \leq D$ but destroys affine invariance. On the other hand as demonstrated in Sect. (4), the method is still capable of dramatically alleviating scaling issues.

Having resolved the rank deficiency issue by moving to the choice of B_i in (10), one difficulty remains. As described in the previous section, for many problems we might expect that the global covariance of π is reasonably well scaled but that the sampling problem is still poorly scaled (the Hessian of $-\log\pi$ has large condition number in highly probable regions of the sample space). To address problems of this type, we define a localized covariance matrix that better approximates the Hessian at a point q_i while retaining full rank. We weight samples in the covariance matrix based on their distance (scaled by the global covariance) to a walker's current position, i.e. we use

$$B_i(Q) = \sqrt{I_D + \eta\,\mathrm{wcov}(Q_{[i]}, \omega_\lambda(Q_{[i]}, q_i))}, \tag{11}$$

for parameters $\eta, \lambda > 0$, where now $\mathrm{wcov}(x, w)$ is a weighted covariance matrix of $K < L$ samples $q \in \mathbb{R}^{K\times D}$ with potentially unnormalized weights $w \in \mathbb{R}^K_+$:

$$(\mathrm{wcov}(q, w))_{ij} = \sum_{k=1}^{K} \frac{w_k}{W}(q_{k,i} - \bar{q}_i)(q_{k,j} - \bar{q}_j),$$

$$\bar{q}_i = \sum_{k=1}^{K} \frac{w_k}{W} q_{k,i}$$

with $W = \sum_k w_k$ and

$$(\omega_\lambda(Q, q))_j = \exp\left(-\frac{\lambda}{2}\|Q_j - q\|^2\right).$$

Note that using $Q_{[i]}$ and not Q in (11) is essential for preserving the validity of the scheme. Choosing $\lambda = 0$ reduces (11) to (10), whereas a large value of λ gives more refined estimation of the local scaling properties of the system. The divergence term in (7) can be computed explicitly by computing partial derivatives of $B_i(q)$, making use of the formula for the derivative of the square root of a matrix: $\partial_i M(x) = M\Phi(M^{-1}(\partial_i(MM^T))M^{-T})$, where $\Phi(M) = \mathrm{lower}(M) + \mathrm{diag}(M)/2$. Note that the matrices $B_i B_i^{\mathrm{T}}$ for B_i in (11) are sums of the identity and L rank one matrices so that all manipulations involving B_i can be accomplished in linear cost in the dimension D. In "Appendix 2", we detail a Metropolis–Hastings step that can be implemented (if needed) to correct any introduced bias. Because our ensemble scheme preserves π exactly when δt is small, one can also use the scheme absent of any Metropolis–Hastings step, improving the prospects for it to scale to very high dimension. Omission of the Metropolis–Hastings step for Langevin type methods is common practice in molecular dynamics MCMC simulations (see Leimkuhler and Matthews 2015) and has been considered in the context of computational statistics in (Dalalyan 2016; Durmus and Moulines 2016; Welling and Teh 2011).

We write out explicit pseudocode for the scheme in Algorithm 1. We consider dividing L walkers into G groups, where walker w is in group number $g(w)$. We also choose the number of steps to take between parallel communication, $T \leq N$, and initialize the momentum vector for each walker w so $p_w \sim N(0, I_D)$.

Algorithm 1 Ensemble Quasi-Newton

```
n = 0
while n < N do
    // Loop over each group in serial
    for i from 1 to G do
        // Loop over each group's walkers in parallel
        parfor w from 1 to L such that g(w) = i do
            // We take T steps using information from the
            // walkers in the other groups Q_[w]
            for t from 1 to T do
                (q_w, p_w) ← step(q_w, p_w; Q_[w])
                // The step function takes one step of
                // the discretization (see (7) or Appendix 2)
            end for
        end parfor
        // Communicate the changes to the overall state Q
        Broadcast new positions of the walkers in group i
    end for
    // Move time forward by T steps
    n ← n + T
end while
```

Typically it is most efficient to choose the size of each group to be a multiple of the number of available cores, in order to make the **parfor** loop efficient. The *step* function uses one new evaluation of the force $\nabla\log(\pi)$ each time it is called, as well as a new evaluation of the B matrix and its derivative. We can minimize parallel communication by setting T large to infrequently broadcast the new walker data.

4 Numerical tests

We consider two numerical experiments to demonstrate the potential improvements that this method offers. A python package with example implementations of the code is available at Matthews (2016).

4.1 Gaussian mixture model

We use the model presented in Chopin et al. (2012) which involves fitting the distribution of a dataset y to a univariate

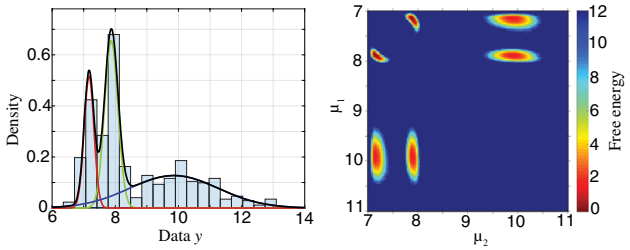

Fig. 2 We plot a maximum likelihood state (*left*) with the three component densities coloured as *red*, *green* and *blue*, with their sum in *black*, along with the original stamp data y as a histogram. The six modes due to label switching can be seen when looking at the log posterior plot (*right*) in μ_1 and μ_2. (Color figure online)

mixture model as the sum of three Gaussian distributions. The state vector is described by the means, precisions and weights of the three Gaussian distributions, denoted μ_i, λ_i and z_i, respectively. Due to the sum of the weights equalling unity, this gives us eight variables describing the mixture model. We also include a hyperparameter β that describes the rate parameter in the prior distribution on the precisions, giving $D = 9$ for the state overall. A full description of the problem is available in "Appendix 3".

We consider the Hidalgo stamps benchmark dataset, studied in (Izenman and Sommer 1988), as the data y with 485 datapoints. This example is well suited to the local covariance approach we present above, due to the invariance of the likelihood under a permutation of components (the label-switching problem). Thus, the system admits sets of $3! = 6$ equivalent modes, see Fig. 2, each with a local scaling matrix that has the same eigenvalues with permuted eigenvectors.

Though strictly speaking the problem is multimodal, the high barriers between modes make hopping between the basins extremely unlikely (we did not observe any switching in any simulations). Thus, this problem effectively tests the exploration rate within one well, with the symmetry between the modes guaranteeing the same challenges in each basin. The walkers may initialize in the neighbourhood of different local modes so that using a "global" preconditioning strategy would be sub-optimal. The best preconditioning matrix for the current position of a walker depends on which mode is closest to the walker. Instead, we use the covariance information from proximal walkers as in (11) to determine the optimal scaling.

We test the EQN scheme against the standard HMC scheme and a Metropolized version of Langevin dynamics. We used $L = 64$ walkers for the each scheme and compare the computed integrated autocorrelation times for an ensemble mean of quantities that vary slowly, shown in Table 1. The autocorrelation times are computed using the ACOR package (Goodman 2009).

We consider all three methods as equivalent in cost, as they require the same number of evaluations of $\nabla \log(\pi)$ per step, and scale similarly with the size of the data vector y. Comparing the slowest motions of the system, the EQN scheme is about 100 times more efficient compared to Langevin dynamics and 350 times more efficient than HMC. We found that removing the divergence term in the EQN scheme had no significant impact on the results.

Table 1 Computed autocorrelation times for slow variables, with the variable with the slowest motion marked in bold for each method

Scheme	min(z)	max(λ)	min(μ)	β
HMC	21495	**42935**	27452	7148
Langevin dynamics	6825	**13279**	8384	4641
Ensemble Q-N	69	83	98	**115**

4.2 Log Gaussian Cox model

To illustrate the method in a high-dimensional setting, we compare results for inference in the Log Gaussian Cox point process as in (Christensen et al. 2005). We aim to infer the latent variable field X from given observation data Y.

We make use of the RMHMC Matlab code template in our experiments (Girolami and Calderhead 2011b). In the model, we discretize the unit square into a 32×32 grid, with the observed intensity in each cell denoted $Y_{i,j}$ and Gaussian field $X_{i,j}$. We use two hyperparameters σ^2 and β to govern the priors, making the dimensionality of the problem $D = 32^2 + 2 = 1026$ dimensions. Full details of the model are provided in "Appendix 4".

As the evaluation of the derivative of the likelihood is significantly cheaper with respect to the latent x variables (tests showed computing the hyperparameter's derivatives to be about one hundred times slower), we employ a partial resampling strategy to first sample the latent variables using multiple steps and then perform one iteration for the hyperparameter distribution.

We generate synthetic test data Y, plotted in Fig. 3, and compare the HMC and Langevin dynamics schemes to EQN

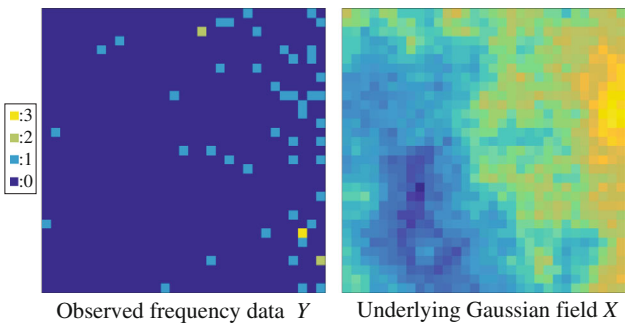

Fig. 3 The synthetic observed intensity Y (*left*) and the true Gaussian field X (*right*)

Table 2 Maximum autocorrelation times for each variable using each scheme

Scheme	x	σ^2	β	Efficiency
HMC	800.7	1041.6	1318.7	1.0
RMHMC	2158.9	34.0	1502.0	0.15
LD	405.1	140.6	435.3	3.5
$\cdots$ (no Metropolis)	81.6	20.5	136.**5**	11.2
EQN	71.9	49.2	239.5	5.4
$\cdots$ (no Metropolis)	64.4	8.8	47.8	26.8

(using 160 walkers) and the RMHMC scheme (Girolami and Calderhead 2011a). We additionally compare the results using the Langevin dynamics and EQN scheme without Metropolization, as the dynamics themselves sample π, and the Metropolis step only serves to remove discretization error (which is dominated by the sampling error in this example). RMHMC uses Hessian information to obtain scaling data for the distribution. This gives it a significant increase in cost, but improves the rate at which the sampler decorrelates. For the model, the RMHMC scheme requires approximately 2.2s per step, whereas the other schemes require approximately 0.35s per step.

In Table 2, we give the integrated autocorrelation times for ensemble averages of the hyperparameters β and σ^2, along with the autocorrelation time for the slowest component of the x variables. The efficiency is also shown, calculated as the wall time required per step divided by the autocorrelation time for the slowest hyperparameter (then normalized with respect to the HMC result). The slowest hyperparameter is compared instead of the slowest component of x because evolving the x dynamics requires less computation, hence it is trivial to reduce the autocorrelation time of x without significantly impacting the wall time.

In the results, the EQN scheme significantly outperforms the other methods, with the slowest motion of the system (the β hyperparameter) decorrelating more rapidly than the HMC or Langevin schemes for approximately the same cost. The RMHMC scheme's requires significant extra computation, making it much less efficient than the standard HMC scheme in this example.

5 Conclusion

We have presented a sampling algorithm that utilizes information from an ensemble of walkers to make more efficient moves through space, by discretizing a continuous ergodic quasi-Newton dynamics sampling the target distribution $\pi(x)$. The information from the other walkers can be introduced in several ways, and we give two examples using either local or global covariance information. The two forms of the B_i preconditioning matrix are then tested on benchmark test cases, where we see significant improvement compared to standard schemes.

The EQN scheme is cheap to implement, requiring no extra evaluations of $\nabla \log \pi(x)$ compared to schemes like MALA, and needing no higher derivative or memory terms. The scheme is also easily parallelizable, with communication between walkers being required infrequently. The dynamics (6) is novel in their approach to the introduction of the scaling information, and we build on previous work using walkers running in parallel to provide a cheap alternative to Hessian data.

The full capabilities of the EQN method, in the context of complex data science challenges, remain to be explored. It is likely that more sophisticated choices of B_i are merited for particular types of applications. The propagation of an ensemble of walkers also suggests natural extensions of the method to sensitivity analysis and to estimation of the sampling error in the MCMC scheme. Also left to be explored is the estimation of the convergence rate as a function of the number of walkers, which may be possible for simplified model problems.

Acknowledgements The authors would like to thank Aaron Dinner, Andrew Duncan and Mark Girolami for many useful discussions and comments. We also would like to thank the anonymous referees for many helpful suggestions.

Appendix 1: Autocorrelation times for poorly conditioned problems

For comparisons of efficiency in MCMC methods, the integrated correlation time (IAT) τ is often used as a rough measure of the rate of convergence of the chain, giving the approximate time between independent samples. For a function of interest $f(x)$ with mean zero, its IAT τ_f is

$$\tau_f = 1 + \frac{2}{\mathrm{var}[f(x)]} \sum_{n=1}^{\infty} \mathrm{cov}[f(x^{(n)}), f(x^{(0)})],$$

where E_{x_0} denotes the expectation with respect to the initial conditions $x^{(0)}$. We shall consider sampling the distribution $\pi(x) \propto \exp(-x^T M^{-1} x/2)$ for some symmetric positive definite matrix M, in order to investigate the efficiency of the scheme (2)

$$x^{(n+1)} = x^{(n)} + \delta t \, \nabla \log(\pi(x^{(n)})) + \sqrt{2\delta t}\, \mathrm{R}^{(n)}.$$

As in Section VIII of Goodman and Sokal (1989), we will use the worst-case IAT for functions of form $f(x) = v^T x$.

Suppose that $\lambda_{\max}$ is the largest eigenvalue of M^{-1} with corresponding eigenvector $v_{\max}$. Then if we choose $x^{(0)} = v_{\max}$

$$\mathrm{E}[x^{(n)}] = (1 - \delta t \lambda_{\max})^n v_{\max},$$

and hence we must choose δt such that $|1 - \delta t \lambda_{\max}| < 1$ to ensure convergence to the invariant mean vector zero. This gives the linear stability condition $\delta t < 2/\lambda_{\max}$.

We would expect that using a larger value of δt would lead to a more rapid decorrelation between samples. We can quantify this by computing the covariance between samples, where

$$\begin{aligned}\mathrm{cov}[x^{(n)}, x^{(0)}] &= (I - \delta t M^{-1})\mathrm{cov}[x^{(n-1)}, x^{(0)}] \\ &= (I - \delta t M^{-1})^n M\end{aligned}$$

from the definition of the update scheme. Similarly for a test function $f(x) = v^T x$, we have

$$\mathrm{cov}[f(x^{(n)}), f(x^{(0)})] = v^T (I - \delta t M^{-1})^n M v,$$

with integrated autocorrelation time

$$\tau_f = 1 + \frac{2}{v^T M v} v^T \left(\sum_{n=1}^{\infty} (I - \delta t M^{-1})^n \right) M v.$$

Assuming the linear stability condition is satisfied, we can rewrite this expression as

$$\tau_f = 1 + \frac{2}{\delta t\, v^T M v} v^T (M - \delta t I) M v = \frac{2 v^T M^2 v}{\delta t\, v^T M v} - 1. \quad (12)$$

Plugging in $v = v_{\min}$, the eigenvector corresponding to the minimum eigenvalue of M^{-1} (with eigenvalue $\lambda_{\min}$) we find that for the particular observable $f_*(x) = v_{\min}^T x$

$$\tau_{f_*} = \frac{2}{\delta t\, \lambda_{\min}} - 1,$$

and given the constraint $\delta t < 2/\lambda_{\max}$ this gives

$$\tau_{f_*} > \frac{\lambda_{\max}}{\lambda_{\min}} - 1,$$

so that even choosing the largest timestep possible, the rate of convergence will be slow whenever M has a wide range of eigenvalues.

Appendix 2: Metropolization of discretized scheme

In order to improve stability of the scheme, or to correct for numerical bias, we may seek to impose a Metropolis condition on the discretization of the dynamics (6). The discretization we use is given in (7), which we rewrite here for clarity:

$$p^{(n+1/4)} = p^{(n)} + \frac{\delta t}{2} F(q^{(n)}), \quad (13a)$$

$$q^{(n+1/2)} = q^{(n)} + \frac{\delta t}{2} B\left(q^{(n+1/2)}\right) p^{(n+1/4)} \quad (13b)$$

$$p^{(n+2/4)} = p^{(n+1/4)} + \frac{\delta t}{2} \mathrm{div}\left(B\left(q^{(n+1/2)}\right)^{\mathrm{T}}\right) \quad (13c)$$

$$\hat{p}^{(n+2/4)} = \alpha p^{(n+2/4)} + \sqrt{1-\alpha^2}\mathrm{R}^{(n)} \quad (13d)$$

$$p^{(n+3/4)} = \hat{p}^{(n+2/4)} + \frac{\delta t}{2} \mathrm{div}\left(B\left(q^{(n+1/2)}\right)^{\mathrm{T}}\right) \quad (13e)$$

$$q^{(n+1)} = q^{(n+1/2)} + \frac{\delta t}{2} B\left(q^{(n+1/2)}\right) p^{(n+3/4)} \quad (13f)$$

$$p^{(n+1)} = p^{(n+3/4)} + \frac{\delta t}{2} F(q^{(n+1)}), \quad (13g)$$

with $\alpha = \exp(-\gamma \delta t)$, $\mathrm{R}^{(n)} \sim N(0, I)$ and $F(q) = B(q)^{\mathrm{T}} \nabla \log \pi(q)$. Note that the step in (13b) must be solved implicitly, likely requiring many evaluations of the matrix B. However, as this requires no communication between walkers and no evaluations of $\nabla \log \pi(q)$, we consider this a "cheap" operation.

The ratio of transition probabilities necessary for the acceptance rule is

$$\begin{aligned}&\frac{T((q^{(n)}, p^{(n)}) \to (q^{(n+1)}, p^{(n+1)}))}{T((q^{(n+1)}, -p^{(n+1)}) \to (q^{(n)}, -p^{(n)}))} \\ &= \frac{f_\alpha(\mathrm{R}^{(n)})}{f_\alpha(\alpha \hat{p}^{(n+2/4)} - p^{(n+2/4)})} \frac{|I + \frac{1}{2}\delta t \nabla \otimes B(q^{(n+1/2)}) p^{(n+3/4)}|}{|I - \frac{1}{2}\delta t \nabla \otimes B(q^{(n+1/2)}) p^{(n+1/4)}|},\end{aligned}$$

with derivatives taken with respect to q and where

$$f_\alpha(x) = \exp\left(-\frac{\|x\|^2}{2(1-\alpha^2)}\right).$$

In an efficient implementation, the calculation of the derivatives of $B(q)$ (needed for the transition probabilities and the divergence term) is only required once per step, between (13b) and (13c), while the evaluation of the $F(q)$ term is also once per step (after initialization) between lines (13f) and (13g). A single step can then be Metropolized using

- Set $(q^*, p^*) \leftarrow Upd(q^{(n)}, p^{(n)})$
- Draw $u \sim U(0, 1)$
- Compute

$$U = \min\left(1, \frac{\hat{\pi}(q^*, p^*)}{\hat{\pi}(q^{(n)}, p^{(n)})} \frac{T((q^{(n)}, p^{(n)}) \to (q^*, p^*))}{T((q^*, -p^*) \to (q^{(n)}, -p^{(n)}))}\right)$$

– Then if $u < U$ we accept the move and set $(q^{(n+1)}, p^{(n+1)}) \leftarrow (q^*, p^*)$, otherwise we reject the move and flip the sign of the momentum, so set $(q^{(n+1)}, p^{(n+1)}) \leftarrow (q^{(n)}, -p^{(n)})$.

where the *Upd* function corresponds to a step of the discretization in (13). Similarly we can perform multiple steps and then accept/reject the trajectory by multiplying the associated transition probabilities. An example implementation in Python is available at Matthews (2016). For Metropolization of overdamped schemes such as (3), see Bou-Rabee et al. (2014).

It is important to note that in Algorithm 1, though we evolve all walkers inside a group in parallel, we iterate over the groups in serial. This is because, by construction, the preconditioning matrix for walker i in group g is a function of the positions of other walkers not in group g, which are fixed while we evolve walkers in group g. Thus, we preserve the Markov property by making sure that no two walkers who require each other's information are evolved simultaneously. The approach in Algorithm 1 reduces to partial resampling over the groups, ensuring we sample the product distribution $\bar{\pi}$.

Appendix 3: Details of the Gaussian mixture experiment

In this section, we provide the details of the Gaussian mixture experiment for fitting the Hidalgo stamp data y to the density

$$\rho(x \mid \theta) = \sum_{k=1}^{3} z_k N(x \mid \mu_k,\ \lambda_k^{-1}),$$

where μ_k and λ_k are the centre and precision of the component Gaussians, respectively, with weights $z_k > 0$ such that $\sum z_k = 1$. Let θ be the parameter/hyperparameter vector for this model. The target distribution for θ is $\pi(\theta) \propto p(\theta)\rho(y \mid \theta)$.

We use the prior distribution $p(\theta)$ such that for $k \in \{1, 2, 3\}$

$$\mu_k \sim N(m, \kappa^{-1}), \qquad \lambda_k \sim \text{Gamma}(\alpha, \beta),$$
$$(z_1, z_2, z_3) \sim \text{Dirichlet}_3(1, 1, 1),$$

with hyperparameter $\beta \sim \text{Gamma}(g, h)$ and constants $m = \text{mean}(y)$, $r = \text{range}(y)$, $\kappa = 4/r^2$, $\alpha = 2$, $g = 0.2$, $h = 100g/(\alpha r^2)$.

We compare the standard HMC scheme, with a Metropolized Langevin dynamics scheme and the EQN scheme presented in the paper. For each of the schemes, we tweak the stepsize until the acceptance is on average about $75 - -80\%$. The HMC and Langevin schemes are run by taking 50 steps per single iteration, and using a Metropolis step on the obtained trajectory, while the EQN scheme takes 5 steps per iteration. The Langevin and EQN scheme used a friction of $\gamma = 0.01$.

All schemes used 64 walkers, which amounts to 64 independent runs for the HMC and Langevin schemes. The EQN run used four groups of 16 walkers with the localized form of the covariance matrix, with $\eta = 100$ and $\lambda = 12$ in (11), however with the weighting kernel only using the Euclidean distance in the μ_i space rather than the entirety of θ. We observed that increasing η further reduced the acceptance probability significantly.

We verified that the autocorrelation function was well resolved given the number of samples that we had computed, in order that the computed IAT made sense. The experiment was run on an Intel Xeon Processor E5-2670 using 16 threads in Python, utilizing the MPI4PY package. This gives efficient parallelization in the EQN experiment, as each walker per group is mapped to one thread (the Langevin and HMC runs are already embarrassingly parallel).

Appendix 4: Details of the log Gaussian Cox experiment

We run a larger experiment with $1024+2$ total parameters to estimate. In the model, we discretize a unit square into a 32×32 grid, with the observed intensity in each cell denoted $Y_{i,j}$ and Gaussian field $X_{i,j}$.

The intensities are assumed to be conditionally independent and Poisson distributed with means $m\Lambda(i, j) = m \exp(X_{i,j})$ for latent intensity process $\Lambda(i, j)$ and $m = 1/32^2$. $X = \{X_{i,j}\}$ is a Gaussian process, where $x = \text{vec}(X)$ we have its mean $E(x) = m$ and covariance matrix

$$\Sigma_{(i,j),(i',j')} = \sigma^2 \exp(\delta(i, i', j, j')/32\beta),$$
$$\delta(i, i', j, j') = \sqrt{(i - i')^2 + (j - j')^2},$$

with parameters m, σ^2 and β.

Our goal is to sample likelihoods for the latent variables X but also sample values for the hyperparameters β and σ^2, whose priors we assume are exponentially distributed.

We generate synthetic data Y from a field X, created using $\beta = 1/33, \sigma^2 = 1.91$ and $m = \log(126) - \sigma^2/2$. We aim to infer X from our synthetic Y, along with the hyperparameters used for the model, using the HMC, RMHMC, Langevin dynamics and ensemble quasi-Newton methods. We make use of the RMHMC template code for the problem, available at http://www.ucl.ac.uk/statistics/research/rmhmc.

In order to run multiple highly-resolved simulations, we use a 32×32 grid rather than the 64×64 grid used in the original version of the problem. This reduces the dimension

of the latent variables from 4096 to 1024, which we still consider large enough for a significant test of the samplers' abilities, but this reduction allows us to run for longer and recover more accurate statistical information. However, the change in the model requires us to alter some parameters used in the RMHMC method, for example the timestep, in order to recover good efficiency. The timestep is increased until we reach 75% acceptance (though we do not claim our choice is optimal).

We implement all of the schemes in MATLAB, with each walker running on a single thread. For all methods, one iteration uses 50 latent variable steps for each one hyperparameter step. The ergodic property of the Langevin-dynamics-based schemes allows us to run without Metropolization, if we are willing to endure some discretization bias that is not removed with further sampling. We argue that in most practical cases, when using a sensible discretization parameter sampling error should always dominate the discretization bias.

For the ensemble quasi-Newton sampler, we run using 160 walkers using the global covariance formulation for B_i (effectively $\lambda = 0$). We use a partial resampling approach to sample the latent variables and hyperparameters, with the schemes using a B_i for each partition of variables. For the hyperparameters, we used $\eta = 1$, and for the latent variables, we used $\eta = 7$. We use five groups of 32 walkers with the walkers within each group running in parallel for ten thousand steps before switching to the next group. The cost of this communication is negligible as it is done so infrequently.

We verified that the autocorrelation function was well resolved given the number of samples that we had computed, in order that the computed IAT made sense. All experiments were run on an Intel Xeon Processor E5-2670 using 16 threads and Matlab's parallel toolbox.

Appendix 5: Affine invariance

We can extend our notion of affine invariance to the underdamped case, where the system state is $x = (q, p)$, with target distribution $\hat{\pi}(x) \propto \pi(q)\exp(-\|p\|^2/2)$. We consider affine transformations exclusively of the form $\hat{\psi}(x) = (\psi(q), p) = (Aq + v, p)$, defining the transformed distribution

$$\hat{\pi}_{\hat{\psi}} \propto \pi_\psi(q)\exp(-\|p\|^2/2), \qquad \pi_\psi(q) \propto \pi(\psi(q)),$$

for any invertible matrix A and fixed vector v. We may apply the discretization (13) to the density $\hat{\pi}_\psi$, using some scaling matrix $B_{\pi_\psi}(q)$ that we shall choose later, with $B_{\pi_\psi}(q) = B_\pi(\psi(q))$. The first computation (13a) for this density is

$$p^{(n+1/4)} = p^{(n)} + \frac{\delta t}{2} B_{\pi_\psi}(q)^{\mathrm{T}} \nabla \log \pi_\psi(q^{(n)}),$$

which is equivalent to, when writing $y = \psi(q)$,

$$p^{(n+1/4)} = p^{(n)} + \frac{\delta t}{2} B_\pi(y^{(n)})^{\mathrm{T}} A^{\mathrm{T}} \nabla \log \pi(y^{(n)}). \tag{14}$$

whereas writing $q = A^{-1}(y - v)$ in line (13b) and multiplying by A we have

$$y^{(n+1/2)} = y^{(n)} + \frac{\delta t}{2} A B_\pi\left(y^{(n+1/2)}\right) p^{(n+1/4)}. \tag{15}$$

Finally, line (13c) becomes

$$p^{(n+2/4)} = p^{(n+1/4)} + \frac{\delta t}{2} g\left(y^{(n+1/2)}\right) \tag{16}$$

where $g(x) = \mathrm{div}(B_\pi(x)^{\mathrm{T}} A)$.

Line (13d) remains unchanged, given our choice of affine invariant map $\hat{\psi}$ does not affect the momentum p. Suppose now that we choose

$$B_{\pi_\psi}(q) B_{\pi_\psi}(q)^T = A^{-1} C(q) A^{-T},$$

for some symmetric positive definite matrix C, and thus $B_{\pi_\psi}(q) = A^{-1}\sqrt{C(q)}$. For this choice, the discretization steps in (14)–(16) become independent of the scale factor A, and hence we obtain a scale-independent sampler when we use (13). One such choice is to use the square root of the (constant) covariance matrix $\mathrm{cov}_{\pi_\psi}(q)$ as the $B(q)$ matrix, as

$$\begin{aligned} B_{\pi_\psi} B_{\pi_\psi}^T &= \mathrm{cov}_{\pi_\psi}(q) = \mathrm{cov}_\pi(\psi^{-1}(x)) \\ &= A^{-1}\mathrm{cov}_\pi(x) A^{-T}, \end{aligned} \tag{17}$$

satisfying the invariance property as the covariance is automatically symmetric positive definite. Alternatively, one may choose B to be the inverse square root of the Hessian matrix of $\log \pi_\psi$, which shares this property (subject to some conditions on the Hessian).

We have shown that the discretization using this choice of B is affine invariant only up to affine transformations in the q variables. However, we may consider dynamics that are invariant under a transformation

$$\tilde{\psi}(q, p) = (Aq + v, A^{-T} p),$$

which acts in both components, where as before A_i is any invertible matrix and v_i is a vector. The dynamics we consider sample the distribution $\tilde{\pi}(q, p)$ ergodically, where

$$\begin{aligned} &\tilde{\pi}(q, p) \propto \pi(q)\tilde{\varphi}(q, p), \\ &\tilde{\varphi}(q, p) = \exp(-p^T M_\pi^{-1}(q) p/2 - \log|M_\pi(q)|/2), \end{aligned}$$

so that the marginal distribution of $\tilde{\pi}(q, p)$ in the momenta is $\pi(q)$. The mass matrix $M_\pi^{-1}(q)$ is a symmetric positive definite matrix that may be dependent on position q. For the choice of

$$J = \begin{bmatrix} 0 & -I \\ I & 0 \end{bmatrix} \qquad S = \begin{bmatrix} 0 & 0 \\ 0 & \gamma I \end{bmatrix}$$

in (1) we obtain the standard underdamped Langevin dynamics, and applying it to $\tilde{\pi}(q, p)$ with $\gamma = 1$ gives

$$\begin{aligned} \mathrm{d}q &= M_\pi^{-1}(q)p\mathrm{d}t, \\ \mathrm{d}p &= \nabla \log \pi(q)\mathrm{d}t + \nabla \log \tilde{\varphi}(q, p)\mathrm{d}t \\ &\quad - p\mathrm{d}t + \sqrt{2M_\pi(q)}\mathrm{d}W_t, \end{aligned}$$

where the gradient ∇ is taken with respect to the position q. Now applying this dynamics to the transformed distribution instead

$$\tilde{\pi}_{\tilde{\psi}}(q, p) \propto \pi_\psi(q)\tilde{\varphi}_\psi(q, p) = \pi(Aq + v)\tilde{\varphi}(Aq + v, A^{-T}p)$$

we obtain

$$\mathrm{d}q = M_{\pi_\psi}^{-1}(q)p\,\mathrm{d}t, \tag{18}$$

$$\begin{aligned} \mathrm{d}p &= \nabla \log \pi_\psi(q)\mathrm{d}t - \tfrac{1}{2}\nabla \log |M_{\pi_\psi}(q)|\mathrm{d}t \\ &\quad - \tfrac{1}{2}\nabla(p^T M_{\pi_\psi}^{-1}(q)p)\mathrm{d}t - p\mathrm{d}t + \sqrt{2M_{\pi_\psi}(q)}\mathrm{d}W_t, \end{aligned} \tag{19}$$

where ∇ denotes gradient with respect to the position coordinates. Changing variables $r = A^{-T}p$ gives

$$\begin{aligned} \mathrm{d}q &= M_{\pi_\psi}^{-1}(q)A^T r\,\mathrm{d}t, \\ \mathrm{d}r &= A^{-T}\nabla \log \pi_\psi(q)\mathrm{d}t - \tfrac{1}{2}A^{-T}\nabla \log |M_{\pi_\psi}(q)|\mathrm{d}t \\ &\quad - \tfrac{1}{2}A^{-T}\nabla(r^T A M_{\pi_\psi}^{-1}(q)A^T r)\mathrm{d}t \\ &\quad - r\mathrm{d}t + A^{-T}\sqrt{2M_{\pi_\psi}(q)}\mathrm{d}W_t. \end{aligned}$$

Then writing $y = Aq + v$ we have $\pi_\psi(q) \propto \pi(y)$, and if we assume M is chosen such that

$$M_{\pi_\psi}^{-1}(q) = A^{-1}C(y)A^{-T} \tag{20}$$

for some symmetric positive definite matrix $C(q)$, then the dynamics become

$$\begin{aligned} \mathrm{d}y &= C(y)r\,\mathrm{d}t, \\ \mathrm{d}r &= \nabla \log \pi(y)\mathrm{d}t + \tfrac{1}{2}\nabla \log |C(y)|\mathrm{d}t \\ &\quad - \tfrac{1}{2}\nabla(r^T C(y)r)\mathrm{d}t - r\mathrm{d}t + \sqrt{2C(y)}\mathrm{d}W_t, \end{aligned}$$

eliminating the A scaling term in the dynamics and yielding affine invariance with respect to the transformations $\tilde{\psi}$.

Using the inverse covariance matrix as the mass is one such choice for the M matrix, as similar to (17) we have

$$M_{\pi_\psi} = \mathrm{cov}_{\pi_\psi}(q) = \mathrm{cov}_\pi(\psi^{-1}(x)) = A^{-1}\mathrm{cov}_\pi(x)A^{-T}$$

satisfying (20). The inverse Hessian is another such matrix with this property.

This result applies directly to the RMHMC scheme (Girolami and Calderhead 2011a) which uses a position dependent mass matrix; however, it periodically redraws the momentum and sets $\gamma = 0$. If the inverse Hessian is used (assuming it remains symmetric positive definite), or another matrix such that (20) holds, then the resulting dynamics will be affine invariant under transformations $\tilde{\psi}(q, p) = (Aq + v, A^{-T}p)$.

Computationally there is no obvious benefit to including a p-scaling term in the affine transformation, as sampling $\pi(q)$ is the ultimate goal of our sampling, and the inclusion of momentum is to increase efficiency. However, the usage of (18) may become practically inefficient in the case of ensemble sampling, as the joint distribution will be

$$\begin{aligned} \bar{\pi}(Q, P) = &\prod_{i=1}^{L} \pi(q_i) \\ &\exp(-p_i^T M_i^{-1}(Q)p/2 - \log |M_i(Q)|/2). \end{aligned}$$

In the target joint distribution, the walker positions are no longer independent. This complicates Metropolization of the scheme which will now require calculations involving all walkers when any one walker's position is changed. Additionally in each walker's dynamics (18), evaluating the $\nabla_{q_i}\bar{\pi}(Q, P)$ term requires computing derivatives of each of the L-many M_i matrices, causing a significant amount of additional computational overhead per step.

Appendix 6: Noisy estimation of the divergence term

The divergence of a positive definite matrix appears in many of the schemes we consider above; however, the derivative is sometimes prohibitively computationally expensive to obtain, or infeasible to compute analytically. A Metropolization condition can be enforced to recover the correct sampling without this term, but we may instead approximate the divergence of a matrix $M(x)$ using a random update. Formally,

for a small constant $\epsilon > 0$ and vector $R \in \mathbb{R}^D$ we have

$$\begin{aligned} Z &= [M(x + \epsilon R) - M(x)]R \\ &= \epsilon \sum_{i=1}^{D} R_i \partial_i M(x) R \\ &\quad + \epsilon^2 \sum_{i=1}^{D} \sum_{j=1}^{D} R_i R_j \partial_i \partial_j M(x) R + O(\epsilon^3). \end{aligned}$$

If $R \sim N(0, I)$, then taking the expectation of Z yields

$$(\mathrm{E}[Z])_i = \epsilon \sum_{j=1}^{D} \partial_i M_{i,j}(x) + O(\epsilon^3),$$

and so $\mathrm{E}[Z] = \epsilon \operatorname{div}(M(x)) + O(\epsilon^3)$. This gives a cheap "noisy" approximation of the divergence term that will introduce some small bias into the system (depending on the spectrum of $M(x)$).

References

Andrés Christen, J., Fox, C., et al.: A general purpose sampling algorithm for continuous distributions (the t-walk). Bayesian Anal. **5**(2), 263–281 (2010)

Bennett, C.H.: Mass tensor molecular dynamics. J. Comput. Phys. **19**(3), 267–279 (1975)

Bou-Rabee, N., Vanden-Eijnden, E.: Pathwise accuracy and ergodicity of metropolized integrators for SDEs. Commun. pure appl. math. **63**(5), 655–696 (2010)

Bou-Rabee, N., Donev, A., Vanden-Eijnden, E.: Metropolis integration schemes for self-adjoint diffusions. Multiscale Model. Simul. **12**(2), 781–831 (2014). doi:10.1137/130937470

Cappé, O., Guillin, A., Marin, J.M., Robert, C.P.: Population Monte Carlo. J. Comput. Gr. Stat. **13**(4), 907–929 (2004). doi:10.1198/106186004X12803

Chopin, N., Lelièvre, T., Stoltz, G.: Free energy methods for Bayesian inference: efficient exploration of univariate Gaussian mixture posteriors. Stat. Comput. **22**(4), 897–916 (2012)

Christensen, O.F., Roberts, G.O., Rosenthal, J.S.: Scaling limits for the transient phase of local Metropolis–Hastings algorithms. J. R. Stat. Soc. Ser. B (Stat. Methodol.) **67**(2), 253–268 (2005)

Dalalyan, A.S.: Theoretical guarantees for approximate sampling from smooth and log-concave densities. J. R. Stat. Soc. Ser. B (Stat. Methodol.) (2016). doi:10.1111/rssb.12183

Duane, S., Kennedy, A.D., Pendleton, B.J., Roweth, D.: Hybrid Monte Carlo. Phys. Lett. B **195**(2), 216–222 (1987). doi:10.1016/0370-2693(87)91197-X

Duncan, A.B., Lelièvre, T., Pavliotis, G.A.: Variance reduction using nonreversible Langevin samplers. J. Stat. Phys. **163**(3), 457–491 (2016). doi:10.1007/s10955-016-1491-2

Durmus, A., Moulines, E.: High-dimensional Bayesian inference via the Unadjusted Langevin Algorithm. https://hal.inria.fr/TELECOM-PARISTECH/hal-01304430v2 (2016)

Foreman-Mackey, D., Hogg, D.W., Lang, D., Goodman, J.: emcee: the MCMC hammer. Publ. Astron. Soc. Pac. **125**(925), 306 (2013)

Gilks, W.R., Roberts, G.O., George, E.I.: Adaptive direction sampling. J. R. Stat. Soc. Ser. D (Stat.) **43**(1), 179–189 (1994)

Girolami, M., Calderhead, B.: Riemann manifold Langevin and Hamiltonian Monte Carlo methods. J. R. Stat. Soc. Ser. B (Stat. Methodol.) **73**(2), 123–214 (2011a). doi:10.1111/j.1467-9868.2010.00765.x

Girolami, M., Calderhead, B.: Matlab code for the RMHMC scheme. http://www.ucl.ac.uk/statistics/research/rmhmc, (2011b). [Online; accessed 01-Dec-2015]

Goodman, J.: ACOR package. http://www.math.nyu.edu/faculty/goodman/software/, (2009). [Online; accessed 01-Dec-2015]

Goodman, J., Sokal, A.D.: Multigrid Monte-Carlo method-conceptual foundations. Phys. Rev. D **40**(6), 2035–2071 (1989)

Goodman, J., Weare, J.: Ensemble samplers with affine invariance. Commun. appl. math. comput. sci. **5**(1), 65–80 (2010)

Greengard, P.: An ensemblized Metropolized Langevin sampler. Master's thesis, NYU, (2015)

Haario, H., Saksman, E., Tamminen, J.: An adaptive Metropolis algorithm. Bernoulli **7**(2), 223–242 (2001)

Hairer, M., Weare, J.: Improved diffusion Monte Carlo. Commun. Pure Appl. Math. **67**, 1995–2021 (2014)

Hammersley, J.M., Morton, K.W.: Poor man's Monte Carlo. J. R. Stat. Soc. B **16**(1), 23–38 (1954)

Hwang, C.-R., Hwang-Ma, S.-Y., Sheu, S.-J.: Accelerating Gaussian diffusions. Ann. Appl. Probab. **3**(3), 897–913 (1993)

Hwang, C.-R., Hwang-Ma, S.-Y., Sheu, S.-J., et al.: Accelerating diffusions. Ann. Appl. Probab. **15**(2), 1433–1444 (2005)

Iba, Y.: Population Monte Carlo algorithms. Trans. Jpn. Soc. Artif. Intell. **16**(2), 279–286 (2001). doi:10.1527/tjsai.16.279

Izenman, A.J., Sommer, C.J.: Philatelic mixtures and multimodal densities. J. Am. Stat. assoc. **83**(404), 941–953 (1988)

Jasra, A., Stephens, D.A., Holmes, C.C.: On population-based simulation for static inference. Stat. Comput. **17**(3), 263–279 (2007). doi:10.1007/s11222-007-9028-9

Leimkuhler, B., Matthews, C.: Molecular Dynamics: With Deterministic and Stochastic Numerical Methods. Interdisciplinary Applied Mathematics. Springer International Publishing, New York (2015)

Leimkuhler, B., Matthews, C., Stoltz, G.: The computation of averages from equilibrium and nonequilibrium Langevin molecular dynamics. IMA J. Numer. Anal. (2015). doi:10.1093/imanum/dru056

Liu, J.: Monte Carlo Strategies in Scientific Computing. Springer, New York (2002)

Ma, Y.A, Chen, T., Fox, E.: A complete recipe for stochastic gradient MCMC. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems 28, pp. 2917–2925. Curran Associates, Inc., New York (2015)

Martin, J., Wilcox, L.C., Burstedde, C., Ghattas, O.: A stochastic Newton MCMC method for large-scale statistical inverse problems with application to seismic inversion. SIAM J. Sci. Comput. **34**(3), A1460–A1487 (2012)

Matthews, C.: Ensemble Quasi-Newton python package. http://bitbucket.org/c_matthews/ensembleqn, (2016). [Online; accessed 01-Jul-2016]

Milstein, G., Tretyakov, M.: Stochastic Numerics for Mathematical Physics. Springer, New York (2004)

Monnahan, C.C., Thorson, J.T., Branch, T.A.: Faster estimation of Bayesian models in ecology using Hamiltonian Monte Carlo. Methods Ecol. Evol. (2016). doi:10.1111/2041-210X.12681

Neal, R.M., et al.: MCMC using Hamiltonian dynamics. Handb. Markov Chain Monte Carlo **2**, 113–162 (2011)

Pavliotis, G.A.: Stochastic Processes and Applications: Diffusion Processes, the Fokker-Planck and Langevin Equations. Texts in Applied Mathematics. Springer, New York (2014)

Rey-Bellet, L., Spiliopoulos, K.: Irreversible Langevin samplers and variance reduction: a large deviations approach. Nonlinearity **28**(7), 2081 (2015)

Roberts, G.O., Rosenthal, J.S.: Coupling and ergodicity of adaptive Markov chain Monte Carlo algorithms. J. Appl. Prob. **44**(2), 458–475 (2007)

Roberts, G.O., Tweedie, R.L.: Geometric convergence and central limit theorems for multidimensional Hastings and Metropolis algorithms. Biometrika **83**(1), 95 (1996). doi:10.1093/biomet/83.1.95

Rosenbluth, M.N., Rosenbluth, A.W.: Monte Carlo calculation of the average extension of molecular chains. J. Chem. Phys. **23**(2), 356–359 (1955)

Rossky, P.J., Doll, J.D., Friedman, H.L.: Brownian dynamics as smart Monte Carlo simulation. J. Chem. Phys. **69**(10), 4628–4633 (1978). doi:10.1063/1.436415

Sun, W., Yuan, Y.X.: Optimization Theory and Methods: Nonlinear Programming. Springer Optimization and Its Applications. Springer, USA (2006)

ter Braak, C.J.F.: A Markov chain Monte Carlo version of the genetic algorithm differential evolution: easy Bayesian computing for real parameter spaces. Stat. Comput. **16**(3), 239–249 (2006)

Welling, M., Teh, Y.W.: Bayesian learning via stochastic gradient Langevin dynamics. In: Proceedings of the 28th International Conference on Machine Learning (ICML-11), 681–688 (2011)

Zhang, Y., Sutton, C.A.: Quasi-Newton methods for Markov chain Monte Carlo. In: Shawe-Taylor, J., Zemel, R.S., Bartlett, P.L., Pereira, F., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 24, pp. 2393–2401. Curran Associates, Inc., New York. http://papers.nips.cc/paper/4464-quasi-newton-methods-for-markovchain-monte-carlo.pdf (2011)