Pseudo-extended Markov chain Monte Carlo

Christopher Nemeth[1], Fredrik Lindsten[2], Maurizio Filippone[3] and James
Hensman[4]

[1]Department of Mathematics and Statistics, Lancaster University, U.K.
[2]Department of Information Technology, Uppsala University, Sweden.
[3]Department of Data Science, EURECOM, France.
[4]PROWLER.io, Cambridge, U.K.

August 18, 2017

Abstract

Sampling from the posterior distribution using Markov chain Monte Carlo (MCMC) methods can require an exhaustive number of iterations to fully explore the correct posterior. This is often the case when the posterior of interest is multi-modal, as the MCMC sampler can become trapped in a local mode for a large number of iterations. In this paper, we introduce the pseudo-extended MCMC method as an approach for improving the mixing of the MCMC sampler in complex posterior distributions. The pseudo-extended method augments the state-space of the posterior using pseudo-samples as auxiliary variables, where on the extended space, the MCMC sampler is able to easily move between the well-separated modes of the posterior.

We apply the pseudo-extended method within an Hamiltonian Monte Carlo sampler, and show that by using the No U-turn algorithm (Hoffman and Gelman, 2014), our proposed sampler is completely tuning free. We compare the pseudo-extended method against well-known tempered MCMC algorithms and show the advantages of the new sampler on a number of challenging examples from the statistics literature.

Keywords: Markov chain Monte Carlo; Hamiltonian Monte Carlo; multi-modal distribution; pseudo-extended method;

1 Introduction

Markov chain Monte Carlo (MCMC) methods (Brooks et al., 2011) are generally regarded as the gold standard approach for sampling from high-dimensional distributions. In particular, MCMC

arXiv:1708.05239v1 [stat.ME] 17 Aug 2017

algorithms have been extensively applied within the field of Bayesian statistics (Green et al., 2015) to sample from posterior distributions, when the posterior density can only be evaluated point-wise. Under mild conditions, it can be shown that asymptotically, the limiting distribution of the samples generated from the MCMC algorithm will converge to the posterior distribution of interest. While theoretically elegant, one of the main drawbacks of MCMC methods is that running the algorithm to stationarity can be prohibitively expensive if the posterior distribution is of a complex form, for example, contains multiple unknown modes. In these settings, the Markov chain can become stuck for long periods of time without fully exploring the posterior distribution. This has led to recent interest in variational methods (Blei et al., 2016), where rather than sampling from the posterior, a tractable approximation is found which minimises the Kullback-Leibler divergence between the posterior and the approximation. These methods can be significantly faster than MCMC algorithms (Blei and Jordan, 2006), with the drawback that it can be difficult to quantify the accuracy of the variational approximation.

In this paper we present the pseudo-extended Markov chain Monte Carlo method as an approach for augmenting the state-space of the original posterior distribution to allow the MCMC sampler to easily move between areas of high posterior density. The pseudo-extended method introduces *pseudo-samples* on the extended space to improve the mixing of the Markov chain. We show that samples generated using the pseudo-extended method admit the posterior of interest as the limiting distribution once weighted using a *post-hoc* correction step. The pseudo-extended method can be applied as an extensive of many popular MCMC algorithms (Roberts et al., 1997; Roberts and Rosenthal, 1998; Neal, 2010). In this paper we use the Hamiltonian Monte Carlo (HMC) algorithm (Neal, 2010) as our preferred MCMC sampler, with the extension of the NUTS algorithm (Hoffman and Gelman, 2014), as implemented in STAN (Carpenter et al., 2017), to automatically tune the HMC sampler. Therefore, aside from choosing the number of pseudo-samples, the pseudo-extended method is completely tuning free.

The focus of the paper is on sampling from multi-modal distributions where standard MCMC algorithms tend to struggle. A popular solution to this problem is to use tempered MCMC approaches (Geyer, 1991, 2011; Marinari and Parisi, 1992; Neal, 2001), where MCMC is applied to a posterior distribution that is raised to a some power (commonly known as the *temperature*), that has the effect of flattening out the modes. However, these methods can be difficult to tune and may be computationally expensive requiring multiple interacting Markov chains. We show that the pseudo-extended method is a strong competitor against these tempering methods, with the advantage that we do not require a pre-tuned temperature schedule. Finally, we show that our approach can be applied to sampling from complex distributions where the posterior need not be multi-modal, but may be challenging for standard MCMC samplers.

2 Markov chain Monte Carlo sampling

Let π be a target probability density on $\mathbb{R}^d$ with respect to Lebesgue measure, defined for all $\mathbf{x} \in \mathcal{X} := \mathbb{R}^d$ by

$$\pi(\mathbf{x}) := \frac{\gamma(\mathbf{x})}{Z} = \frac{\exp\{-\phi(\mathbf{x})\}}{Z}, \tag{1}$$

where $\phi : \mathcal{X} \to \mathbb{R}$ is a continuously differentiable function and Z is the normalising constant and is typically intractable. Throughout we will refer to $\pi(\mathbf{x})$ as the target density, however, in the Bayesian setting, this would be the posterior, where for data $\mathbf{y} \in \mathcal{Y}$, the likelihood is denoted as $p(\mathbf{y}|\mathbf{x})$ with parameters $\mathbf{x}$ assigned a prior density $\pi_0(\mathbf{x})$. The posterior density of the parameters given the data is derived from Bayes theorem to give $\pi(\mathbf{x}) = p(\mathbf{y}|\mathbf{x})\pi_0(\mathbf{x})/p(\mathbf{y})$, where the marginal likelihood $p(\mathbf{y})$ is the normalising constant Z which is typically not available analytically.

For many statistical models, and in particularly Bayesian models, the target distribution $\pi(\mathbf{x})$ must be approximated numerically. If the target can be evaluated point-wise and under some general conditions (see Meyn and Tweedie (2012) for details), samples can be drawn from $\pi(\mathbf{x})$ using Markov chain Monte Carlo simulation. The most popular MCMC algorithm is the Metropolis-Hastings algorithm, with many other algorithms derived as special cases. Samples $\mathbf{x}'$ are drawn from a proposal distribution with transition density $q(\mathbf{x}'|\mathbf{x})$ and accepted as samples from the target $\pi(\mathbf{x})$ with probability,

$$\min\left\{1, \frac{\pi(\mathbf{x}')q(\mathbf{x}|\mathbf{x}')}{\pi(\mathbf{x})q(\mathbf{x}'|\mathbf{x})}\right\}. \tag{2}$$

The Metropolis-Hastings algorithm generates an ergodic Markov chain $\mathbf{x}_1, \mathbf{x}_2, \ldots$ which admits $\pi(\mathbf{x})$ as the invariant probability density (Brooks et al., 2011). The choice of proposal $q(\mathbf{x}'|\mathbf{x})$ does not impact the validity of the algorithm, but has a strong effect on the efficiency of the MCMC sampler by affecting the autocorrelation of the Markov chain and the probability of accepting new samples $\mathbf{x}'$.

2.1 Sampling from multi-modal target distributions

Many target densities of interest are multi-modal, where modes of the target are separated by low density regions. Notable examples in Bayesian inference, include the posterior over model parameters in mixture models (McLachlan and Peel, 2000), deep neural networks (Neal, 1996), and differential equation models (Calderhead and Girolami, 2009). The Metropolis-Hastings algorithm is known to poorly explore multi-modal targets as the sampler can easily become trapped in a local mode for a large number of MCMC iterations. Many popular solutions to this problem can be categorised as annealed/population/tempered MCMC (see Jasra et al. (2007) for a full review). The main idea behind tempered MCMC is to sample from a sequence of

tempered targets of the form,

$$\pi_t(\mathbf{x}) \propto \exp\left\{-\beta_t \phi(\mathbf{x})\right\}, \qquad t = 1, \ldots, T,$$

where β_t is a tuning parameter referred to as the *temperature* that is associated with $\pi_t(\mathbf{x})$. A sequence of temperatures, commonly known as the *ladder*, is chosen prior to sampling, where $0 = \beta_0 < \beta_1 < \ldots < \beta_T = 1$ and $\pi_T(\mathbf{x}) \propto \pi(x)$ corresponds to the target density of interest. The intuition behind tempered MCMC is that when β_t is small, the modes of the target are flattened out making is easier for the MCMC sampler to traverse through the regions of low density separating the modes. One of the most popular tempering algorithms is parallel tempering (PT) (Geyer, 1991), where in parallel, T separate MCMC algorithms are run with each sampling from one of the tempered targets $\pi_t(\mathbf{x})$. Samples from neighbouring Markov chains are exchanged (i.e. sample from chain t exchanged with chain $t-1$ or $t+1$) using a Metropolis-Hasting step. These exchanges improve the convergence of the Markov chain to the target of interest $\pi(\mathbf{x})$, however, information from low β_t targets can be slow to traverse up to the target of interest. A serial version of this algorithm, known as simulated tempering (ST) was proposed by Marinari and Parisi (1992), where there is a single Markov chain and β acts as an augmented state that is updated by moving up and down the temperature ladder. An alternative importance sampling based approach introduced by Neal (2001) uses annealed importance sampling (AIS) to draw samples from a simple base distribution, and then via a sequence of intermediate transition densities, moves the samples along the temperature ladder giving a weighted sample from the target distribution. The advantage of AIS over tempered MCMC is that tempered transitions require twice the computational effort of AIS as a sequence of upward and downward transitions are required. Generally speaking, these tempered approaches can be very difficult to apply in practice often requiring extensive tuning. In the case of parallel tempering, the user needs to choose the number of parallel chains T, temperature schedule, step-size for each chain and the number of exchanges at each iteration.

2.2 Hamiltonian Monte Carlo

The efficiency of the MCMC sampler is largely determined by the choice of proposal mechanism $q(\mathbf{x}'|\mathbf{x})$, where ideally a proposal is chosen that yields a high Metropolis-Hastings acceptance rate, while minimising the autocorrelation of the resulting Markov chain. In recent years, the Hamiltonian Monte Carlo algorithm has been shown to satisfy many of the conditions for a desirable MCMC algorithm and is the preferred MCMC scheme used in this paper. We will give a brief overview of HMC and the interested reader is referred to Neal (2010) for a full and detailed review. The HMC algorithm differs from standard MCMC by considering the sampling problem as the exploration of a physical system, which has the benefit of avoiding the random walk behaviour of simpler MCMC algorithms such as the random-walk Metropolis Roberts et al. (1997). We can treat our parameters of interest $\mathbf{x} \in \mathcal{X} := \mathbb{R}^d$ as position variables, and

introduce artificial momentum variables $\boldsymbol{\rho} \in \mathbb{R}^d$ that are independent of $\mathbf{x}$. The Hamiltonian $H(\mathbf{x}, \boldsymbol{\rho})$, represents the total energy of the system as the combination of the potential $\phi(\mathbf{x})$ and kinetic $\frac{1}{2}\boldsymbol{\rho}^\top \mathbf{M}^{-1}\boldsymbol{\rho}$ energy

$$H(\mathbf{x}, \boldsymbol{\rho}) := \phi(\mathbf{x}) + \frac{1}{2}\boldsymbol{\rho}^\top \mathbf{M}^{-1}\boldsymbol{\rho},$$

where $\phi(\mathbf{x})$ is defined in (1) and $\mathbf{M}$ is a mass matrix and is often set to the identity matrix. The Hamiltonian now augments our target distribution so that we are now sampling $(\mathbf{x}, \boldsymbol{\rho})$ from the joint distribution $\pi(\mathbf{x}, \boldsymbol{\rho}) \propto \exp\{H(\mathbf{x}, \boldsymbol{\rho})\} = \pi(\mathbf{x})\mathcal{N}(\boldsymbol{\rho}|0, \mathbf{M})$, which admits the target as the marginal.

Assuming that $\pi(\mathbf{x})$ is continuously differentiable, we can can drive $(\mathbf{x}_0, \boldsymbol{\rho}_0)$ through fictitious time $\tau \geq 0$ using Hamilton's equations (see Neal (2010) for details)

$$dx = \mathbf{M}^{-1}\boldsymbol{\rho}d\tau, \qquad d\boldsymbol{\rho} = \nabla_\mathbf{x}\phi(\mathbf{x})d\tau, \qquad (3)$$

so that $H(\mathbf{x}_0, \boldsymbol{\rho}_0) = H(\mathbf{x}_\tau, \boldsymbol{\rho}_\tau)$. Aside from a few special cases, Hamilton's equations (3) cannot be calculated analytically but can be solved numerically using symplectic integrators. Several numerical integrators are available which preserve the volume and reversibility of the Hamiltonian system (Girolami and Calderhead, 2011), the most popular being the *leapfrog*, or Stormer-Verlet integrator. The leapfrog integrator takes L steps, each of size ϵ, on the Hamiltonian dynamics (3), with one step given as follows (with the full HMC algorithm given in Appendix A):

$$\begin{aligned}
\boldsymbol{\rho}_{\tau+\frac{\epsilon}{2}} &= \boldsymbol{\rho}_\tau + \frac{\epsilon}{2}\nabla\phi(\mathbf{x}_\tau) \\
\mathbf{x}_{\tau+\epsilon} &= \mathbf{x}_\tau + \epsilon\mathbf{M}^{-1}\boldsymbol{\rho}_{\tau+\frac{\epsilon}{2}} \\
\boldsymbol{\rho}_{\tau+\epsilon} &= \boldsymbol{\rho}_{\tau+\frac{\epsilon}{2}} + \frac{\epsilon}{2}\nabla\phi(\mathbf{x}_{\tau+\epsilon})
\end{aligned}$$

Using a discretisation means that the Hamiltonian no longer preserves the target π (Neal, 2010). It can, however, be used as proposal distribution q within the Metropolis-Hastings algorithm (2), where the acceptance step corrects for the discretisation error. This reduces the acceptance rate from one, in the ideal analytic case, to a rate that is usually much higher than the random-walk Metropolis and results in the acceptance of proposed samples which are far from the current position in the Markov chain. One of the disadvantages of HMC is that, like other MCMC algorithms, it can become trapped in local modes of the target. HMC relies on gradient information when generating new samples causing the algorithm to be mode seeking, where proposed samples will likely be selected in the direction of the gradient of the target, meaning that compared to the simpler random-walk Metropolis, it can take much longer for the algorithm to escape a local mode of the target.

As with many MCMC algorithms, the performance of the HMC algorithm is highly dependent on the tuning of the parameters (ϵ, L). One way to tune the algorithm is to optimise

the parameters such that the acceptance rate is approximately 65% (Beskos et al., 2013). Alternatively, the parameters could be automatically tuned using the No-U-turn sampler (NUTS) introduced by Hoffman and Gelman (2014), where the trajectory length L is tuned to avoid the sampler doubling back on itself and ϵ is estimated using dual averaging (Nesterov, 2009). This algorithm has been implemented in the popular STAN software package (Carpenter et al., 2017) for probabilistic programming. STAN has the additional feature that the derivatives required to use HMC are calculated using automatic differentiation making HMC user friendly and tuning free. All of the simulations in Section 4, including our proposed pseudo-extended algorithm (see Section 3), are implemented in STAN and fully reproducible[1].

3 The pseudo-extended method

We extend the state-space of the original target distribution (1) by introducing N *pseudo-samples*, $\mathbf{x}_{1:N} = \{\mathbf{x}_i\}_{i=1}^{N}$, where the extended-target distribution $\pi^N(\mathbf{x}_{1:N})$ is defined on $\mathcal{X}^N$. The pseudo-samples act as auxiliary variables, where for each $\mathbf{x}_i$, we have a proposal $q(\mathbf{x}_i) \propto \exp\{-\delta(\mathbf{x}_i)\}$ which follows the usual support assumptions for importance sampling. In a similar vain to the *pseudo-marginal MCMC* algorithm (Beaumont, 2003; Andrieu and Roberts, 2009) (see Section 3.1 for details), our extended-target, including the auxiliary variables, is now of the form,

$$\pi^N(\mathbf{x}_{1:N}) := \frac{1}{N} \sum_{i=1}^{N} \pi(\mathbf{x}_i) \prod_{j \neq i} q(\mathbf{x}_j) = \frac{1}{Z} \left\{ \frac{1}{N} \sum_{i=1}^{N} \frac{\gamma(\mathbf{x}_i)}{q(\mathbf{x}_i)} \right\} \times \prod_i q(\mathbf{x}_i), \tag{4}$$

where $\gamma(\cdot)$ and Z are defined in (1).

We can sample from the extended-target using the Hamiltonian Monte Carlo algorithm, where the Hamiltonian system is now,

$$H(\mathbf{x}_{1:N}, \boldsymbol{\rho}) \propto -\log\left[\sum_{i=1}^{N} \exp\{-\phi(\mathbf{x}_i) + \delta(\mathbf{x}_i)\}\right] + \sum_{i=1}^{N} \delta(\mathbf{x}_i) + \frac{1}{2}\boldsymbol{\rho}^\top \mathbf{M}^{-1}\boldsymbol{\rho},$$

and $\boldsymbol{\rho} \in \mathbb{R}^{d+N}$. Using the same leapfrog integrator technique as discussed in Section 2.2, we can sample from the extended Hamiltonian system using the HMC algorithm, where after marginalising out the momentum variables $\boldsymbol{\rho}$, our MCMC samples will be distributed according to the target $\pi(\mathbf{x}_{1:N})$. We make use of the Hamiltonian Monte Carlo algorithm because of its impressive mixing rates (Neal, 2010); however, it is important to note that alternative MCMC schemes, such as the random-walk Metropolis Roberts et al. (1997), Metropolis-adjusted Langevin algorithm Roberts and Rosenthal (1998), slice sampler (Neal, 2003) and other MCMC algorithms can also be applied.

In the case where $N = 1$, our extended-target (4) simplifies back to the original target

---

[1] Code available here https://github.com/chris-nemeth/pseudo-extended-mcmc-code

$\pi(\mathbf{x}) = \pi^N(\mathbf{x}_{1:N})$ (1). For $N > 1$, the marginal distribution of the ith pseudo-sample is a mixture between the target and the proposal

$$\pi(\mathbf{x}_i) = \frac{1}{N}\pi(\mathbf{x}_i) + \frac{N-1}{N}q(\mathbf{x}_i).$$

Samples from the original target of interest, $\pi(\mathbf{x})$ are given by using a *post-hoc* correction step, where the samples from the extended-target are weighted with weights proportional to $\gamma(\mathbf{x}_i)/q(\mathbf{x}_i)$, for $i = 1, \ldots, N$.

Proposition 3.1. *Let $\{\mathbf{x}_{1:N}^{(m)}\}_{m=1}^M$ be M samples from the extended-target π^N. Weighting each sample with weights proportional to $\gamma(\mathbf{x}_i)/q(\mathbf{x}_i)$, for $i = 1, \ldots, N$ gives samples from the invariant distribution, $\pi(\mathbf{x})$.*

Proof given in Appendix B.

Introducing N pseudo-samples has the effect of linearly increasing the state-space of the target. In general, we would prefer to perform MCMC on a lower dimensional space rather as increasing the size of the state-space as this leads to slower mixing of the Markov chain Roberts and Rosenthal (2004). This is less problematic for the HMC algorithm compared to the random-walk Metropolis as the step-size should be scaled as $d^{-1/4}$ for HMC compared to d^{-1} for the variance of the random-walk Metropolis proposal (Beskos et al., 2013), giving a computational complexity of $\mathcal{O}(d^{1/4})$ and $\mathcal{O}(d)$, respectively, to explore the target in stationarity. This indicates that in high-dimensional spaces, the HMC algorithm is able to make bigger moves and explore the target space faster.

3.1 Connections to pseudo-marginal MCMC

The pseudo-extended target (4) can be viewed as an extension of the pseudo-marginal target of Andrieu and Roberts (2009). In the pseudo-marginal setting, it is assumed that the target density is of the form $\pi(\mathbf{x}) = \int_{\mathcal{Z}} \pi(\mathbf{x}, \mathbf{z})d\mathbf{z}$, where $\mathbf{z} \in \mathcal{Z}$ are latent variables that cannot be integrated out analytically. Using importance sampling, an unbiased estimate of the target $\tilde{\pi}(\mathbf{x})$ is computed using Monte Carlo by sampling latent variables $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N$ from a proposal distribution with density $q(\mathbf{z})$ and then approximating the integral as

$$\tilde{\pi}(\mathbf{x}) := \frac{1}{N} \sum_{i=1}^N \frac{\pi(\mathbf{x}, \mathbf{z}_i)}{q(\mathbf{z}_i)}, \quad \text{where} \quad \mathbf{z}_i \sim q(\cdot).$$

The pseudo-marginal target is then of a similar form to the pseudo-extended target (4),

$$\tilde{\pi}^N(\mathbf{x}, \mathbf{z}) := \frac{1}{N} \sum_{i=1}^N \pi(\mathbf{x}, \mathbf{z}_i) \prod_{j \neq i} q(\mathbf{z}_j), \tag{5}$$

where the latent variables $\mathbf{z}$ are sampled using importance sampling, and the state of interest $\mathbf{x}$ (e.g. parameters) is sampled using MCMC moves. It can be shown that this is a valid MCMC scheme sampling from the joint space $(\mathbf{x}, \mathbf{z})$. The main difference between the pseudo-marginal and pseudo-extended methods is that under the pseudo-extended scheme the target $\pi(\mathbf{x})$ is of a more general form where there are no latent variables to integrate out. All of the variables on the extended space (i.e. $\mathbf{x}_{1:N}$) are updated through MCMC moves and so importance sampling is not required, meaning that we do not need to sample from q, but only to evaluate it point-wise (discussed further in Section 3.3).

The pseudo-marginal method is a powerful technique for sampling from intractable likelihoods. In the subsequent sections we show how the pseudo-extended method can be useful for sampling from complex target distributions where the landscape of the target is difficult for standard MCMC samplers to traverse within an non-exhaustive number of MCMC iterations. In particular, where the target distribution is multi-modal, we show that extending the state-space allows our MCMC sampler to more easily explore the modes of the target. This is illustrated visually in the following example.

3.2 One-dimensional illustration

Consider a bimodal target of the form (see Figure 1),

$$\pi(\mathbf{x}) \propto \mathcal{N}(-1, 0.1) + \mathcal{N}(1, 0.02).$$

If there are $N = 2$ pseudo-samples, the pseudo-extended target (4) simplifies to

$$\pi(\mathbf{x}_{1:2}) \propto \gamma(\mathbf{x}_1)q(\mathbf{x}_2) + \gamma(\mathbf{x}_2)q(\mathbf{x}_1),$$

where, for the sake of illustration, we choose $q(\mathbf{x}) = \mathcal{N}(0, 2)$.

Density plots for the original and pseudo-extended target are given in Figure 1. On the original target, the modes are separated by a region of low density and an MCMC sampler will therefore only pass between the modes with low probability, thus potentially requiring an exhaustive number of iterations. On the pseudo-extended target, the modes of the original target $\pi(\mathbf{x})$ are now connected on the extended space $\pi(\mathbf{x}_{1,2})$. The proposal q has the effect of increasing the density in the low probability regions of the target which separate the modes. A higher density between the modes means that the MCMC sampler can now traverse between the modes with higher probability than under the original target.

In Figure 2, density plots of the original target are overlaid with samples drawn from the original and pseudo-extended targets using the HMC algorithm (Section 2.2) with Hamiltonians 2.2 and 3, respectively. The NUTS (Hoffman and Gelman, 2014) sampler is used to automatically adapt the tuning parameters of the HMC algorithm and this implemented in the STAN (Carpenter et al., 2017) software package. After 10,000 iterations of the HMC sampler on the original
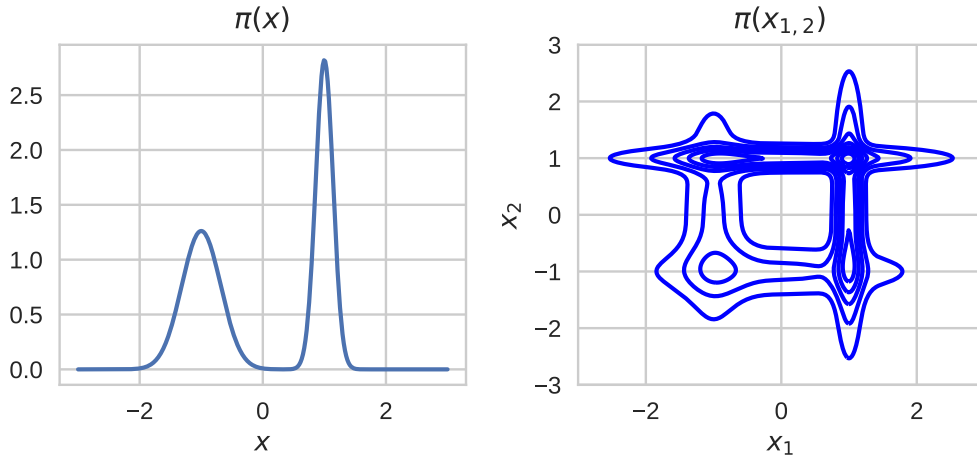
Figure 1: Left: Density plot of the target $\pi(\mathbf{x})$. Right: Contour plot of the extended target with $N = 2$.

target only one mode is discovered. Applying the same HMC algorithm on the pseudo-extended target, and then weighting the samples (as discussed in Section 3, both modes of the original target are discovered and the samples produce a good empirical approximation to the target. Tuning the HMC parameters manually, rather than automatically with the NUTS algorithm, could improve the mixing of the HMC algorithm on the original target. However, manually tuning the number of leapfrog steps and the stepsize parameter so as to sufficiently explore the target, would require *a priori* knowledge of the target structure (e.g. number of modes) which we assume is unavailable.
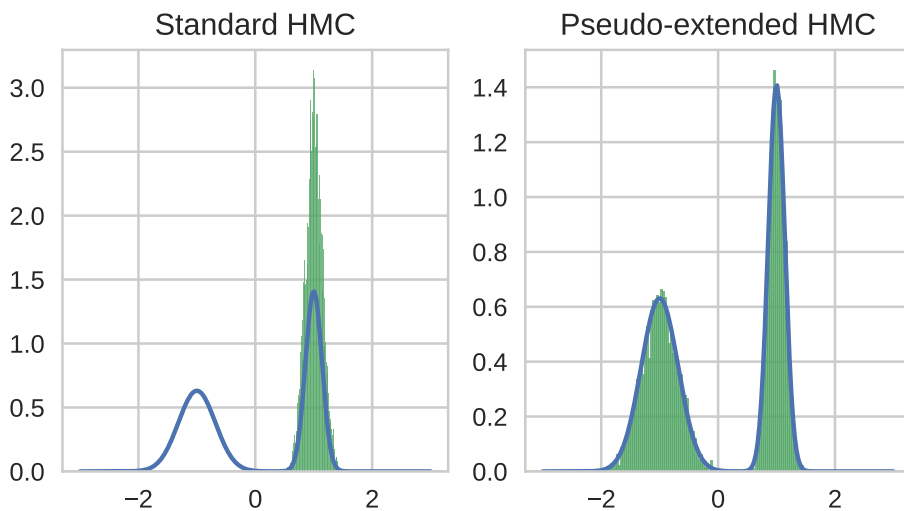

Figure 2: 10,000 samples from the original target (left) and extended target (right) using HMC sampler.

3.3 Tempered targets as proposals

Ideally, the optimal proposal $q(\mathbf{x})$ would be the target $\pi(\mathbf{x})$. As we are unable to sample from the target, we instead could aim to find a tractable proposal for q which sufficiently covers the support of the target and captures its multi-modality. Approximations, such as the Laplace approximation (Rue et al., 2009) and variational Bayes (Bishop, 2006, Chapter 10) tend to produce local approximations to one mode of the target. An alternative variational approach is *expectation-propagation* (EP) (Minka, 2001), which minimises the reverse Kullback-Leibler divergence used in traditional variational methods. Where variational Bayes tends to underestimate the uncertainty in the target distribution, EP has the effect of overestimating the uncertainty and provides a global approximation to the target encapsulating multiple modes. Expectation-propagation, however, can be limited in application requiring the target to be easily factorised.

Alternatively, rather than finding a closed-form approximation for the target, it is much simpler to find an approximation which can be evaluated point-wise. This is a simpler condition to satisfy and allows us to find better approximations to the target by using the target itself as the proposal distribution.

We assume that we have a family of approximating distributions,

$$
\Pi := \left\{ \pi_\beta(\mathbf{x}) = \frac{\gamma_\beta(\mathbf{x})}{z(\beta)} : \beta \in [0,1] \right\},
\tag{6}
$$

where $\gamma_\beta(\mathbf{x}) = \exp\{-\beta\phi(\mathbf{x})\}$ can be evaluated point-wise and $z(\beta)$ is typically intractable.

We will construct an extended target distribution $\pi^N(\mathbf{x}_{1:N}, \beta_{1:N})$ on $\mathcal{X}^N \times [0,1]^N$ consisting of N pseudo-pairs $(\mathbf{x}_i, \beta_i)$, for $i = 1, \ldots, N$. This target distribution will be constructed in such a way that the marginal distribution of each $\mathbf{x}$-sample is a mixture with components selected from Π. This will typically make the marginal distribution more diffuse than the target π itself, encouraging better mixing.

Let $q(\mathbf{x}, \beta) = \pi_\beta(\mathbf{x})q(\beta)$ where (for the sake of tractability) we let $q(\beta) = \frac{z(\beta)g(\beta)}{C}$, where $g(\beta)$ can be evaluated point-wise and C is a normalising constant. Thus,

$$
q(\mathbf{x}, \beta) = \frac{\gamma_\beta(\mathbf{x})g(\beta)}{C}.
\tag{7}
$$

The joint proposal $q(\mathbf{x}, \beta)$ does not admit a closed-form expression and in general it is not possible to sample from it. We can, however, use an MCMC sampler on the extended-target and therefore only require that $q(\mathbf{x}, \beta)$ can be evaluated point-wise. Under the proposal (6), the

pseudo-extended target (4) is now

$$\pi^N(\mathbf{x}_{1:N}, \beta_{1:N}) := \frac{1}{N} \sum_{i=1}^{N} \pi(\mathbf{x}_i)\pi(\beta_i) \prod_{j \neq i} q(\mathbf{x}_j, \beta_j) \tag{8}$$

$$= \frac{1}{ZC^{N-1}} \left\{ \frac{1}{N} \sum_{i=1}^{N} \frac{\gamma(\mathbf{x}_i)\pi(\beta_i)}{\gamma_{\beta_i}(\mathbf{x}_i)g(\beta_i)} \right\} \prod_{j=1}^{N} \gamma_{\beta_j}(\mathbf{x}_j)g(\beta_j),$$

where $\pi(\beta)$ is some arbitrary user-chosen target distribution for β. Through our choice of $q(\mathbf{x}, \beta)$, the normalising constants for the target and proposal, Z and C respectively are not dependent on $\mathbf{x}$ or β and so cancel in the Metropolis-Hastings ratio (2).

One of the main differences between our pseudo-extended method and the tempered MCMC algorithms of Section 2.1, is that the temperature parameter β is treated as unknown and is estimated as part of the MCMC scheme. This can be particularly beneficial as the temperature ladder does not need to be set prior to implementing the MCMC sampler, which itself can be difficult to tune *a priori*. Using a coarse grid of temperatures can cause the sampler to miss modes of the target, whereas a fine grid of temperatures leads to a significantly increased computational cost of running the sampler. An issue that is bypassed with pseudo-extended method.

4   Numerical illustrations

We test the performance of the pseudo-extended method on four different targets. The first two (Sections 4.1 and 4.2) are chosen to show how the pseudo-extended method performs when the target is multi-modal and we compare against popular competing algorithms from the literature. The final two targets (Section 4.3) are not multi-modal, but are interesting toy models which are challenging for MCMC algorithms, particularly HMC, to sample from.

For each of the simulation studies, we assume that $\pi(\beta) \propto 1$ and $g(\beta) \propto 1$, we also use a logit transformation for β to map the parameters onto the unconstrained space. All of the simulations for the pseudo-extended method use the tempered proposal (Section 3.3) for $q(\cdot)$ and thus the pseudo-extended target is given by (8). The HMC algorithm is used as the MCMC sampler of choice, and more specifically, is implemented within the STAN framework (Carpenter et al., 2017).

4.1 Mixture of Gaussians

We consider a popular example from the literature (Kou et al., 2006; Tak et al., 2016), where the target is a mixture of 20 bivariate Gaussians,

$$\pi(\mathbf{x}) = \sum_{j=1}^{20} \frac{w_j}{2\pi\sigma_j^2} \exp\left\{ \frac{-1}{2\sigma_j^2}(\mathbf{x} - \boldsymbol{\mu}_j)^\top(\mathbf{x} - \boldsymbol{\mu}_j) \right\}, \tag{9}$$

and $\{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_{20}\}$ are specified in Kou et al. (2006). We consider two scenarios, where (a) each mixture component has weight $w_j = 1/20$ and variance $\sigma_j^2 = 1/100$ and where (b) the weights $w_j = 1/||\boldsymbol{\mu}_j - (5,5)^\top||$ and variances $\sigma_j^2 = ||\boldsymbol{\mu}_j - (5,5)^\top||/20$ are unequal. Under scenario (a) the modes are well separated where most of the modes are more than 15 standard deviations apart. The distance between the modes makes this a challenging target distribution for standard MCMC algorithms which would require an exhaustive number of iterations to explore all of the modes (see Figure 3). In scenario (b) the mixture components close to (5,5) have a higher weight with smaller variance.
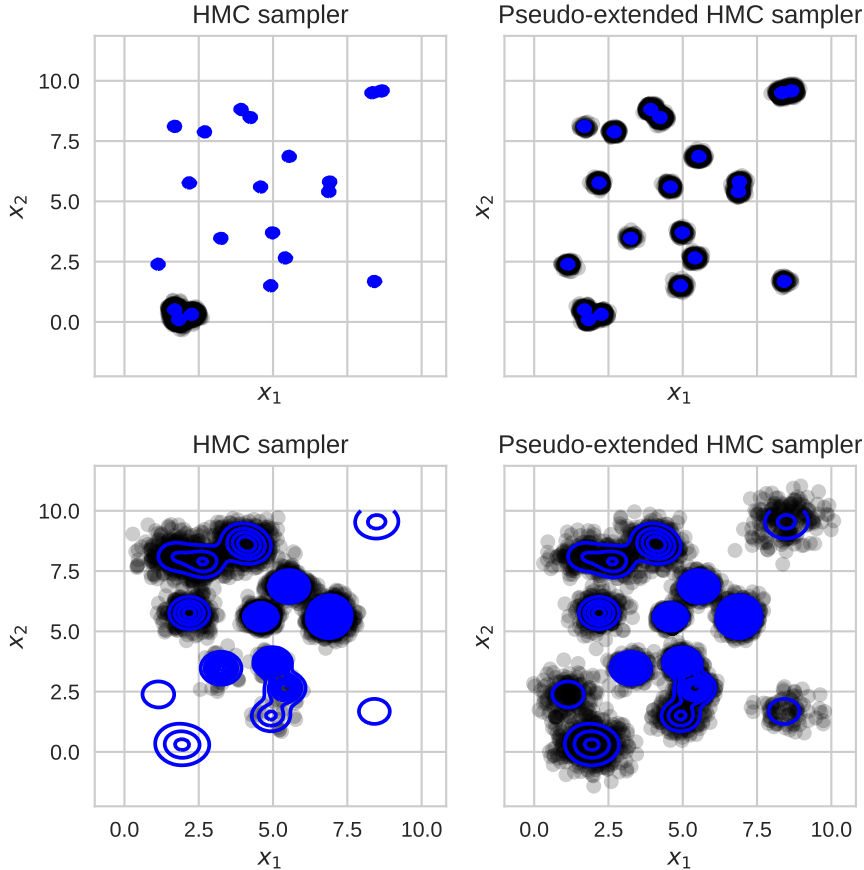

Figure 3: 10,000 samples drawn from the the target under scenario (a) (top) and scenario (b) (bottom) using the HMC sampler (left) and pseudo-extended method (right).

We ran the HMC and pseudo-extended HMC samplers under the same conditions as in Kou et al. (2006) and Tak et al. (2016). We initialise the samplers by drawing the initial values from $[0, 1]^2$ and run the samplers for 10,000 iterations. Figure 3 shows the samples drawn using the standard HMC and pseudo-extended HMC samplers overlayed with the contour plots for the two scenarios. As expected, in the case where the modes are well-separated (scenario (a)), the HMC sampler is only able to explore the modes locally clustered together, whereas using the pseudo-extended target with the proposal q as outlined in Section 3.3, the pseudo-extended HMC sampler is able to explore all of the modes. Under scenario (b), the unnormalised weights and variances of the mixture components is higher than under scenario (a), as a result, the density separating the modes is now larger making it easier for the HMC sampler to move between the mixture components. Compared to the pseudo-extended HMC sampler, the standard HMC sampler is still not able to explore all of the modes of the target.

We compare the pseudo-extended sampler against parallel tempering (PT) (Geyer, 1991), repelling-attracting Metropolis (RAM) (Tak et al., 2016) and the equi-energy (EE) MCMC sampler (Kou et al., 2006), all of which are designed for sampling from multi-modal targets. Table 1 gives the exact first and second moments of the target under scenarios (a) and (b) with the mean and standard deviation of each of the estimated moments for each sampler taken over 20 independent simulations. Each sampler was ran for 50,000 iterations (after burn-in) and the specific tuning details for the temperature ladder of PT and the energy rings for EE are given in Kou et al. (2006). Unlike the competing methods, the HMC and pseudo-extended (PE) HMC samplers are automatically tuned using the NUTS algorithm (Hoffman and Gelman, 2014) and simulations using the pseudo-extended target are implemented with $N = \{2, 5, 10, 20\}$ pseudo-samples. Furthermore, while not reported here, using the HMC sampler on the pseudo-extended target produces a Markov chain with significantly reduced autocorrelation compared to the EE and RAM samplers, which both rely on random walk updates.

The results of Table 1 show that all of the samplers, with the exception of HMC, provide accurate estimates of the first two moments of the target. Under scenario (a), the HMC sampler produces significantly biased estimates as a result of not exploring all of the modes of the target (see Figure 3), whereas under scenario (b), while still performing worse than the other samplers, the HMC estimates are significantly less biased as the sampler is able to explore the majority of modes of the target. The RAM, EE samplers perform equally well with PT showing the highest standard deviation of the moment estimates under both scenarios. Under some of the simulations PT did not explore all of the modes, and as discussed in Kou et al. (2006), parallel tempering has to be carefully tuned to avoid becoming trapped in local modes.

The pseudo-extended sampler with tempered proposal (Section 3.3) performs well in both scenarios. For the smallest number of pseudo-samples ($N = 2$), the pseudo-extended HMC sampler performs equally as well as the competing methods. Increasing the number of pseudo-samples leads to a decrease in the standard deviation of the moment estimates. However, increasing the number of pseudo-samples also increases the overall computational cost of the

	Scenario (a)			
	$\mathbb{E}[\mathbf{X}_1]$	$\mathbb{E}[\mathbf{X}_2]$	$\mathbb{E}[\mathbf{X}_1^2]$	$\mathbb{E}[\mathbf{X}_2^2]$
Truth	4.478	4.905	25.605	33.920
RAM	4.471 (0.091)	4.932 (0.101)	25.572 (0.900)	33.223 (1.100)
EE	4.502 (0.107)	4.944 (0.139)	25.924 (1.098)	34.476 (1.373)
PT	4.419 (0.170)	4.879 (0.283)	24.986 (1.713)	33.597 (2.867)
HMC	3.133 (2.857)	1.741 (0.796)	18.002 (27.201)	3.679 (2.226)
PE (N=2)	4.383 (0.042)	4.913 (0.101)	24.585 (0.434)	34.038 (1.000)
PE (N=5)	4.469 (0.039)	4.893 (0.049)	25.553 (0.374)	33.834 (0.437)
PE (N=10)	4.467 (0.027)	4.903 (0.027)	25.479 (0.251)	33.907 (0.234)
PE (N=20)	4.477 (0.017)	4.910 (0.020)	25.576 (0.145)	33.945 (0.213)

	Scenario (b)			
	$\mathbb{E}[\mathbf{x}_1]$	$\mathbb{E}[\mathbf{x}_2]$	$\mathbb{E}[\mathbf{x}_1^2]$	$\mathbb{E}[\mathbf{x}_2^2]$
Truth	4.688	5.030	25.558	31.378
RAM	4.673 (0.026)	5.029 (0.035)	25.508 (0.263)	31.456 (0.334)
EE	4.699 (0.072)	5.037 (0.086)	25.693 (0.739)	31.433 (0.839)
PT	4.709 (0.116)	5.001 (0.134)	25.813 (1.122)	31.105 (1.186)
HMC	4.460 (0.151)	4.843 (0.474)	22.811 (1.481)	28.691 (3.978)
PE (N=2)	4.666 (0.050)	5.054 (0.077)	25.572 (0.461)	31.922 (0.669)
PE (N=5)	4.675 (0.015)	5.033 (0.021)	25.629 (0.167)	31.663 (0.220)
PE (N=10)	4.670 (0.010)	5.026 (0.016)	25.598 (0.087)	31.646 (0.169)
PE (N=20)	4.674 (0.013)	5.023 (0.013)	25.628 (0.134)	31.577 (0.121)

Table 1: Moment estimates for two mixture scenarios calculated using the repelling-attractive Metropolis (Tak et al., 2016) algorithm, the equi-energy sampler (Kou et al., 2006), parallel tempering (Geyer, 1991), standard HMC (Neal, 2010) and the pseudo-extended HMC sampler. Results include the means and standard deviations (in brackets) calculated over 20 independent simulations.

pseudo-extended sampler. Figure 4 measures the cost of the pseudo-extended sampler as the average mean squared error (over 20 runs) multiplied by the computational time. From the figure we see that by minimising the error relative to computational cost, the optimal number of pseudo-samples, under both scenarios, is between 2 and 5. We also note that Figure 4 suggests that the number of pseudo-samples may be problem specific. In scenario (a), where the modes are well-separated, increasing the number of pseudo-samples beyond 5 does not significantly increase the cost of the sampler, whereas under scenario (b), using more than 5 pseudo-samples (where the mixture components are easier to explore) introduces a significant increase in the computational cost without a proportional reduction in the error.
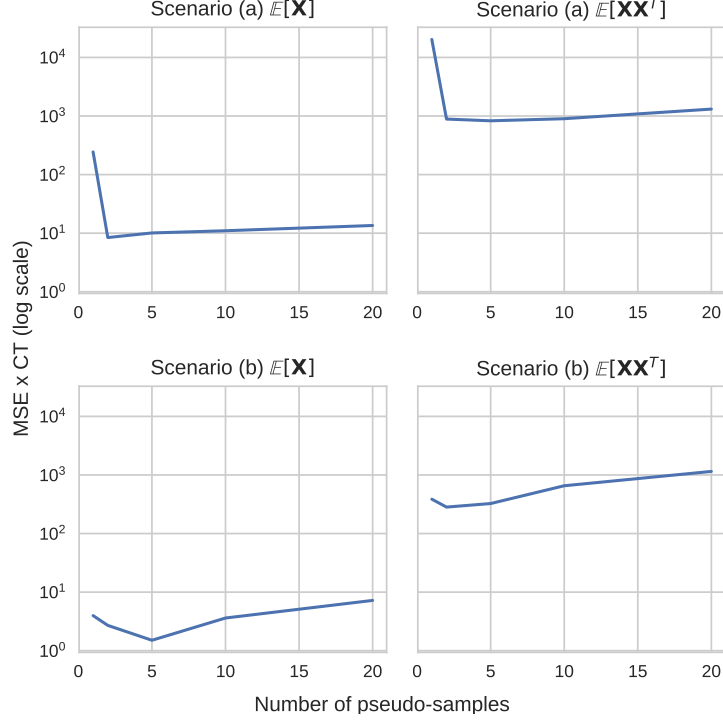
Figure 4: Average mean squared error (MSE) (given on the log scale) of the first and second moments taken over 20 independent simulations for varying number of pseudo-samples N, where MSE is scaled by computational time (CT) and plotted as MSE $\times$ CT.

4.2 Boltzmann machine relaxations

A challenging inference problem from the statistical physics literature is sampling from a Boltzmann machine distribution (Jordan et al., 1999), which is an inference problem equivalent to that of the Ising model. The Boltzmann machine distribution is defined on the binary space $\mathbf{s} \in \{-1, 1\}^{d_b} := \mathcal{S}$ with mass function

$$P(\mathbf{s}) = \frac{1}{Z_b} \exp\left\{\frac{1}{2}\mathbf{s}^\top \mathbf{W}\mathbf{s} + \mathbf{s}^\top \mathbf{b}\right\}, \quad \text{with} \quad Z_b = \sum_{\mathbf{s} \in \mathcal{S}} \exp\left\{\frac{1}{2}\mathbf{s}^\top \mathbf{W}\mathbf{s} + \mathbf{s}^\top \mathbf{b}\right\}, \quad (10)$$

where $\mathbf{W}$ is a $d_b \times d_b$ real symmetric matrix and $\mathbf{b} \in \mathbb{R}^{d_b}$ are the model parameters. Sampling from this distribution typically requires Gibbs steps (Geman and Geman, 1984) which tend to mix very poorly as the states can be strongly correlated when the Boltzmann machine has high levels of connectivity (Salakhutdinov, 2010).

Hamiltonian Monte Carlo methods have been shown to perform significantly better than Gibbs sampling when the states of the target distribution are highly correlated (Girolami and Calderhead, 2011). Unfortunately, general HMC algorithm is restricted to sampling from continuous rather than discrete state spaces. Using the *Gaussian integral trick* (Hertz et al., 1991), we can introduce auxiliary variables $\mathbf{x} \in \mathbb{R}^d$, where in the joint distribution $(\mathbf{x}, \mathbf{s})$ the

$\mathbf{s}^\top \mathbf{W}\mathbf{s}$ term from (10) cancels. From the joint distribution we can marginalise out the now uncorrelated and independent variables s_i to give the marginal density

$$\pi(\mathbf{x}) \propto \exp\left\{\frac{1}{2}\mathbf{x}^\top\mathbf{x} - \sum_{k=1}^{d_b} \log\cosh(\mathbf{q}_k^\top\mathbf{x} + b_k)\right\}, \tag{11}$$

which is commonly referred to as the *Boltzmann machine relaxation*. The vectors $\{\mathbf{q}_k^\top\}_{k=1}^{d_b}$ are rows of the $d_b \times d$ matrix $\mathbf{Q}$, where $\mathbf{Q}\mathbf{Q}^\top = \mathbf{W} + \mathbf{D}$ and $\mathbf{D}$ is a diagonal matrix chosen to ensure that $\mathbf{W} + \mathbf{D}$ is positive semi-definite. Full derivation details are given in Appendix C.

The moments of the Boltzmann relaxation distribution (11) correspond directly with the moments of the discrete Boltzmann machine distribution (10) (see Appendix C). Therefore, we can directly compare the moment estimates generated from our MCMC algorithm on the continuous space to the analytically tractable moments of the discrete Boltzmann machine distribution. While analytically tractable, the moments of the discrete Boltzmann machine distribution are calculated by enumerating over the 2^{d_b} discrete states, which will become computationally intractable for large d_b, as such, we restrict our experiments to $d_b = 28$ ($d = 27$) so as to compare our MCMC estimates with the true moments.

For our experiments we follow the set-up of Graham and Storkey (2017). The bias parameters are sampled as $\mathbf{b} \sim \mathcal{N}(0, 0.1^2)$ and the weight parameter $\mathbf{W}$ was generated by first sampling a $d_b \times d_b$ random orthogonal matrix $\mathbf{R}$ (Stewart, 1980) along with a vector of eigenvalues $\mathbf{e}$, where $e_i = \lambda_1 \tanh(\lambda_2\eta_i)$ and $\eta_i \sim \mathcal{N}(0, 1)$, for $i = 1, 2, \ldots, d_b$. The weight parameter is then set to $\mathbf{W} = \mathbf{R}\operatorname{diag}(\mathbf{e})\mathbf{R}^\top$, with the diagonal elements set to $\mathbf{W}_{i,i} = 0$, for $i = 1, 2, \ldots, d_b$. We set $\lambda_1 = 6$ and $\lambda_2 = 2$ which has been found empirically to produce highly multi-modal distributions (see Figure 5 for a two-dimensional projection).

Under the settings given above, we compare the HMC and pseudo-extended (PE) HMC algorithms against annealed importance sampling (AIS) and stimulated tempering (ST). We also compare against the recently proposed continuously tempered HMC algorithm of Graham and Storkey (2017) (GS), which is similar to the pseudo-extended algorithm in the sense that both approaches estimate the temperature parameter β on the continuous space. The continuously tempered HMC approach also relies on a variational approximation to the target, which plays a role similar to that of the proposal q in the pseudo-extended method.
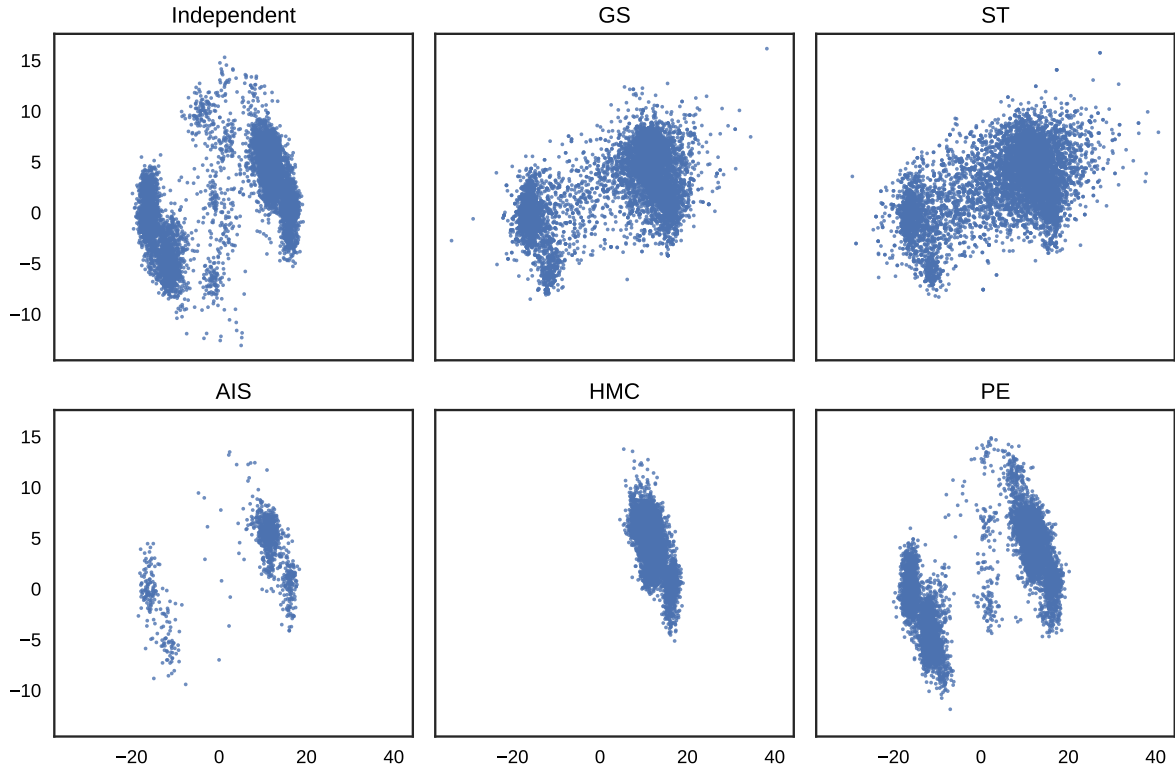
Figure 5: Two-dimensional projection of samples drawn from the target using each of the proposed methods, where the first plot gives the ground-truth sampled directly from the Boltzmann machine relaxation distribution.

All of the algorithms were hand tuned to achieve optimal performance with a temperature ladder of length 1,000 used for both simulated tempering and annealed importance sampling. The final 10,000 iterations for each algorithm were used to calculate the root mean squared errors of the estimates of the first two moments, taken over 10 independent runs, and are given Figure 6. The multi-modality of the target makes it difficult for the standard HMC algorithm to adequately explore the target, and as shown in Figure 5, the HMC algorithm is not able to traverse the modes of the target. The remaining algorithms perform reasonably well in approximating the first two moments of the distribution with some evidence supporting the improved performance of the pseudo-extended algorithm and simulated tempering approach.

In Figure 5, samples drawn using each algorithm are plotted alongside independent draws from the Boltzmann relaxation distribution (only computationally feasible for low to moderate dimensional settings). Most of the methods, with the exception of HMC, are able to capture the predominate modes of the target. The pseudo-extended method provides an accurate approximation of the target landscape, in particular the areas of low density.
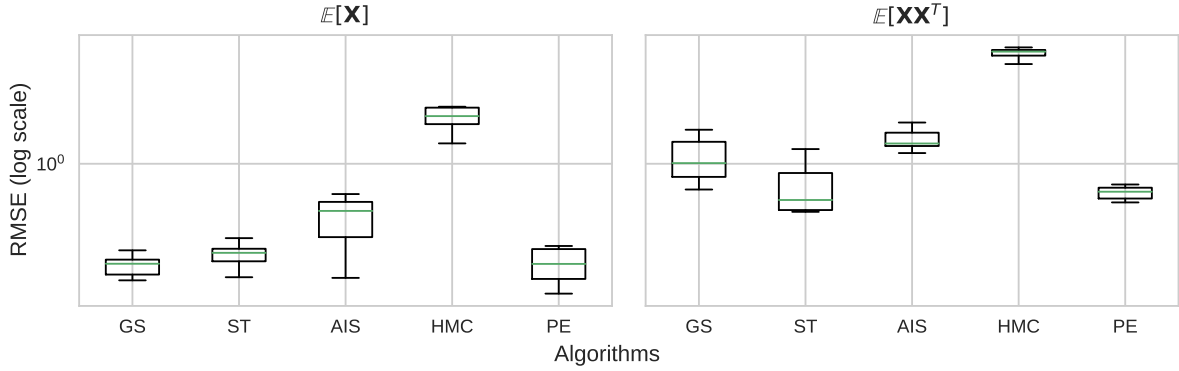
17

Figure 6: Root mean squared error (given on the log scale) of the first and second moment of the target taken over 10 independent simulations and calculated for each of the proposed methods.

As noted in the mixture of Gaussians example (Section 4.1), increasing the number of pseudo-samples improves the accuracy of the pseudo-extended method, but at a computational cost which grows linearly with N. When choosing the number of pseudo-samples it is sensible that N increases linearly with the dimension of the target. However, taking into account computational cost, a significantly smaller number of pseudo-samples can be used while still achieving a high level of sampling accuracy. In Figure 7, the average mean squared error (MSE) taken over 10 independent simulations is given for increasing N, where $N = 1$ corresponds with the HMC sampler. The MSE in Figure 7 is scaled by computational time to give a metric with which to compare the pseudo-extended algorithm for varying N. Without accounting for the computational cost, the lowest MSE would be attributed to the sampler with the largest N. Using this metric shows that, relative to computational cost, for the Boltzmann relaxation example using $N = 5$ pseudo-samples is appropriate, even though the state-space of the target is $d = 27$.
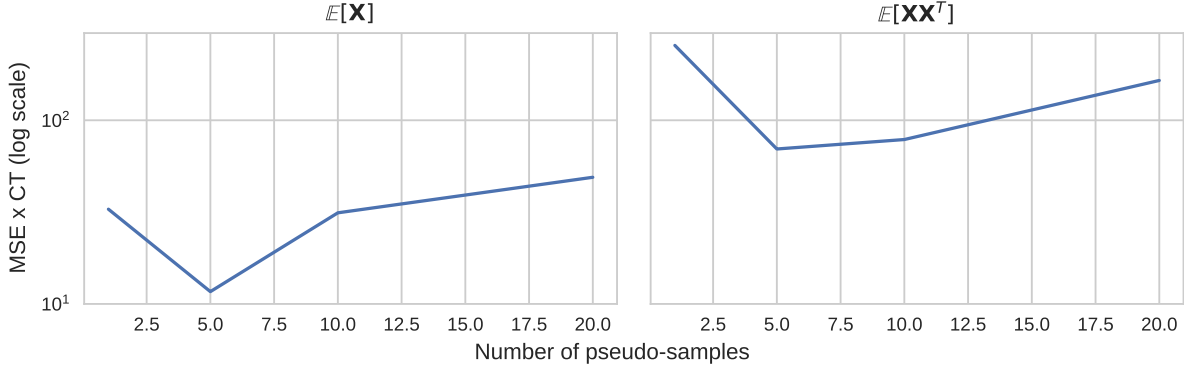
Figure 7: Average mean squared error (MSE) (given on the log scale) taken over 10 independent simulations with varying number of pseudo-samples N, where the MSE is scaled by computational time as MSE $\times$ CT

4.3 Challenging targets

The examples considered so far have focused on using the pseudo-extended framework to sample from multi-modal targets. In a more general sense, the pseudo-extended method can be applied to sample from target distributions which are challenging for standard MCMC samplers. We consider two illustrative examples to highlight the improvement of pseudo-extended HMC over standard HMC. The first is the warped Gaussian target, popularly known as banana-shaped target given in Haario et al. (1999)

$$\pi(\mathbf{x}) = \mathcal{N}(x_1|0, \nu)\mathcal{N}(x_2|b(x_1^2 - \nu), 1),$$

where we set $b = 0.1$ and $\nu = 100$ for our experiments. Our second example is the flower target from Sejdinovic et al. (2014),

$$\pi(\mathbf{x}) = \exp\left\{ -\frac{\sqrt{x_1^2 + x_2^2} - r - A\cos(\omega\text{arctan2}(x_2, x_1))}{2\sigma^2} \right\},$$

where the distribution represents a periodic perturbation with amplitude A and frequency ω. For our experiments we set $r = 10$, $A = 6$, $\omega = 6$ and $\sigma = 1$.

We use the NUTS (Hoffman and Gelman, 2014) tuning algorithm for HMC as implemented within STAN (Carpenter et al., 2017) for both standard HMC and pseudo-extended HMC, with $N = 2$ pseudo-samples and both algorithms ran for 10,000 iterations, with the first half of the chain removed as burn-in. Figure 8 shows the trace plots for both targets overlaid with a contour plot. For both targets the pseudo-extended model completely explores the target space, whereas the standard HMC algorithm struggles to fully explore the flower target within the fixed number of iterations and appears to under explore the tails of the banana target. The lines

given in the plot show the trajectory of the Markov chain, what is particularly notable is that the pseudo-extended method allows the HMC sampler to make large jumps across the target space which reduces the autocorrelation of the Markov chain.
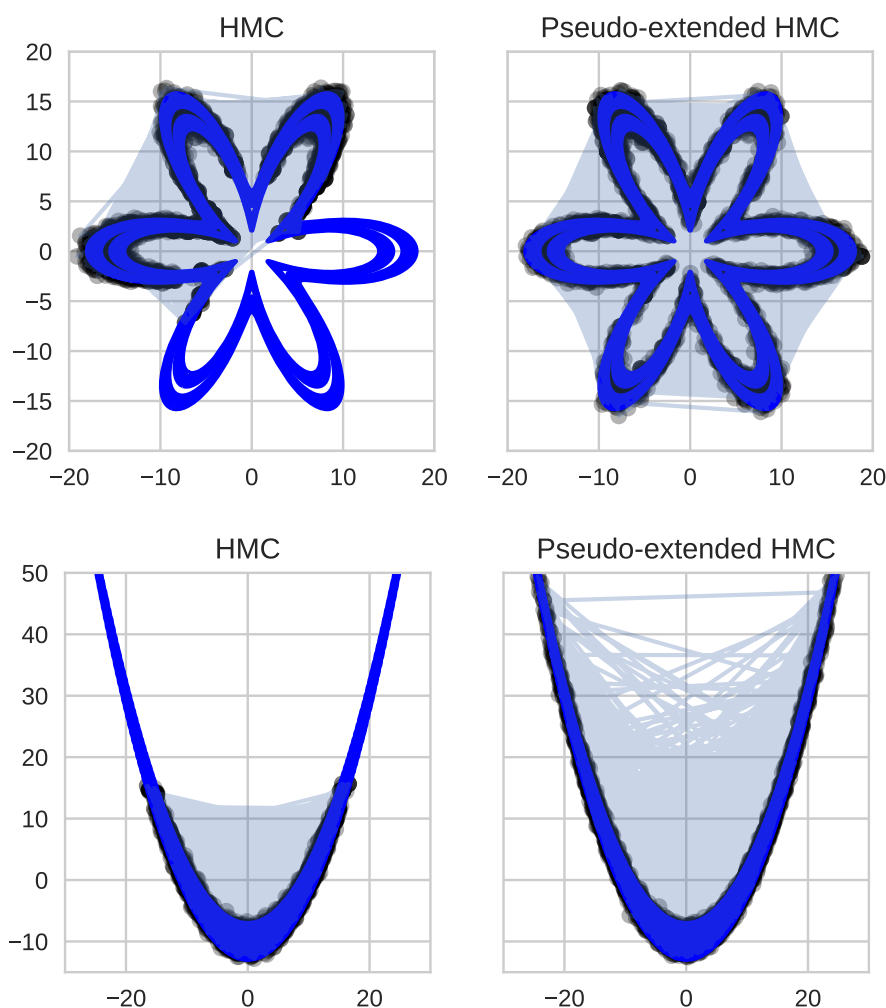

Figure 8: Flower target and Warped Gaussian

5   Discussion

We have introduced the pseudo-extended method as an approach for augmenting the target distribution for MCMC sampling. We have shown that the pseudo-extended method can be used to sample from distributions where the target density is of a complex form with possibly multiple separated modes that are challenging for standard MCMC algorithms to sample from. The pseudo-extended method can easily be applied within the Hamiltonian Monte Carlo algorithm leading to efficient MCMC sampling. Furthermore, using the NUTS sampler and STAN software package, the pseudo-extended HMC algorithm is completely tuning free, this is particularly beneficial when sampling from multi-modal targets as standard tempering MCMC methods often require extensive tuning.

We have suggested that a sensible proposal q for the pseudo-extended method is to use a tempered version of the target. However, there are many alternative proposals that could also be used, for example, a closed-form density or a mixture approximation. Further work investigating these alternatives is ongoing.

Acknowledgements

References

Andrieu, C. and Roberts, G. O. (2009). The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, 37(2):697–725. 6, 7

Beaumont, M. a. (2003). Estimation of population growth or decline in genetically monitored populations. *Genetics*, 164(3):1139–60. 6

Beskos, A., Pillai, N., Roberts, G., Sanz-Serna, J.-M., and Stuart, A. (2013). Optimal tuning of the hybrid Monte Carlo algorithm. *Bernoulli*, 19(5A):1501–1534. 6, 7

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer. 10

Blei, D. and Jordan, M. (2006). Variational Inference for {Dirichlet} Process Mixtures. *Bayesian Analysis*, 1(1):121–144. 2

Blei, D. M., Kucukelbir, A., and Mcauliffe, J. D. (2016). Variational Inference: A Review for Statisticians. pages 1–33. 2

Brooks, S., Gelman, A., Jones, G., and Meng, X.-L. (2011). *Handbook of markov chain monte carlo*. CRC press. 1, 3

Calderhead, B. and Girolami, M. (2009). Estimating Bayes factors via thermodynamic integration and population MCMC. *Computational Statistics and Data Analysis*, 53(12):4028–4045. 3

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., and Riddell, A. (2017). Stan: A probabilistic programming language. *Journal of Statistical Software*, 76(1):1–32. 2, 6, 8, 11, 19

Geman, S. and Geman, D. (1984). Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Trans Pattern Analysis and Machine Intelligence*, 6(6):721–741. 15

Geyer, C. J. (1991). Markov chain {Monte Carlo} maximum likelihood. In *Computing Science and Statistics: Proc. 23rd Symp. on the Interface*, pages 156–163. Fairfax. 2, 4, 13, 14

Geyer, C. J. (2011). Importance sampling, simulated tempering, and umbrella sampling. In Brooks, S., Gelman, A., Jones, G., and Meng, X.-L., editors, *Handbook of markov chain monte carlo*. CRC press. 2

Girolami, M. and Calderhead, B. (2011). Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214. 5, 15

Graham, M. M. and Storkey, A. J. (2017). Continuously tempered Hamiltonian Monte Carlo. In *Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence*, pages 1–12. 16, 25

Green, P. J., Latuszy, K., Pereyra, M., and Robert, C. P. (2015). Bayesian computation: a perspective on the current state, and sampling backwards and forwards. *arXiv preprint arXiv:1502.01148*, (Metropolis 1987). 2

Haario, H., Saksman, E., and Tamminen, J. (1999). Adaptive proposal distribution for random walk Metropolis algorithm. *Computational Statistics*, 14(3):375. 19

Hertz, J. A., Krogh, A. S., and Palmer, R. G. (1991). *Introduction to the theory of neural computation*, volume 1. Basic Books. 15, 25

Hoffman, M. and Gelman, A. (2014). The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(April):1593–1623. 1, 2, 6, 8, 13, 19

Jasra, A., Stephens, D. A., and Holmes, C. C. (2007). On population-based simulation for static inference. *Statistics and Computing*, 17(3):263–279. 3

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233. 15

Kou, S., Zhou, Q., and Wong, W. H. (2006). Equi-energy sampler with applications in statistical inference and statistical mechanics. *Annals of Statistics*, 34(4):1581–1619. 12, 13, 14

Marinari, E. and Parisi, G. (1992). Simulated Tempering : a New Monte Carlo Scheme. *Europhysics Letters*, 19(6):451–458. 2, 4

McLachlan, G. J. and Peel, D. (2000). *Finite mixture models*. Wiley Series in Probability and Statistics, New York. 3

Meyn, S. P. and Tweedie, R. L. (2012). *Markov chains and stochastic stability*. Springer Science & Business Media. 3

Minka, T. P. (2001). *A Family of Algorithms for Approximate Bayesian Inference*. PhD thesis, Cambridge, MA, USA. 10

Neal, R. M. (1996). *Bayesian Learning for Neural Networks*. Springer-Verlag New York, Inc., Secaucus, NJ, USA. 3

Neal, R. M. (2001). Annealed importance sampling. *Statistics and Computing*, 11:125–139. 2, 4

Neal, R. M. (2003). Slice sampling. *Annals of Statistics*, 31(3):743–748. 6

Neal, R. M. (2010). MCMC Using Hamiltonian Dynamics. In *Handbook of Markov Chain Monte Carlo (Chapman & Hall/CRC Handbooks of Modern Statistical Methods)*, pages 113–162. 2, 4, 5, 6, 14

Nesterov, Y. (2009). Primal-dual subgradient methods for convex problems. *Mathematical Programming*, 120(1):221–259. 6

Roberts, G. and Rosenthal, J. S. (2004). General state space Markov chains and MCMC algorithms. *Probability Surveys*, 1:20–71. 7

Roberts, G. O., Gelman, A., and Gilks, W. (1997). Weak Convergence and Optimal Scaling of the Random Walk Metropolis Algorithms. *The Annals of Applied Probability*, 7(1):110–120. 2, 4, 6

Roberts, G. O. and Rosenthal, J. S. (1998). Optimal scaling of discrete approximations to Langevin diffusions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(1):255–268. 2, 6

Rue, H., Martino, S., and Chopin, N. (2009). Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(2):319–392. 10

Salakhutdinov, R. (2010). Learning Deep Boltzmann Machines using Adaptive MCMC. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, volume 10, pages 943—-950. 15

Sejdinovic, D., Strathmann, H., Garcia, M. L., Andrieu, C., and Gretton, A. (2014). Kernel Adaptive Metropolis-Hastings. In *International Conference on Machine Learning*, volume 31, pages 1665—-1673. 19

Stewart, G. W. (1980). The Efficient Generation of Random Orthogonal Matrices with an Application to Condition Estimators. *SIAM Journal of Numerical Analysis*, 17(3):403–409. 16

Tak, H., Meng, X.-L., and van Dyk, D. A. (2016). A Repulsive-Attractive Metropolis Algorithm for Multimodality. pages 1–13. 12, 13, 14

Zhang, Y., Sutton, C., Storkey, A., and Ghahramani, Z. (2012). Continuous Relaxations for Discrete Hamiltonian Monte Carlo. In *Advances in Neural Information Processing Systems 25*, pages 3194–3202. 25, 26

A Hamiltonian Monte Carlo

Algorithm 1 Hamiltonian Monte Carlo

Input: Initial parameters $\mathbf{x}^{(0)}$, step-size ϵ and trajectory length L.

for $m = 1$ **to** M **do**

   Sample momentum $\boldsymbol{\rho} \sim \mathcal{N}(0, M)$

   Set $\mathbf{x}_1 \leftarrow \mathbf{x}^{t-1}$ and $\boldsymbol{\rho}_1 \leftarrow \boldsymbol{\rho}$

   for $l = 1$ **to** L **do**

      $\boldsymbol{\rho}_{l+\frac{1}{2}} \leftarrow \boldsymbol{\rho}_l + \frac{\epsilon}{2}\nabla \log \pi(\mathbf{x}_l)$

      $\mathbf{x}_{l+1} \leftarrow \mathbf{x}_l + \epsilon M^{-1}\boldsymbol{\rho}_{l+\frac{1}{2}}$

      $\boldsymbol{\rho}_{l+1} \leftarrow \boldsymbol{\rho}_{l+\frac{1}{2}} + \frac{\epsilon}{2}\nabla \log \pi(\mathbf{x}_{l+1})$

   end for

   Set $\mathbf{x} \leftarrow \mathbf{x}_{L+1}$ and $\boldsymbol{\rho} \leftarrow \boldsymbol{\rho}_{L+1}$

   With probability $\min\{1, \exp[H(\mathbf{x}^{m-1}, \boldsymbol{\rho}^{m-1}) - H(\mathbf{x}, \boldsymbol{\rho})]\}$

   set $\mathbf{x}^m \leftarrow \mathbf{x}$ and $\boldsymbol{\rho}^m \leftarrow \boldsymbol{\rho}$

end for

Output: Samples $\{\mathbf{x}^m\}_{m=1}^M$ from $\pi(\mathbf{x})$.

B Proof of Proposition 3.1

Consider the following generative model of the extended target:

- Sample I uniformly on $1 : N$,

- Draw $\mathbf{x}_I \sim \pi(\mathbf{x})$.

- For $j \neq I$, draw $\mathbf{x}_j \sim q(\mathbf{x})$.

Marginalising over the discrete variable I gives the extended target (4).

Let $b_i \in \{0, 1\}$ be a binary indicator variable for each pseudo-sample, $b_{1:N} = \{b_i\}_{i=1}^N$, where $\sum_i b_i = 1$. From the generative process, $b_i = 1$ indicates that $\mathbf{x}_i \sim \pi(\mathbf{x})$.

The joint target of $(\mathbf{x}_{1:N}, b)$ is

$$\pi^N(\mathbf{x}_{1:N}, b_{1:N}) = \frac{1}{Z}\sum_{i=1}^N b_i \left\{ \frac{\gamma(\mathbf{x}_i)}{q(\mathbf{x}_i)} \right\} \times \prod_{i=1}^N q(\mathbf{x}_i) \times \prod_{i=1}^N \left(\frac{1}{N} \right)^{b_i} \tag{12}$$

Marginalising over the binary variables gives $\sum_b \pi(\mathbf{x}_{1:N}, b_{1:N}) = \pi(\mathbf{x}_{1:N})$. Therefore, our original target of interest is, $\pi(\mathbf{x}) = \pi(\mathbf{x}_i | b_i = 1)$, where the conditional distribution on the binary variable is,

$$\pi(b_i = 1 | \mathbf{x}_{1:N}) = \frac{\pi(\mathbf{x}_{1:N} | b_i = 1)\pi(b_i = 1)}{\sum_{j=1}^{N} \pi(\mathbf{x}_{1:N} | b_j = 1)\pi(b_j = 1)} \propto \frac{\gamma(\mathbf{x}_i)}{q(\mathbf{x}_i)} \tag{13}$$

Drawing samples from the target $\pi(\mathbf{x}) = \pi(\mathbf{x}_i | b_i = 1)$ we first sample $\pi(\mathbf{x}_{1:N})$ using HMC, then compute the posterior for b conditioned on each sample, and then select the appropriate $\mathbf{x}_i$ by sampling from $\pi(b | \mathbf{x}_{1:N})$. $\qquad\square$

C   Boltzmann machine relaxation derivations

The Boltzmann machine distribution is defined on the binary space $\mathbf{s} \in \{-1, 1\}^{d_b} := \mathcal{S}$ with mass function

$$P(\mathbf{s}) = \frac{1}{Z_b} \exp\left\{\frac{1}{2}\mathbf{s}^\top \mathbf{W}\mathbf{s} + \mathbf{s}^\top \mathbf{b}\right\}, \qquad Z_b = \sum_{\mathbf{s} \in \mathcal{S}} \exp\left\{\frac{1}{2}\mathbf{s}^\top \mathbf{W}\mathbf{s} + \mathbf{s}^\top \mathbf{b}\right\}, \tag{14}$$

where $\mathbf{b} \in \mathbf{R}^{d_b}$ and $\mathbf{W}$ is a $d_b \times d_b$ real symmetric matrix are the model parameters.

Following the approach of Graham and Storkey (2017) and Zhang et al. (2012) we convert the problem of sampling on the 2^{d_b} discrete space to a continuous problem using the Gaussian integral trick (Hertz et al., 1991). We introduce the auxiliary variable $\mathbf{x} \in \mathbb{R}^d$ which follows a conditional Gaussian distribution,

$$\pi(\mathbf{x} | \mathbf{s}) = \frac{1}{(2\pi)^{d/2}} \exp\left\{-\frac{1}{2}(\mathbf{x} - \mathbf{Q}^\top \mathbf{s})^\top (\mathbf{x} - \mathbf{Q}^\top \mathbf{s})\right\}, \tag{15}$$

where $\mathbf{Q}$ is a $d_b \times d$ matrix such that $\mathbf{Q}\mathbf{Q}^\top = \mathbf{W} + \mathbf{D}$ and $\mathbf{D}$ is a diagonal matrix chosen to ensure that $\mathbf{W} + \mathbf{D}$ is a positive semi-definite matrix.

Combining (14) and (15) the joint distribution is,

$$\pi(\mathbf{x}, \mathbf{s}) = \frac{1}{(2\pi)^{d/2} Z_b} \exp\left\{-\frac{1}{2}\mathbf{x}^\top \mathbf{x} + \mathbf{s}^\top \mathbf{Q}\mathbf{x} - \frac{1}{2}\mathbf{s}^\top \mathbf{Q}\mathbf{Q}^\top \mathbf{s} + \frac{1}{2}\mathbf{s}^\top \mathbf{W}\mathbf{s} + \mathbf{s}^\top \mathbf{b}\right\} \tag{16}$$

$$= \frac{1}{(2\pi)^{d/2} Z_b} \exp\left\{-\frac{1}{2}\mathbf{x}^\top \mathbf{x} + \mathbf{s}^\top (\mathbf{Q}\mathbf{x} + \mathbf{b}) - \frac{1}{2}\mathbf{s}^\top \mathbf{D}\mathbf{s}\right\} \tag{17}$$

$$= \frac{1}{(2\pi)^{d/2} Z_b \exp\left\{\frac{1}{2}\mathrm{Tr}(\mathbf{D})\right\}} \exp\left\{-\frac{1}{2}\mathbf{x}^\top \mathbf{x}\right\} \prod_{k=1}^{d_b} \exp\left\{s_k(\mathbf{q}_k^\top \mathbf{x} + b_k)\right\}, \tag{18}$$

$$\tag{19}$$

where $\{\mathbf{q}_k^\top\}_{k=1}^{d_b}$ are the rows of $\mathbf{Q}$. The key feature of this trick is that the $\frac{1}{2}\mathbf{s}^\top \mathbf{W}\mathbf{s}$ term cancel.

On the joint space the binary variables **s** variables are now decoupled and can summed over independently to give the marginal density,

$$\pi(\mathbf{x}) = \frac{2^{d_b}}{(2\pi)^{d/2} Z_b \exp\left\{\frac{1}{2}\mathrm{Tr}(\mathbf{D})\right\}} \exp\left\{-\frac{1}{2}\mathbf{x}^\top\mathbf{x}\right\} \prod_{i=k}^{d_b} \cosh(\mathbf{q}_k^\top\mathbf{x} + b_k), \qquad (20)$$

which is referred to as the Boltzmann machine relaxation density, which is a Gaussian mixture with 2^{d_b} components.

We can rearrange the terms in the Boltzmann machine relaxation density to match our generic target $\pi(\mathbf{x}) = Z^{-1}\exp\{-\phi(\mathbf{x})\}$, (1), where

$$\phi(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top\mathbf{x} - \sum_{k=1}^{d_b} \log\cosh(\mathbf{q}_k^\top\mathbf{x} + b_k),$$

and the normalising constant is directly related to the Boltzmann machine distribution

$$Z = \log Z_b + \frac{1}{2}\mathrm{Tr}(\mathbf{D}) + \frac{d}{2}\log(2\pi) - d_b\log 2.$$

Converting a discrete problem onto the continuous space does not automatically guarantee that sampling from the continuous space will be any easier than on the discrete space. In fact, if the elements of $\mathbf{D}$ are large, then on the relaxed space, the modes of the 2^{d_b} mixture components will be far apart making it difficult for an MCMC sampler to explore the target. Following Zhang et al. (2012), for the experiments in this paper we select $\mathbf{D}$ by minimising the maximum eigenvalue of $\mathbf{W} + \mathbf{D}$ which has the effect of decreasing the separation of the mixture components on the relaxed space.

Finally, the first two moments of the relaxed distribution can be directly related to their equivalent moments for the Boltzmann machine distribution by

$$\mathbb{E}[\mathbf{X}] = \int_{\mathcal{X}} \mathbf{x}\sum_{\mathbf{s}\in\mathcal{S}}\pi(\mathbf{s}|\mathbf{x})P(\mathbf{s})\mathrm{d}\mathbf{x} = \sum_{\mathbf{s}\in\mathcal{S}}\left[\int_{\mathcal{X}}\mathbf{x}\mathcal{N}(\mathbf{x}|\mathbf{Q}^\top\mathbf{s}, \mathbf{I})\mathrm{d}\mathbf{x}P(\mathbf{s})\right] = \mathbb{E}[\mathbf{Q}^\top\mathbf{S}] = \mathbf{Q}^\top\mathbb{E}[\mathbf{S}],$$

$$\mathbb{E}[\mathbf{X}\mathbf{X}^\top] = \sum_{\mathbf{s}\in\mathcal{S}}\left[\int_{\mathcal{X}}\mathbf{x}\mathbf{x}^\top\mathcal{N}(\mathbf{x}|\mathbf{Q}^\top\mathbf{s}, \mathbf{I})\mathrm{d}\mathbf{x}P(\mathbf{s})\right] = \mathbb{E}[\mathbf{Q}^\top\mathbf{S}\mathbf{S}^\top\mathbf{Q} + \mathbf{I}] = \mathbf{Q}^\top\mathbb{E}[\mathbf{S}\mathbf{S}^\top] + \mathbf{I}.$$