THE UNIVERSITY *of* EDINBURGH

Edinburgh Research Explorer

Pseudo-Marginal Slice Sampling

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Publisher's PDF, also known as Version of record

Published In:
Proceedings of the 19th International Conference on Artificial Intelligence and Statistics 2016

OPEN ACCESS

Pseudo-Marginal Slice Sampling

Iain Murray
University of Edinburgh

Matthew M. Graham
University of Edinburgh

Abstract

Markov chain Monte Carlo (MCMC) methods asymptotically sample from complex probability distributions. The pseudo-marginal MCMC framework only requires an unbiased estimator of the unnormalized probability distribution function to construct a Markov chain. However, the resulting chains are harder to tune to a target distribution than conventional MCMC, and the types of updates available are limited. We describe a general way to clamp and update the random numbers used in a pseudo-marginal method's unbiased estimator. In this framework we can use slice sampling and other adaptive methods. We obtain more robust Markov chains, which often mix more quickly.

1 Introduction

Markov chain Monte Carlo (MCMC) methods asymptotically sample from a user-specified probability distribution, by simulating a Markov chain with the required equilibrium distribution. Most MCMC methods require an ability to evaluate the target distribution up to a constant. Pseudo-marginal MCMC (Andrieu and Roberts, 2009) only requires unbiased estimates of the target distribution. Pseudo-marginal MCMC has been applied across a range of domains, with applications in genetic population modelling (Beaumont, 2003), continuous time stochastic processes (Georgoulas et al., 2015), hierarchical models involving Gaussian processes (Filippone and Girolami, 2014), and 'doubly-intractable' distributions (Murray et al., 2006) such as undirected graphical models.

As the number and complexity of applications of MCMC increases, methods that adapt free parame-

ters for the user are becoming more popular (e.g. Hoffman and Gelman, 2014; Murray et al., 2010; Roberts and Rosenthal, 2009). Slice sampling (Neal, 2003) is relatively insensitive to its free parameters, and can locally adapt to the target density. Unfortunately, these methods either can't be applied with pseudo-marginal MCMC methods, or no longer work as designed.

In this paper we explore a simple means to bring together the advantages of pseudo-marginal methods and slice sampling. Like recent work on particle Gibbs methods (Chopin and Singh, 2015), we identify randomness used in an unbiased estimator as auxiliary variables that could be updated with a variety of MCMC methods. The framework is simple, and requires only a small change to existing pseudo-marginal methods that use Metropolis–Hastings style updates. Where the estimator has high variance, our methods can lead to a large increase in performance. The framework also allows Metropolis–Hastings step sizes to be tuned, or slice sampling to be used, and can make pseudo-marginal chains more robust and less prone to sticking.

2 Background

The standard task for Markov chain Monte Carlo (MCMC) is to draw correlated samples from a target distribution of interest

$$\pi(\theta) = f(\theta)/C, \qquad (1)$$

where we need to be able to evaluate $f(\theta)$ pointwise, but not the normalizer C.

2.1 Pseudo-Marginal MCMC

In this paper we consider cases where $f(\theta)$ cannot easily be evaluated pointwise, but where an unbiased estimator $\widehat{f}(\theta)$ can be sampled for any given state θ. This estimator might come from standard importance sampling estimates, particle methods (Andrieu et al., 2010), or randomized series truncation (Lyne et al., 2015; Georgoulas et al., 2015).

In the 'pseudo-marginal' or 'exact-approximate' paradigm (Andrieu and Roberts, 2009), we can simulate

---

Inputs: current state θ, previous estimate of its unnormalized target probability $\widehat{f}$, proposal dist. q, unbiased estimator s.t. $\mathbb{E}_{\epsilon(\widehat{f};\,\theta)}[\widehat{f}] = f(\theta)$ for all θ,

Output: new state-estimate pair $(\theta, \widehat{f})$.

---

1. Propose new state and estimate its probability:
$$\theta' \sim q(\,\cdot\,; \theta)$$
$$\widehat{f}' \sim \epsilon(\,\cdot\,; \theta')$$

2. Metropolis–Hastings style acceptance rule,

    with probability $\min\left(1,\ \dfrac{\widehat{f}'}{\widehat{f}}\dfrac{q(\theta; \theta')}{q(\theta'; \theta)}\right)$:

    Accept: **return** $(\theta', \widehat{f}')$

    else:

    Reject: **return** $(\theta, \widehat{f})$

Figure 1: Pseudo-marginal update as analysed by Andrieu and Roberts (2009).

the target distribution (1) by constructing a Markov chain on a state pair $(\theta, \widehat{f})$. The state evolves according to the algorithmic update in Figure 1. When the estimator is deterministic and exact — $\widehat{f}(\theta) = f(\theta)$ for all θ — the update is precisely the Metropolis–Hastings (MH) algorithm, with acceptance ratio

$$a = \frac{\pi(\theta')\,q(\theta; \theta')}{\pi(\theta)\,q(\theta'; \theta)} = \frac{f(\theta')\,q(\theta; \theta')}{f(\theta)\,q(\theta'; \theta)}. \qquad (2)$$

When the estimator is noisy, the estimate for the current state must be stored and reused until a move is accepted. That detail is made explicit by incorporating the estimate $\widehat{f}$ into the Markov chain state. Special cases of the pseudo-marginal algorithm had previously been identified and explored (Kennedy and Kuti, 1985; Beaumont, 2003; Møller et al., 2006).

Because $\widehat{f}$ is part of the Markov chain state, we must keep *both* the current variables θ *and* its corresponding estimate $\widehat{f}$ if the MH rule rejects a move. We can't simply replace the noisy function value with a new draw of the estimator from its distribution $\epsilon(\widehat{f}; \theta)$. When the distribution of the estimator is heavy-tailed, it is hard to accept a change away from a large $\widehat{f}$, and the chain can stop moving for a long time. Pseudo-marginal algorithms are notorious for this 'sticking' behavior.

2.2 Doubly-Intractable Distributions

Pseudo-marginal methods can be applied to *doubly-intractable* target distributions. These arise when inferring the parameters θ from a data generating process

$$p(y \,|\, \theta) = g(y; \theta) \,/\, Z(\theta), \qquad (3)$$

where $g(y; \theta)$ is a function we can evaluate, but $Z(\theta)$ is an intractable function of the parameters. Given a prior over parameters $p(\theta)$ and observation y, the target posterior distribution from Bayes' rule is:

$$\pi(\theta) = p(y \,|\, \theta)\,p(\theta)\big/p(y) = f(\theta)\big/C. \qquad (4)$$

We can choose to write this target distribution with reference to a model with fixed parameters $\hat{\theta}$:

$$f(\theta) = g(y; \theta)\,p(\theta)\,\frac{Z(\hat{\theta})}{Z(\theta)},\ \text{ with unbiased estimator}$$

$$\widehat{f}(\theta) = g(y; \theta)\,p(\theta)\,\frac{g(x; \hat{\theta})}{g(x; \theta)}, \quad x \sim p(y \,|\, \theta). \qquad (5)$$

This importance sampling estimator requires a sample from each model considered. Formally, exact/perfect sampling methods such as 'coupling from the past' (Propp and Wilson, 1996) are required. Using estimator (5) in the pseudo-marginal framework corresponds to the MCMC method proposed by Møller et al. (2006).

2.3 Slice Sampling

Slice sampling (Neal, 2003) is a family of algorithms with update mechanisms that can locally adapt to the target density. The Markov chain explores the uniform distribution underneath the $f(\theta)$ surface, so that the probability of being above setting θ is proportional to $\pi(\theta)$ as desired. A state/height pair (θ, h) are updated alternately according to Figure 2. Neal (2003) proved that this procedure (with some conditions) will sample from the target distribution.

The algorithm in Figure 2 has a step-size parameter w, but can adapt to bad settings. If the step-size is too small, step 3 of the algorithm can expand the interval explored. If the interval is too large, an adaptive rejection procedure in steps 4–7 will shrink the interval exponentially quickly towards the current state until an acceptable update is found. Given a continuous function $f(\theta)$, the update always moves the current state.

3 Noisy Slice Sampling

As slice sampling algorithms like Figure 2 always move the variables being updated, it is tempting to apply them in the pseudo-marginal setting, where chains often stick. It turns out that slice sampling is still valid if replacing the function $f(\theta)$, with a noisy but unbiased estimate $\widehat{f}(\theta)$. A special case is discussed by Murray (2007, §5.8.2). Intuitively, exploring uniformly underneath the noisy surface means spending time in a region above θ proportional to the average value of $\widehat{f}(\theta)$, and thus sampling from the target distribution $\pi(\theta)$.

Although slice sampling with noisy values can be valid, the local adaptation of the proposal interval is not

Input: current state θ, unnormalized target distribution f, initial search width w, whether to do optional part of update step_out.

Output: a new state θ'. When θ is drawn from $\pi(\theta) \propto f(\theta)$, the marginal distribution of θ' is also π.

1. Random height under curve:
$$u_1 \sim \text{Uniform}[0, 1]$$
$$h \leftarrow u_1 f(\theta)$$

2. Randomly place interval around the current state:
$$u_2 \sim \text{Uniform}[0, w]$$
$$[\theta_{\min}, \theta_{\max}] \leftarrow [\theta - u_2, \ \theta - u_2 + w]$$

3. **if** step_out expand interval (linear step version):
    while $f(\theta_{\min}) > h$: $\theta_{\min} \leftarrow \theta_{\min} - w$
    while $f(\theta_{\max}) > h$: $\theta_{\max} \leftarrow \theta_{\max} + w$

4. Sample proposal on interval:
$$\theta' \sim \text{Uniform}[\theta_{\min}, \theta_{\max}]$$

5. **if** $f(\theta') \leq h$ **then:**
    Shrink the bracket and try a new point:
6.   **if** $\theta' < \theta$ **then:** $\theta_{\min} \leftarrow \theta'$ **else:** $\theta_{\max} \leftarrow \theta'$
7.   **GoTo** 4.
8. **else:**
9.   Accept: **return** θ'

Figure 2: Single-variable slice sampling with linear stepping out, adapted from Neal (2003).

designed for this use case. If $f(\theta)$ is occasionally overestimated by a large amount, even nearby and equally probable states will be deemed unacceptable. In practice the proposal interval can easily collapse to the current state to numerical machine precision. Although the variables of interest θ are no longer changing, an implementation must be careful to either keep proposing $(\theta, \widehat{f})$ states until an acceptable $\widehat{f}$ has been found, or to return the original $(\theta, \widehat{f})$ pair from before the update began. Either way, the Markov chain will effectively contain rejections.

4   Clamping randomness

We aim to ease exploration of variables θ while exploiting a noisy estimator $\widehat{f}$, by removing the noise from the update. We will assume as little about the estimators as possible, to keep the appealing 'black-box' spirit of the original pseudo-marginal framework.

We assume that the estimation procedure uses random numbers from a convenient distribution, $q(\mathbf{u})$, such as a uniform or Gaussian distribution. This assumption is not strong: the choices $\mathbf{u}$ could be the results of all of the calls to a random number generator (often called `rand()`) within the computer code for the esti-

mator. The estimate $\widehat{f}(\theta; \mathbf{u})$ is then deterministically computed for the current variables given these choices.

Instead of sampling $\mathbf{u}$ from a random number generator, we will evolve them as part of a Markov chain on a joint auxiliary target distribution:

$$\pi(\theta, \mathbf{u}) = \widehat{f}(\theta; \mathbf{u}) \, q(\mathbf{u}) \, / \, C. \tag{6}$$

If the estimator is unbiased, we know its average under its random choices $\mathbf{u}$:

$$\mathbb{E}_{q(\mathbf{u})}[\widehat{f}] = \int \widehat{f}(\theta; \mathbf{u}) \, q(\mathbf{u}) \, \mathrm{d}\mathbf{u} = f(\theta). \tag{7}$$

Then the marginal distribution over the variables of interest, $\pi(\theta) = \int \pi(\theta, \mathbf{u}) \, \mathrm{d}\mathbf{u} = f(\theta)/C$, is the user-specified target marginal distribution (1).

Pseudo-marginal MCMC can be seen as a standard Metropolis–Hastings update on the auxiliary target distribution (6). After proposing new variables with probability $q(\theta'; \theta)$ we also propose new random choices with probability $q(\mathbf{u}')$ giving joint proposal probability

$$q(\theta', \mathbf{u}'; \theta, \mathbf{u}) = q(\theta'; \theta) \, q(\mathbf{u}'). \tag{8}$$

Substituting (6) and (8) into the MH acceptance ratio (2) gives

$$a = \frac{\pi(\theta', \mathbf{u}') \, q(\theta, \mathbf{u}; \theta', \mathbf{u}')}{\pi(\theta, \mathbf{u}) \, q(\theta', \mathbf{u}'; \theta, \mathbf{u})} = \frac{\widehat{f}(\theta'; \mathbf{u}') \, q(\theta; \theta')}{\widehat{f}(\theta; \mathbf{u}) \, q(\theta'; \theta)}, \tag{9}$$

as used in the acceptance rule in Figure 1.

4.1   Alternative transition operators

Splitting out the random choices $\mathbf{u}$ allows us to apply Markov chain operators that were not available before. Given a joint distribution $\pi(\mathbf{u}, \theta)$, we can alternately update $\mathbf{u}$ and θ using any standard MCMC updates for the conditional target distributions $\pi(\mathbf{u} \mid \theta)$ and $\pi(\theta \mid \mathbf{u})$. This is our simple *Auxiliary Pseudo-Marginal* (APM) framework, Figure 3.

We could implement the first step in Figure 3 by a Metropolis Independence (MI) proposal. We propose new random choices with probability $q(\mathbf{u}')$, and accept or reject them with the standard Metropolis ratio, which here simplifies to $\widehat{f}(\theta; \mathbf{u}')/\widehat{f}(\theta; \mathbf{u})$. This step is a standard pseudo-marginal update, but where the proposal for variables θ has zero-width.

The difference when updating the variables θ in the second step of Figure 3 is that the random choices $\mathbf{u}$ are fixed. This step could be implemented by simply resetting the seed of a pseudo-random number generator to the seed used to produce the estimate that was last accepted in step 1. Given clamped random numbers, we have the usual case of a conditional distribution

Inputs: current state: parameters θ, randomness $\mathbf{u}$; unbiased estimator s.t. $\mathbb{E}_{q(\mathbf{u}')}[\widehat{f}(\theta';\mathbf{u}')] = f(\theta')$ for all θ', **Output:** new state $(\theta, \mathbf{u})$.

1. Update $\mathbf{u}$ leaving invariant its target conditional:
$$\pi(\mathbf{u}\,|\,\theta) \propto \widehat{f}(\theta;\mathbf{u})\,q(\mathbf{u})$$

2. Update θ leaving invariant its target conditional:
$$\pi(\theta\,|\,\mathbf{u}) \propto \widehat{f}(\theta;\mathbf{u})$$

Figure 3: Framework for Auxiliary Pseudo-Marginal (APM) methods. Here 'leaving invariant' means that if a variable was drawn from the specified probability distribution before an update, it will retain the same distribution after the update. This property is satisfied by all standard MCMC update rules.

proportional to a deterministic function. Slice sampling will move θ if $\widehat{f}(\theta;\mathbf{u})$ is continuous almost everywhere.

Although we can now ensure that the variables of interest θ will almost always move, we might still suffer from sticking of the random choices $\mathbf{u}$. Rather than making global, Metropolis Independence (MI) proposals from $q(\mathbf{u})$, we could instead attempt to perturb $\mathbf{u}$ using other MCMC update rules.

Naming scheme: In the rest of the paper we will consider Auxiliary Pseudo-Marginal (APM) methods with a variety of MCMC algorithms providing the updates for the two steps in Figure 3. We will use the shorthand MI for Metropolis Independence updates, MH for perturbative Metropolis–Hastings updates, and SS for slice sampling. We will specify the update of the random choices $\mathbf{u}$ first, followed by the update for θ. For example, an APM method which uses a Metropolis Independence update for $\mathbf{u}$ given θ, and a slice sampling update for θ given $\mathbf{u}$, will be denoted APM MI+SS.

4.2 Slice Sampling auxiliary u variables

As discussed above, we recommend trying perturbative proposals of the random choices $\mathbf{u}$. In this paper we will only consider methods with no tweak parameters, as we want our algorithms to be easy to use.

We will use *Elliptical Slice Sampling* (Murray et al., 2010) if $q(\mathbf{u})$ is Gaussian. This algorithm has no free parameters, and only requires samples from $q(\mathbf{u})$ and the ability to evaluate $\widehat{f}(\theta;\mathbf{u})$. The algorithm initially attempts to make large, nearly-independent moves and should work well if the original Metropolis Independence updates do. Like conventional slice sampling, the elliptical version can also back off to smaller moves exponentially quickly.

In many implementations of estimators, the initial

random choices $\mathbf{u}$ will be uniformly distributed. Here we use a variant of linear slice sampling (like Figure 2) but applied along a random direction. We sample a direction vector ν (with a random length) by sampling elements independently from a standard normal, $\nu_i \sim \mathcal{N}(0, 1)$. The slice sampler explores an interval aligned with this direction, and reflects off the unit hypercube boundaries. Our implementation updates a variable z, which defines the new setting of the randomness:

$$u_i' = \text{Reflect}(u_i + z\nu_i), \qquad (10)$$

where we define $m = \text{mod}(x, 2)$ and then

$$\text{Reflect}(x) = \begin{cases} m & m < 1 \\ 2 - m & m \geq 1 \end{cases}. \qquad (11)$$

Each update starts with $z = 0$. We fix the step width to $w = 1$ and omit the stepping out (Figure 2, step 3). The slice sampling routine needs to evaluate the target density proportional to $\widehat{f}(\theta;\mathbf{u}'(z))$. If the estimator function $\widehat{f}$ chooses the number of random numbers it uses on the fly, these can be lazily generated from (10). Each ν_i is an independent Gaussian variate. Each u_i is either a cached random number that was previously used, or it didn't affect the previous estimator outcome, and can be retrospectively sampled from a uniform distribution.

Because reflections have unit Jacobian, the original proofs for slice sampling follow through. Reflections in slice sampling have previously been considered by Downs et al. (2000) and Neal (2003).

5 Demonstrations

As a first illustration, we choose a 5-dimensional Gaussian target distribution, $\pi(\theta) = \mathcal{N}(\theta; 0, I)$. We can imagine this distribution results from a doubly-intractable setup (Section 2.2) with:

$$g(x;\theta) = \mathcal{N}(x;\theta, I), \quad y = 0, \quad p(\theta) = 1 \text{ (improper)},$$

We pretend the normalizer $Z(\theta) = \sqrt{2\pi}$ is unknown, and use the estimator (5) with reference parameter $\hat{\theta} = 0$. The estimator's Gaussian variates x, are created from standard normal draws in the usual way:

$$\mathbf{u} \sim \mathcal{N}(0, I), \quad x(\mathbf{u};\theta) = \mathbf{u} + \theta, \qquad (12)$$

so our methods use $q(\mathbf{u}) = \mathcal{N}(\mathbf{u}; 0, I)$.

Updates to the target variables θ were proposed from a spherical Gaussian with step size σ, $q(\theta';\theta) = \mathcal{N}(0, \sigma^2 I)$, or by one-dimensional slice sampling along a random direction (chosen by finding the direction of a random draw from $\mathcal{N}(0, I)$) with step-size $w = 4$ and no stepping out.

(a) PM-MH (b) APM MI+MH (c) $p(\theta_1)$ estimates (d) Cost-scaled autocorrelation
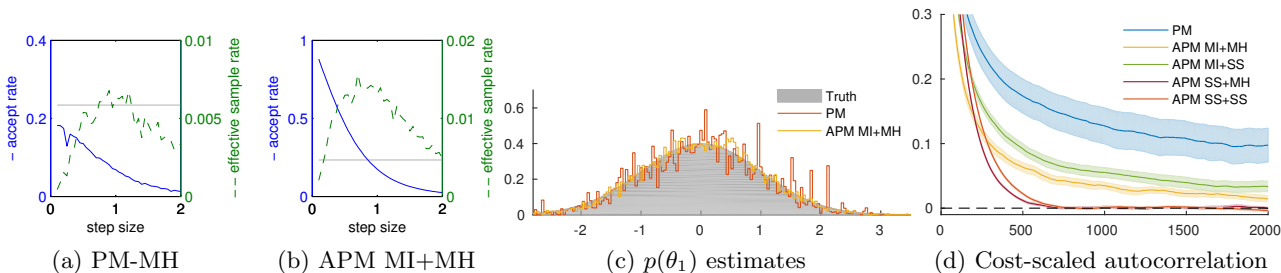
Figure 4: Sampling from a 5-dimensional Gaussian distribution, as if it were doubly-intractable. Acceptance rate and effective sample size per iteration for (a) pseudo-marginal MH updates (PM-MH), and (b) updates split into two steps, where randomness **u** in the estimator is 'clamped' while updating variables θ (APM MI+MH). The gray horizontal line, an acceptance rate of 0.234, indicated an MH step-size of 0.85 for further runs. (c) Estimates of a marginal distribution from 100,000 PM-MH updates, and 50,000 APM MI+MH update pairs. (d) Autocorrelation plotted against steps taken multiplied by the relative cost per step compared to standard PM updates.

Clearer step-size selection: The optimal step-size (or proposal standard deviation) for standard Metropolis proposals in high-dimensions (but not when updating one variable at a time) is the one that gives an acceptance rate of 0.234 under quite general conditions (Roberts et al., 1997). The usual pseudo-marginal algorithm, does not meet these conditions however. In fact, an acceptance rate of 0.234 isn't achievable in our Gaussian test case, even for small step sizes (Figure 4a). Thus step sizes for pseudo-marginal MH can't be tuned with the same heuristics as standard MH.

Splitting each Markov chain update into two steps, as in Figure 3, means we can perform conventional MH proposals on the variables θ. We use independent proposals for randomness **u**, by simply re-running our estimator code as usual without moving the variables θ. We then make Gaussian proposals for variables θ as before, while leaving **u** fixed. Figure 4b shows the acceptance rate for just the updates of variables θ, as a function of their step-size. As expected, the optimal acceptance rate now appears to be close to the theoretical value of 0.234.

Efficiency: It appears that the effective sample size (ESS) per update (Plummer et al., 2006) is slightly more than double what it was before. That implies a similar efficiency to before, given that each APM MI+MH update contains two evaluations of the estimator. However, ESS estimates are misleading on poorly mixing chains. Given the same amount of computation, APM MI+MH gives a smoother estimate of a marginal distribution, while PM still contains artifacts due to long-lasting sticking of its chain (Figure 4c).

Slice sampling: Naively running standard slice sampling using the noisy function $\widehat{f}(\theta)$ worked very poorly, with each update using >100 function evaluations on average. This idea was abandoned. Splitting up the **u** and θ updates allowed slice sampling the θ variables to

work, although slightly less well than MH after taking computation cost into account (Figure 4d). Elliptical slice sampling updates of the **u** variables gave large improvements in the autocorrelation of the chain, even when adjusting for the extra computation over the standard independent pseudo-marginal updates.

5.1 Ising model parameter posterior

Following previous studies (Møller et al., 2006; Murray et al., 2006), our second illustration is an Ising model distribution with $y_i \in \{\pm 1\}$ on a graph with nodes i and edges E:

$$p(y \,|\, \theta) = \frac{1}{Z(\theta)} \exp \Big(\sum_{i \neq j \in E} \theta_J y_i y_j + \sum_i \theta_h y_i \Big). \quad (13)$$

Our experiments used a 10×30 toroidal square Ising lattice. The data y were generated from an exact sample with $\theta_J = 0.3$ and $\theta_h = 0$. We used uniform priors over $|\theta_h| < 1$ and $0 < \theta_J < 0.4$, and an MH step-size of $\sigma = 0.04$.

We used the *Summary States* algorithm (Childs et al., 2001) to draw exact samples for the estimator in equation (5). The reference distribution $\hat{\theta}$ was set to the true parameters. Here we identify **u** as the infinite sequence of uniform random numbers used by a Gibbs sampler started infinitely long ago, which produces an exact sample from (13). The summary states algorithm lazily requests a finite subset of these numbers as required. When perturbing **u**, these were provided on demand by reflective slice sampling (Section 4.2).

We also considered 'MAVM' (Murray et al., 2006), which replaces the simple importance sampling in (5) with an annealed importance sampling (AIS) estimate (Neal, 2001). Here the random choices **u** consists of a fixed length of uniform draws required by the annealing steps of the algorithm, followed by a variable-length
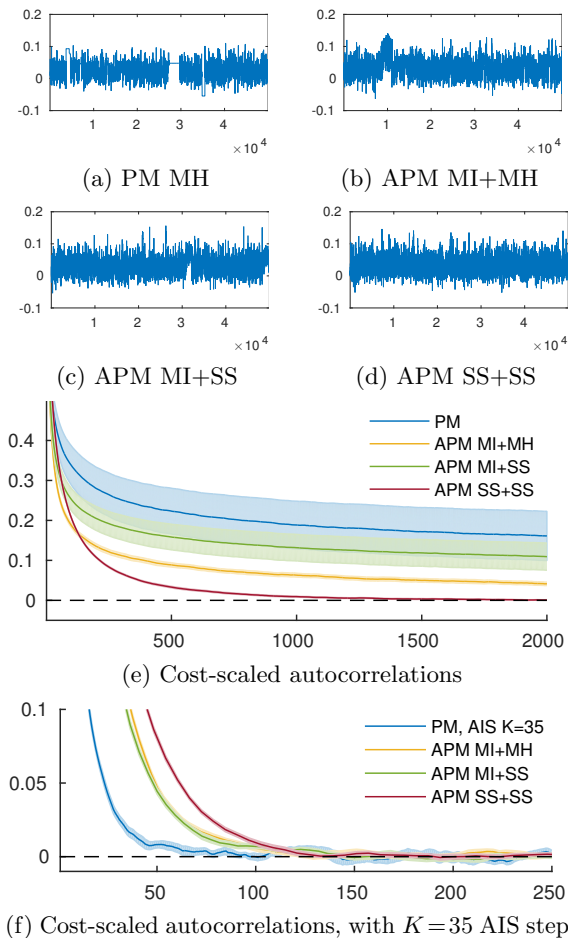
(a) PM MH

(b) APM MI+MH

(c) APM MI+SS

(d) APM SS+SS

(e) Cost-scaled autocorrelations

(f) Cost-scaled autocorrelations, with $K = 35$ AIS steps

Figure 5: Markov chains on the Ising model posterior. (a) Trace plot of θ_h against update number for the standard pseudo-marginal algorithm. (b)–(d) Traces from versions with separate Independent proposals (I), Gaussian Metropolis proposals (MH) or slice sampling updates (SS) for $\mathbf{u} + \theta$. (e) Empirical cost-scaled autocorrelations based on 10 runs of 2 million iterations each. The APM curves are stretched horizontally relative to the extra computation performed per iteration compared to the original algorithm. (f) Empirical auto-correlations when using an annealed importance sampling based estimator with $K = 35$ steps. The x-axis is scaled relative to the costs of one standard update without annealing.

sequence required for an initial exact sample as before.

Updates to the parameters θ_h and θ_J were applied sequentially using one-dimensional Gaussian proposals or slice sampling updates. Slice sampling used $w = 0.1$ (not carefully tuned) and no stepping out.

Efficiency: The Ising model example showed similar behavior to the toy Gaussian example. A trace plot of the first 5×10^4 updates (Figure 5a) shows that the

pseudo-marginal algorithm stuck on one occasion for > 2000 iterations. When $\mathbf{u}$ is updated with separate independent proposals, the parameters can move at each step. Under these updates (MI+MH and MI+SS) the auxiliary variables $\mathbf{u}$ still stick, with a noticeable effect on θ (Figures 5b, 5c). In contrast, slice sampling $\mathbf{u}$ forces it to move. The trace plot (Figure 5d) and autocorrelations (Figure 5e, adjusted for computational cost) appear greatly improved. The consistent behaviour of the SS+SS chain means that the empirical error bands on its autocorrelation are too small to see.

The empirical autocorrelations (Figure 5e) illustrate a danger of MCMC methods. It appears that the MI+MH updates have a tightly-determined autocorrelation compared to the more expensive MI+SS updates, from the same number of iterations. However, the poorer performance and large error band for MI+SS results from one extended transient, a change of behavior for $> 500,000$ iterations at the end of one run. As APM MI+MH shares the same independent $\mathbf{u}$ updates that cause the sticking, its performance is almost certainly over-stated, and longer runs would reveal that its autocorrelation is worse than reported.

The Markov chains can be improved by improving the variance of the estimator. Murray et al. (2006) reported only a modest improvement in effective sample size of $\sim 50\%$ by replacing importance sampling with annealed importance sampling (AIS). However, given the poor convergence we have observed without annealing, effective sample size estimates are not reliable. Figure 5f indicates that the autocorrelations with AIS are in fact dramatically better (note the different x-axis ranges). Applying our framework to this improved Markov chain gives Markov chains with still faster convergence per iteration. However, when scaling the autocorrelations by compute cost, as in Figure 5f, the original MAVM algorithm appears to be better than our proposed variants.

Most MCMC methods don't give guarantees of when estimates will reliably converge. Re-running APM SS+SS without annealing for 100 million further updates still didn't reveal any problems, which shows it is less prone to sticking than the base algorithm, but guarantees little. However, the marginal distributions show very close agreement to those found with annealing methods.

5.2 Gaussian process parameter inference

Our third illustration is a hierarchical Gaussian process classification model, following Filippone and Girolami (2014). The target distribution is the posterior $p(\theta \mid \mathbf{y}, X) \propto p(\mathbf{y} \mid \theta, X)\, p(\theta)$ on the model parameters θ given a set of n observed input features $X = \{\mathbf{x}_1 \ \ldots \ \mathbf{x}_n\}$ of dimension d and a corresponding vector $\mathbf{y}$ of n binary targets $y_i \in \{-1, +1\}$.

	Method	$N_{\text{c.op.}}/10^3$	Acc. rate	Variance σ			Length scale τ		
				ESS	$\frac{\text{ESS}}{N_{\text{c.op.}}}/10^{-3}$	$\hat{R}$	ESS	$\frac{\text{ESS}}{N_{\text{c.op.}}}/10^{-3}$	$\hat{R}$
Pima	PM MH	74.0 (0.026)	0.201 (0.0065)	306 (7.8)	4.14 (0.11)	1.00	441 (8.6)	5.96 (0.12)	1.00
	APM MI+MH	74.1 (0.029)	0.219 (0.0034)	357 (8.8)	4.81 (0.12)	1.00	512 (19)	6.92 (0.26)	1.00
	APM SS+MH	74.1 (0.028)	0.204 (0.0046)	370 (7.1)	4.99 (0.097)	1.00	526 (26)	7.11 (0.35)	1.00
Breast	PM MH	97.8 (0.14)	0.180 (0.013)	185 (19)	1.90 (0.20)	1.01	277 (28)	2.83 (0.28)	1.00
	APM MI+MH	98.3 (0.060)	0.208 (0.0046)	533 (5.8)	5.43 (0.059)	1.00	559 (13)	5.69 (0.13)	1.00
	APM SS+MH	98.4 (0.054)	0.206 (0.0044)	519 (9.8)	5.27 (0.099)	1.00	631 (13)	6.41 (0.13)	1.00

Table 1: Convergence and efficiency results for Gaussian process parameter inference on two UCI classification data sets. All figures are means (standard error estimates given in parenthesis) across 10 independent chains initialised from the prior except for the Gelman–Rubin $\hat{R}$ statistic which is calculated across all chains. Effective sample sizes ESS are shown for both variance hyperparameter σ and length scale τ as well as value normalised by the total number of $\mathcal{O}(n^3)$ matrix operations computed during each run, $N_{\text{c.op.}}$ for a measure of overall computational efficiency.

The latent function values $\mathbf{f}$ have a probit likelihood $p(\mathbf{y}\,|\,\mathbf{f}) = \prod_{i=1}^{n} \Phi(y_i f_i)$, and a zero-mean Gaussian process prior $p(\mathbf{f}\,|\,\theta, X) = \mathcal{GP}(\mathbf{f}\,|\,\mathbf{0}, K)$ with an isotropic squared exponential covariance function. The parameters $\theta = (\sigma, \tau)$ are the variance σ and length-scale τ of the covariance. Both σ and τ have Gamma priors.

The marginal likelihood term $p(\mathbf{y}\,|\,\theta, X)$ in the target posterior cannot be evaluated exactly as the integral to marginalise out the latent function values does not have a closed form solution. Filippone and Girolami (2014) propose using pseudo-marginal MH to sample from the target posterior. An importance sampling estimate

$$\hat{p}(\mathbf{y}\,|\,\theta, X) = \frac{1}{N_{\text{imp}}} \sum_{i=0}^{N_{\text{imp}}} \left[\frac{p(\mathbf{y}\,|\,\mathbf{f}_i)\,p(\mathbf{f}_i\,|\,\theta, X)}{q(\mathbf{f}_i\,|\,\mathbf{y}, \theta, X)} \right], \quad (14)$$
$$\mathbf{f}_i \sim q(\cdot\,|\,\mathbf{y}, \theta, X),$$

where $q(\mathbf{f}\,|\,\mathbf{y}, \theta, X)$ is a Gaussian approximation to the posterior on the latent function values, is used as an unbiased estimator of $p(\mathbf{y}\,|\,\theta, X)$.

We compared the performance of the pseudo-marginal MH algorithm to two auxiliary pseudo-marginal variants: MI+MH and SS+MH. Here $\mathbf{u}$ was the fixed-sized vector of standard normal random draws that is transformed to generate samples from the Gaussian approximate posterior $q(\mathbf{f}\,|\,\mathbf{y}, \theta, X)$. We used elliptical slice sampling to update $\mathbf{u}$ in APM SS+MH.

We specifically considered Gaussian process parameter inference on two UCI classification datasets tested by Filippone and Girolami (2014, §4.4): the Pima data set ($d=9$, $n=682$) and the Breast data set ($d=8$, $n=768$). For a full description of the approximation scheme used in (14), parameter prior hyperparameters and other implementation details please refer to Filippone and Girolami (2014).
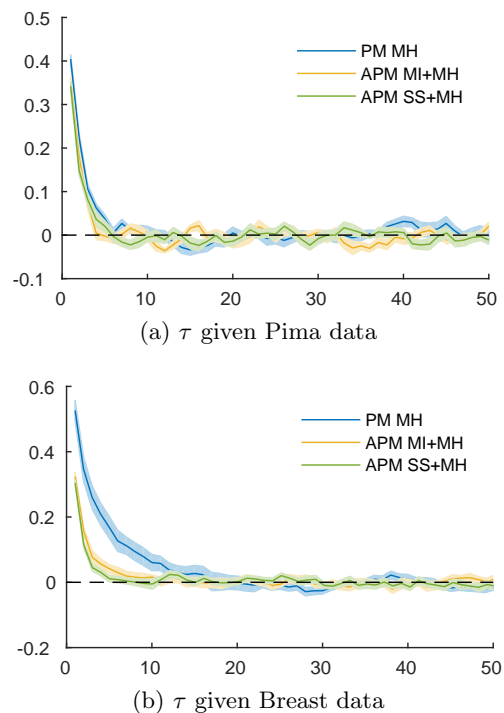

(a) τ given Pima data

(b) τ given Breast data

Figure 6: Cost-scaled autocorrelations (after thinning to every 10 iterations) for Gaussian Process models.

Ease of tuning: All three methods considered used MH updates for θ so we employed the adaptive scheme used in Filippone and Girolami (2014) to tune the proposals to try to achieve an acceptance rate between 0.15 and 0.3. Filippone and Girolami (2014) noted the possibility of chains getting stuck due to large overestimates of the marginal likelihood, and suggested it could upset the adaptive process. Therefore a biased but deterministic approximation for the marginal likelihood was used in the MH accept step during the initial adaptive phase. Preliminary experiments indicated that their

approach worked better, and so was replicated in our experiments for the PM MH method.

The Auxiliary Pseudo-Marginal (APM) framework is meant to provide a clearer signal for adapting proposals, as accept/reject decisions are based on a paired comparison between θ values with the same random $\mathbf{u}$ draw. This intended advantage seems to work in practice: the final acceptance rate in all 40 of the APM chains we ran ended up within the desired acceptance rate bounds. For the standard pseudo-marginal MH case, 2 out of the 20 chains did not achieve an acceptance rate within the target bounds.

Efficiency: The sampling efficiencies of the tested auxiliary pseudo-marginal methods and original pseudo-marginal MH method are summarised in Table 1. Also shown are the Gelman–Rubin $\hat{R}$ convergence statistics (Gelman and Rubin, 1992) for each set of 10 chains: values far from unity would demonstrate failure to converge. Although the diagnostic cannot prove convergence, the fact that all of the calculated $\hat{R}$ are unity to within 0.01 is at least comforting. The autocorrelations (Figure 6) also go to zero quickly.

The effective sample sizes for the two auxiliary pseudo-marginal methods both show significant improvements over the standard pseudo-marginal MH method, with the gain being particularly large in the Breast data set where the ESS is more than doubled and the autocorrelations also show a quicker decay to zero (Figure 6b). Traces (not shown) for the PM MH runs on the Breast data set showed heavy incidence of the chain sticking. These artefacts are also visible for this data set in Filippone and Girolami (2014)'s plots. The auxiliary pseudo-marginal methods appear to mix much better with no extended rejection intervals.

In general the auxiliary pseudo-marginal framework has some computational overhead over standard pseudo-marginal due to splitting the update into two steps. For this Gaussian process task however the dominant computational cost during sampling is in $\mathcal{O}(n^3)$ matrix decomposition operations, which are only required when considering a new θ. Counts for the total number of cubic operations $N_{\mathrm{c.op.}}$ across each run were recorded and the mean values for each method and dataset are shown in Table 1. The computational cost for all three methods was effectively equivalent (in our implementation the wall-clock time of SS+MH was $\sim 6\%$ more per update than MI+MH).

Between the two auxiliary pseudo-marginal methods, applying elliptical slice sampling to the random draws $\mathbf{u}$ seems to give a small gain in sampling efficiency over Metropolis independence sampling or no significant difference. Falling back to smaller intervals to ensure the chain always moves appears to help to some extent.

6 Discussion

Auxiliary variable interpretations have previously been exploited in particle MCMC methods, providing new and better update rules (e.g. Chopin and Singh, 2015). Since submitting this work, we've been told that the idea of clamping random number draws in pseudo-marginal methods was first proposed by Lee and Holmes (2010). There have also been two independent proposals to update Gaussian random numbers within arbitrary pseudo-marginal methods (Deligiannidis et al., 2015; Dahlin et al., 2015). Both of these proposals use the Metropolis proposal mechanism that elliptical slice sampling generalizes. Without slice-sampling exploration, these methods will require tuning, and could be less robust. However, Metropolis updates are simpler to analyse and these related works contain interesting theoretical analysis of the Markov chains.

The main aim in our presentation of the APM framework is to provide tuning-free ways to improve pseudo-marginal MCMC methods. Using the framework to clamp the auxiliary randomness $\mathbf{u}$ requires no alteration to an estimator's existing code: one can simply set the random seed it uses. Users can employ their preferred updates for the target variables θ (and can now tune their step-sizes reliably). Using the proposed slice sampling methods to update random draws $\mathbf{u}$ requires replacing calls to a random number generator with access to a Markov chain state. Apart from that, a user can use either our black-box reflective slice-sampling scheme, or elliptical slice sampling, with no tuning.

In a case where the estimator of the target distribution had low noise (annealed importance sampling applied to the Ising model), the overhead of our framework made the wall-clock time for our Markov chain to converge longer. If prepared to tune free choices, the extra cost of our framework could be reduced by clamping $\mathbf{u}$ for several iterations.

However, our biggest concern is making the methods robust. In current pseudo-marginal methods, it is hard to know what effect noisy estimators will have on the surrounding Markov chain. Doucet et al. (2015) suggest how much computational effort to spend on reducing noise, but under some strong assumptions, which don't apply to our examples. In applications it is difficult to know if an estimator may be heavy tailed, or behave badly for some parameters θ. In the Ising example we saw APM MI+SS suddenly stick after more than a million iterations of apparently equilibrium behavior. Slice sampling the random choices in an estimator may cost a little extra, but provides a route for pseudo-marginal chains to take small steps out of difficulty. This robustness could be the difference between the Markov chain method working or not.

References

C. Andrieu and G. O. Roberts. The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, pages 697–725, 2009.

C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B*, 72(3):269–342, 2010.

M. A. Beaumont. Estimation of population growth or decline in genetically monitored populations. *Genetics*, 164:1139–1160, 2003.

A. M. Childs, R. B. Patterson, and D. J. C. MacKay. Exact sampling from nonattractive distributions using summary states. *Physical Review E*, 63:036113, 2001.

N. Chopin and S. S. Singh. On particle Gibbs sampling. *Bernoulli*, 21(3):1855–1883, 2015.

J. Dahlin, F. Lindsten, J. Kronander, and T. B. Schön. Accelerating pseudo-marginal Metropolis-Hastings by correlating auxiliary variables, 2015. arXiv:1511.05483v1.

G. Deligiannidis, A. Doucet, and M. K. Pitt. The correlated pseudo-marginal method, 2015. arXiv:1511.04992v2.

A. Doucet, M. K. Pitt, G. Deligiannidis, and R. Kohn. Efficient implementation of Markov chain Monte Carlo when using an unbiased likelihood estimator. *Biometrika*, 2015.

O. B. Downs, D. J. C. MacKay, and D. D. Lee. The nonnegative Boltzmann machine. In *Advances in Neural Information Processing Systems 12*, pages 428–434, 2000.

M. Filippone and M. Girolami. Pseudo-marginal Bayesian inference for Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2214–2226, 2014.

A. Gelman and D. B. Rubin. Inference from iterative simulation using multiple sequences. *Statistical science*, pages 457–472, 1992.

A. Georgoulas, J. Hillston, and G. Sanguinetti. Unbiased Bayesian inference for population Markov jump processes via random truncations, 2015. arXiv:1509.08327v1.

M. D. Hoffman and A. Gelman. The no-u-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15:1351–1381, 2014.

A. Kennedy and J. Kuti. Noise without noise: a new Monte Carlo method. *Physical review letters*, 54(23): 2473, 1985.

A. Lee and C. Holmes. Discussion of "particle Markov chain Monte Carlo methods". *Journal of the Royal Statistical Society, Series B*, 72(327), 2010.

A.-M. Lyne, M. Girolami, Y. Atchadé, H. Strathmann, and D. Simpson. Playing Russian roulette with intractable likelihoods, 2015.

J. Møller, A. N. Pettitt, R. Reeves, and K. K. Berthelsen. An efficient Markov chain Monte Carlo method for distributions with intractable normalising constants. *Biometrika*, 93(2):451–458, 2006.

I. Murray. *Advances in Markov chain Monte Carlo methods*. PhD thesis, Gatsby computational neuroscience unit, University College London, 2007.

I. Murray, Z. Ghahramani, and D. J. C. MacKay. MCMC for doubly-intractable distributions. In *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI-06)*, pages 359–366, 2006.

I. Murray, R. P. Adams, and D. J. C. MacKay. Elliptical slice sampling. *JMLR: W&CP*, 9:541–548, 2010.

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

R. M. Neal. Slice sampling. *Annals of Statistics*, 31(3): 705–767, 2003.

M. Plummer, N. Best, K. Cowles, and K. Vines. CODA: Convergence diagnosis and output analysis for MCMC. *R News*, 6(1):7–11, 2006.

J. G. Propp and D. B. Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms*, 9 (1&2):223–252, 1996.

G. Roberts and J. Rosenthal. Examples of adaptive MCMC. *J. Comp. Graph. Stat.*, 18:349–367, 2009.

G. O. Roberts, A. Gelman, and W. R. Gilks. Weak convergence and optimal scaling of random walk Metropolis algorithms. *The Annals of Applied Probability*, 7(1):110–120, 1997.