3 PSEUDO-MARGINAL METHODS

The *Markov chain Monte Carlo* (MCMC) methods considered in Chapter 2 provide a widely applicable set of tools for performing inference in probabilistic models where we can evaluate a, potentially unnormalised, density function for the target distribution of interest. In some models we may not be able to directly evaluate such a function however but instead have access to an unbiased estimator of the target density. The pseudo-marginal framework [2] allows MCMC methods to be extended to such problems.

The typical setting for pseudo-marginal methods is that a distribution on an extended set of variables is constructed which has the target distribution as a marginal. Values of a density function for the target distribution are then estimated by using a Monte Carlo method such as importance sampling to approximately marginalise out the additional variables. The variables which are marginalised out may correspond to latent variables specified in the model but that are not of direct interest for the inference task or variables introduced solely for computational reasons. In both cases it will usually be possible to specify a Markov transition operator which leaves the distribution on the extended set of variables invariant, with such schemes often being described as *data augmentation* [41, 42] or *auxiliary variable* [13, 19] methods. Here we will refer to any variables which are marginalised over as auxiliary variables and the variables of interest we wish to infer plausible values for as the target variables.

The density of the joint distribution on auxiliary and target variables will often have a complex geometry with strong dependencies between the variables and in some cases may be multimodal. This can lead to poor exploration of the extended space by simple MCMC schemes such as random-walk Metropolis–Hastings and Gibbs sampling [2]. The motivation for pseudo-marginal methods is that in some cases the density of the marginal distribution on the target variables will have a simpler geometry than the density of the joint distribution on the exten-

ded space and therefore be more amenable to exploration by standard MCMC methods.

Although in general we cannot analytically integrate out the auxiliary variables, the pseudo-marginal framework shows how an unbiased estimator of the marginal density can be used within a Metropolis–Hastings update while maintaining the asymptoptic exactness of standard MCMC methods. Intuitively the lower the variance of the density estimator the closer the behaviour of the algorithm to the case where the auxiliary variables are analytically marginalised out. We can control the variance of the estimator both by varying the number of auxiliary variable samples used in the Monte Carlo estimate and by using variance reduction methods to increase the estimator efficiency.

By posing the problem of specifying an MCMC algorithm in terms of designing an efficient[1] unbiased estimator of the density of interest, the large literature on methods for constructing low-variance unbiased estimators can be exploited. For example comparatively cheap but biased optimisation-based inference approaches such as Laplace's method (see Appendix C) can be combined with an importance sampling 'debiasing' step to produce an unbiased estimator which can then be used in a pseudo-marginal MCMC update. This provides a way of exploiting cheap but biased approximate inference methods within a MCMC method which still gives guarantees of asymptotically exact results.

The pseudo-marginal framework has been applied to a wide range of probabilistic models were inference might otherwise be intractable. However the standard pseudo-marginal method, which is based on a Metropolis–Hastings transition operator, is suspectible to 'sticking' behaviour where proposed moves are repeatedly rejected for many iterations [2, 40]. The method can also be difficult to tune as it breaks some of the assumptions underlying standard heuristics for adapting the parameters of Metropolis–Hastings methods.

In this chapter we will discuss an alternative formulation of the pseudo-marginal framework which bridges between the approach of directly specifying a Markov transition operator on the extended state space which includes the auxiliary variables and the pseudo-marginal method where the auxiliary variables are approximately marginalised out. This

1 We use 'efficient' in a general sense here rather than the notion of a minimum-variance unbiased estimator satisfying the Cramér-Rao lower bound [7, 37].

auxiliary pseudo-marginal framework still allows the intuitive design of pseudo-marginal algorithms in terms of identifying low-variance unbiased estimators, while overcoming some of the issues of the pseudo-marginal Metropolis–Hastings method. In particular it shows how more flexible adaptive MCMC algorithms such as slice-sampling can be used within the pseudo-marginal setting, which can improve the robustness and ease of application of the approach by minimising the amount of user-tuning of free parameters required.

The work summarised in this chapter is based on a collaboration with Iain Murray which resulted in the published conference paper

- Pseudo-marginal slice sampling. Iain Murray and Matthew M. Graham. *The Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, JMLR W&CP 51:911-919*, 2016.

Iain Murray was the main contributor of the ideas proposed in that publication and responsible for the 'doubly-intractable' Gaussian and Ising model experiments in Sections 5.1 and 5.2 of the paper. We discussed the presentation and details of the work together. My individual contribution was implementing and analysing the Gaussian process classification experiments summarised in Section 5.3 of that work, an extended version of which is reproduced in Section 3.6.2 of this Chapter. The Gaussian latent variable model experiments discussed in Section 3.6.1 were directly inspired by the experiments in Section 5.1 of the above paper, but we use a different latent variable model formulation for the model here and conduct additional empirical studies of the effect of the variance of the estimator on the relative performance of the algorithms and the sensitivity of the performance of the pseudo-marginal slice sampling algorithms to their free parameters. The text and figures in this chapter are all my own work, though inevitably some of the discussion and analysis is similar to sections of the above publication.

3.1 PROBLEM DEFINITION

As in the previous chapter our goal is to be able to compute estimates of expectations with respect to a target distribution of interest, that is integrals of the form

$$\bar{f} = \int_X f(\boldsymbol{x})\, P(\mathrm{d}\boldsymbol{x}) = \int_X f(\boldsymbol{x})\, p(\boldsymbol{x})\, \mu(\mathrm{d}\boldsymbol{x}) \tag{3.1}$$

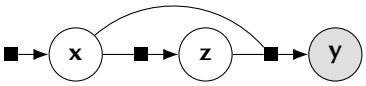


Figure 3.1: Hierarchical model factor graph.

where $f : X \to \mathbb{R}$ is an arbitrary Lebesgue integrable function and P is a probability distribution on a space X with density $p = \frac{\mathrm{d}P}{\mathrm{d}\mu}$. We assume as previously that the density p may have an intractable normalising constant C that we cannot evaluate i.e. $p(\boldsymbol{x}) = \tilde{p}(\boldsymbol{x})/C$. We make the further assumption here that we cannot directly evaluate $\tilde{p}$ either but only compute an unbiased, non-negative estimate of it. More explicitly we assume we can generate values of a non-negative random variable $\hat{\mathrm{p}}$ from a regular conditional distribution $\mathrm{P}_{\hat{\mathrm{p}}|\mathbf{x}}$ such that

$$\tilde{p}(\boldsymbol{x}) = \mathbb{E}[\hat{\mathrm{p}} \mid \mathbf{x} = \boldsymbol{x}] = \int_0^\infty \hat{p}\, \mathrm{P}_{\hat{\mathrm{p}}|\mathbf{x}}(\mathrm{d}\hat{p} \mid \boldsymbol{x}) \quad \forall \boldsymbol{x} \in X. \tag{3.2}$$

Note that we only require that we can generate $\hat{\mathrm{p}}$ values for a given $\mathbf{x}$, not that we can evaluate a density for $\mathrm{P}_{\hat{\mathrm{p}}|\mathbf{x}}$. For concreteness throughout the rest of this chapter we will assume that the target variables take values in a real-valued space $X = \mathbb{R}^D$ and that any density on these variables is defined with respect to the Lebesgue measure $\mu = \lambda^D$.

3.1.1 Example: hierarchical latent variable models

The application of pseudo-marginal methods we focus on is inference in hierarchical probabilistic models where the unobserved variables are split into global latent variables we are interested in inferring and local per datapoint latent variables that we wish to marginalise over the values of, as introduced in Section 1.3.1 in Chapter 1. For notational simplicity we here assume all observed variables are concatenated in a single vector $\mathbf{y}$ and likewise all associated local latent variables in a vector $\mathbf{z}$. The global latent variables, i.e. the target variables for inference, are then $\mathbf{x}$. A factor graph representing the factorisation across the model variables is shown in Figure 3.1.

The target distribution P is then the posterior distribution $\mathrm{P}_{\mathbf{x}|\mathbf{y}}$ given fixed observed values $\boldsymbol{y}$ and the unnormalised target density is chosen as the joint density $\tilde{p}(\boldsymbol{x}) = \mathrm{p}_{\mathbf{x},\mathbf{y}}(\boldsymbol{x}, \boldsymbol{y})$. We can express $\tilde{p}$ as a marginal of the joint density $\mathrm{p}_{\mathbf{x},\mathbf{y},\mathbf{z}}$, which assuming the latent variables $\mathbf{z}$ being

marginalised over are real-valued and have a density with respect to the Lebesgue measure can be written

$$\tilde{p}(\boldsymbol{x}) = \mathrm{p}_{\mathbf{x},\mathbf{y}}(\boldsymbol{x}, \boldsymbol{y}) = \int_Z \mathrm{p}_{\mathbf{x},\mathbf{y},\mathbf{z}}(\boldsymbol{x}, \boldsymbol{y}, z)\, \mathrm{d}z. \tag{3.3}$$

Generally this integral will not have an analytic solution. We can however form an unbiased estimate of (3.3) using importance sampling. We define a *importance distribution* Q which we can generate independent samples from and with a known density q which in general may depend on the values of the target variables $\boldsymbol{x}$ and observations $\boldsymbol{y}$. If $\{\mathbf{z}^{(n)}\}_{n=1}^N$ are a set of independent variables distributed according to Q then we can define a unbiased density estimator $\hat{\mathrm{p}}$ as

$$\hat{\mathrm{p}} = \frac{1}{N}\sum_{n=1}^{N} \frac{\mathrm{p}_{\mathbf{x},\mathbf{y},\mathbf{z}}(\mathbf{x}, \boldsymbol{y}, \mathbf{z}^{(n)})}{q(\mathbf{z}^{(n)} \mid \mathbf{x}, \boldsymbol{y})} \implies \mathbb{E}[\hat{\mathrm{p}} \mid \mathbf{x} = \boldsymbol{x}] = \tilde{p}(\boldsymbol{x}). \tag{3.4}$$

The variance $\mathbb{V}[\hat{\mathrm{p}}]$ is proportional to $\frac{1}{N}$ and so to decrease the estimator variance we can increase the number of importance samples used, however this comes with the tradeoff of an increased computational cost of each density estimate. The estimator variance will also be dependent on the importance distribution used. The optimal choice in terms of minimsing variance would be the conditional distribution $\mathrm{P}_{\mathbf{z}|\mathbf{x},\mathbf{y}}$. Under this choice the density 'estimate' takes the form

$$\hat{\mathrm{p}} = \frac{1}{N}\sum_{n=1}^{N} \frac{\mathrm{p}_{\mathbf{x},\mathbf{y},\mathbf{z}}(\mathbf{x}, \boldsymbol{y}, \mathbf{z}^{(n)})}{\mathrm{p}_{\mathbf{z}|\mathbf{x},\mathbf{y}}(\mathbf{z}^{(n)} \mid \mathbf{x}, \boldsymbol{y})} = \frac{1}{N}\sum_{n=1}^{N} \mathrm{p}_{\mathbf{x},\mathbf{y}}(\mathbf{x}, \boldsymbol{y}) = \tilde{p}(\mathbf{x}) \tag{3.5}$$

and so is equal to the unnormalised target density independent of the sampled $\mathbf{z}^{(n)}$ values with zero variance. In reality however we will not be able to evaluate the density of $\mathrm{P}_{\mathbf{z}|\mathbf{x},\mathbf{y}}$ nor sample from it as this is equivalent to being able to analytically solve the integral in (3.3).

The conditional distribution $\mathrm{P}_{\mathbf{z}|\mathbf{x}}$ will often be tractable to sample from and to evaluate the density of and so is a possible choice for the importance distribution. Typically however $\mathrm{P}_{\mathbf{z}|\mathbf{x}}$ will be much less concentrated than $\mathrm{P}_{\mathbf{z}|\mathbf{x},\mathbf{y}}$. This will mean samples from $\mathrm{P}_{\mathbf{z}|\mathbf{x}}$ will tend to fall in low density regions of $\mathrm{P}_{\mathbf{z}|\mathbf{x},\mathbf{y}}$, with only occassionally sampled values being in regions with high density under $\mathrm{P}_{\mathbf{z}|\mathbf{x},\mathbf{y}}$ leading to a high variance estimator, with the problem becoming more severe as the dimension of $\mathbf{z}$ increases. This can mean a large number of importance samples are needed to achieve an estimator with a reasonable variance.

An alternative is to fit an approximation to $\mathrm{P}_{\mathbf{z}|\mathbf{x},\mathbf{y}}$ to use as the importance distribution using for example one of the optimisation-based approximate inference approaches discussed in Appendix C. For example we could use Laplace's method to fit a multivariate normal approximation $\mathrm{p}_{\mathbf{z}|\mathbf{x},\mathbf{y}}(z \mid \boldsymbol{x}, \boldsymbol{y}) \approx \mathcal{N}\left(z \mid \boldsymbol{\mu}_{\boldsymbol{x},\boldsymbol{y}}, \Sigma_{\boldsymbol{x},\boldsymbol{y}}\right)$ and use this as the importance distribution. As $\mathrm{p}_{\mathbf{z}|\mathbf{x},\mathbf{y}}$ depends on $\boldsymbol{x}$ this involves fitting an approximation for each $\boldsymbol{x}$ value we wish to evaluate the density at. Although computationally costly the significant variance reduction brought by this approach can make this overhead worthwhile in pratice [14].

Inference in hierarchical latent variable models using an importance sampling estimator for the marginal density is just one setting in which pseudo-marginal methods are applied. Other applications of the framework have included inference methods for dynamical state space models using a particle filter estimator [10, 17] for the marginal density of the observed state sequence given the model parameters [1, 6, 34], parameter inference in 'doubly-intractable' distributions [29] where an intractable normaliser depends on the variables of interest using density estimators based on exact sampling methods [26, 27, 36] and random series trunctation [23] and approximate inference in simulator models where the density on the simulator outputs is only implicitly defined [25].

In the discussion and experiments in this chapter we will concentrate on latent variable models and importance sampling density estimators of the form described in this section. Examples of applying the methods discussed here to inference in a doubly intractable distribution were discussed in the associated conference paper [30]. Although particle filtering based methods are a major use case of the pseudo-marginal framework, the associated models and estimators tend to be more complex and we have chosen to avoid further expanding the theoretical background material in this thesis by concentrating on simpler cases here. The use of pseudo-marginal MCMC methods to perform inference in simulator models will be a major topic of the next chapter which specifically considers inference methods applicable in this setting so we will delay discussion of models of this form till then.

Algorithm 8 Pseudo-marginal Metropolis–Hastings.

Input: $(\boldsymbol{x}_n, \hat{p}_n)$: current target variables – density estimate state pair, $\mathrm{P}_{\hat{\mathrm{p}}|\mathbf{x}}$: density estimate conditional distribution, r : proposal density for updates to target variables.

Output: $(\boldsymbol{x}_{n+1}, \hat{p}_{n+1})$: new target variables – density estimate state pair.

$\boldsymbol{x}^* \sim r(\cdot \mid \boldsymbol{x}_n)$ ▷ Propose new values for target variables.
$\hat{p}^* \sim \mathrm{P}_{\hat{\mathrm{p}}|\mathbf{x}}(\cdot \mid \boldsymbol{x}^*)$ ▷ Estimate density at proposed $\boldsymbol{x}^*$.
$u \sim \mathcal{U}(\cdot \mid 0, 1)$
if $u < \frac{r(\boldsymbol{x}_n \mid \boldsymbol{x}^*)\,\hat{p}^*}{r(\boldsymbol{x}^* \mid \boldsymbol{x}_n)\,\hat{p}_n}$ **then**
    $(\boldsymbol{x}_{n+1}, \hat{p}_{n+1}) \leftarrow (\boldsymbol{x}^*, \hat{p}^*)$ ▷ Accept proposal.
else
    $(\boldsymbol{x}_{n+1}, \hat{p}_{n+1}) \leftarrow (\boldsymbol{x}_n, \hat{p}_n)$ ▷ Reject proposal.
return $(\boldsymbol{x}_{n+1}, \hat{p}_{n+1})$

3.2 PSEUDO-MARGINAL METROPOLIS–HASTINGS

The pseudo-marginal Metropolis–Hastings method is summarised in Algorithm 8. The term *pseudo-marginal* was proposed by Andrieu and Roberts in [2], with they also giving an extensive theoretical analysis of the framework. Andrieu and Roberts cite Beaumont [4] as the original source of the algorithm. Special cases of the algorithm have also been independently proposed, for example in the statistical physics literature by Kennedy and Kuti [20] and a MCMC method for doubly intractable distributions by Moller et al. [26].

The algorithm takes an intuitive form, with a very similar structure to the standard Metropolis–Hastings method (Algorithm 2) except for the ratio of densities in the accept probability calculation being replaced with a ratio of the density estimates. Importantly the stochastic density estimates are maintained as part of the chain state: if we reject a proposed update on the next iteration of the algorithm we reuse the same density estimate for the current state as in the previous iteration. This is required for the correctness of the algorithm, but also helps explain the sticking behaviour sometimes encountered with pseudo-marginal Metropolis–Hastings chains. If the density estimator distribution is heavy-tailed occassionally a estimate $\hat{p}_n$ will be sampled for the current target state $\boldsymbol{x}_n$ which is much higher than the expected value $\tilde{p}(\boldsymbol{x}_n)$. Assuming for simplicity a symmetric proposal density r is used such that the accept probability ratio in Algorithm 8 reduces to $\hat{p}^*/\hat{p}_n$, for subsequent proposed $(\boldsymbol{x}^*, \hat{p}^*)$ pairs the $\hat{p}^*$ values will typically be much smaller than the outlier $\hat{p}_n$ value and so the accept probability low. This can cause a long sequence of proposed moves being rejected

until a move is proposed to an $\boldsymbol{x}^*$ where the density is similar to $\hat{p}_n$ or another atypically high density estimate is proposed [2, 14, 40].

The efficiency of the pseudo-marginal Metropolis–Hastings update depends on how noisy the density estimates are and so the choice of the number of Monte Carlo samples N in the density estimate, for example the number of importance samples in (3.4). As N increases, the variance decreases and the algorithm becomes increasingly similar to performing standard Metropolis–Hastings updates under the (marginal) target distribution. Generally a chain will therefore mix better for larger N, with fewer sticking events. Typically however the computational cost of density estimate and so Metropolis–Hastings updates also increases with N and so there is a tradeoff between this improved mixing and increased per-update cost. Several theoretical studies have suggested guidelines for how to tune the parameters of the algorithm to optimise overall efficiency.

For Monte Carlo estimators formed as an average of unbiased estimators (such as the importance sampling estimator discussed above) and under an assumption of that the computational cost of each density estimate scales linearly with the number of Monte Carlo samples N, it has been shown [5, 39] that it is close to optimal to choose $N = 1$. Although the variance reduction in the density estimates for larger N generally gives higher acceptance rates and improved mixing, the gain in the number effective samples in this case is usually smaller than the increased computational cost per update.

As noted in [39] in many practical settings cases the assumption of a linear increase in cost with the number of importance samples N will not be valid, particularly for small N. For example most modern *central processing units* (CPUs) have some degree of parallel compute capability through multiple cores so (assuming the parallelism can be exploited) there will usually be a non-linear increase in cost until all cores are at full utilisation: a rough guideline in this case is to use one sample per core. Another situation in which the linear cost assumption may not hold is when there is a high fixed computational overhead in each density estimate independent of the number of samples. For example if a importance distribution is used which is dependent on the target variables there may be computational operations such as matrix decompositions that can be performed once and then their cost amortised over generation of multiple importance samples.

Particle filtering estimators do not take the form of a simple Monte Carlo average of independent unbiased estimates but are instead are formed as a product of (dependent) Monte Carlo estimates [39]. The result of [39] that using $N = 1$ is close to optimal (with N now the number of particles) is therefore not applicable in this case.

Under an alternative simplifying assumption relevant to the particle filtering setting that the noise in the logarithm of the density estimator is normally distributed and independent of the value of the target variables $\boldsymbol{x}$ and that the computational cost of each density estimate scales linearly with N, it is argued in [11] that N should be chosen so as to make the standard deviation of the logarithm of the density estimator approximately equal to 1.2. In [40] a more specific case is considered of pseudo-marginal Metropolis–Hastings methods using a isotropic Gaussian random-walk Metropolis proposal $r(\boldsymbol{x}' \mid \boldsymbol{x}) = \mathcal{N}\left(\boldsymbol{x}' \mid \boldsymbol{x}, \lambda^2 \mathbf{I}\right)$ and the same assumptions as [11] made of additive normal noise in the logarithm of the density estimator which is independent of $\boldsymbol{x}$ and a computational cost for each density estimate which scales linearly with N. It is shown that for target distributions on a D dimensional space which obey certain regularity assumptions as $D \to \infty$ that computational efficiency is maximised for a choice of λ and N which gives an average accept rate of approximately 0.07 and a noise standard deviation for the logarithm of the density estimator of approximately 1.8.

3.3 REPARAMETRISING THE DENSITY ESTIMATOR

As a first step in considering how to apply alternative transition operators to pseudo-marginal inference problems, we define a reparameterisation of the density estimator in terms of a deterministic function of the auxiliary random variables used in computing the estimate. An equivalent reparameterisation has also been used in other work analysing the pseudo-marginal framework, for example [11].

In general the computation of a density estimate will involve sampling values from known distributions using a pseudo-random number generator and then applying a series of deterministic operations to these auxiliary random variables. Under the simplifying assumption that the estimator uses a fixed number of auxiliary random variables, we can therefore define a non-negative deterministic function $\varepsilon : X \times U \to [0, \infty)$ and a distribution R with known density $\rho = \frac{\partial R}{\partial v}$ such that if $\boldsymbol{u}$

is an independent sample from R, then $\hat{p} = \varepsilon(\boldsymbol{x}, \boldsymbol{u})$ is an independent sample from $\mathrm{P}_{\hat{\mathrm{p}}|\mathbf{x}}(\cdot \mid \boldsymbol{x})$. Here R represents the known distribution of the auxiliary variables and ε the operations performed by the remaining estimator code given values for the target and auxiliary variables. We can use this to reparameterise (3.2) as

$$\tilde{p}(\boldsymbol{x}) = \int_U \varepsilon(\boldsymbol{x}, \boldsymbol{u})\, R(\mathrm{d}\boldsymbol{u}) = \int_U \varepsilon(\boldsymbol{x}, \boldsymbol{u})\, \rho(\boldsymbol{u})\, \nu(\mathrm{d}\boldsymbol{u}) \quad \forall \boldsymbol{x} \in X. \tag{3.6}$$

For example considering the importance-sampling density estimator for a hierachical latent variable model defined in (3.4), if we assume the importance distribution is chosen to be a multivariate normal with density $\mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{x},\boldsymbol{y}}, \Sigma_{\boldsymbol{x},\boldsymbol{y}}\right)$ then defining $\boldsymbol{u} = \left[\boldsymbol{u}^{(1)}; \cdots; \boldsymbol{u}^{(n)}\right]$ as the concatenated vector of standard normal variables used to generate the importance distribution samples, we have $\rho(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{u} \mid \mathbf{0}, \boldsymbol{I})$ and

$$\varepsilon(\boldsymbol{x}, \boldsymbol{u}) = \frac{1}{N} \sum_{n=1}^{N} \frac{\mathrm{p}_{\mathbf{x},\mathrm{y},\mathbf{z}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{L}_{\boldsymbol{x},\boldsymbol{y}} \boldsymbol{u}^{(n)} + \boldsymbol{\mu}_{\boldsymbol{x},\boldsymbol{y}})}{\mathcal{N}\left(\boldsymbol{L}_{\boldsymbol{x},\boldsymbol{y}} \boldsymbol{u}^{(n)} + \boldsymbol{\mu}_{\boldsymbol{x},\boldsymbol{y}} \mid \boldsymbol{\mu}_{\boldsymbol{x},\boldsymbol{y}}, \Sigma_{\boldsymbol{x},\boldsymbol{y}}\right)}, \tag{3.7}$$

where $\boldsymbol{L}_{\boldsymbol{x},\boldsymbol{y}}$ is the lower triangular Cholesky factor of $\Sigma_{\boldsymbol{x},\boldsymbol{y}}$.

Rather than defining the chain state in the pseudo-marginal Metropolis–Hastings update as the target state – density estimate pair $(\boldsymbol{x}, \hat{p})$, we can instead replace the density estimate $\hat{p}$ with the auxiliary random variables $\boldsymbol{u}$ drawn from R used to compute the estimate. As $\hat{p}$ is a deterministic function of $\boldsymbol{x}$ and $\boldsymbol{u}$ these two parameterisations are equivalent. The implementation in Algorithm 8 can be considered a practically motivated variant that avoids the $\boldsymbol{u}$ values needing to be stored in memory and in fact means they do not need to be explicitly defined in the algorithm at all.

While the formulation of the update in Algorithm 8 is the more useful for implementation purposes, showing the correctness of the update is simpler when considering the chain state as $(\boldsymbol{x}, \boldsymbol{u})$. We will briefly go through this derivation now as it provides some useful insights in to the pseudo-marginal Metropolis–Hastings algorithm that will help motivate our alternative proposed approaches.

From (3.6) we known that a distribution on $X \times U$ with density

$$\pi(\boldsymbol{x}, \boldsymbol{u}) = \frac{1}{C}\, \varepsilon(\boldsymbol{x}, \boldsymbol{u})\, \rho(\boldsymbol{u}) \tag{3.8}$$

will have the target distribution on X as its marginal distribution. Showing that the transition operator defined by Algorithm 8 leaves a distribution with density corresponding to (3.8) invariant is therefore sufficient for ensuring the correctness of the algorithm.

The transition operator corresponding to Algorithm 8 has a density

$$\mathrm{t}(\boldsymbol{x}', \boldsymbol{u}' \mid \boldsymbol{x}, \boldsymbol{u}) = r(\boldsymbol{x}' \mid \boldsymbol{x})\rho(\boldsymbol{u}')\alpha(\boldsymbol{x}', \boldsymbol{u}' \mid \boldsymbol{x}, \boldsymbol{u}) + \delta(\boldsymbol{x} - \boldsymbol{x}')\,\delta(\boldsymbol{u} - \boldsymbol{u}') \left(1 - \int_U \int_X r(\boldsymbol{x}' \mid \boldsymbol{x})\rho(\boldsymbol{u}')\alpha(\boldsymbol{x}', \boldsymbol{u}' \mid \boldsymbol{x}, \boldsymbol{u})\,\mu(\mathrm{d}\boldsymbol{x})\nu(\mathrm{d}\boldsymbol{u})\right),$$

with the accept probability α being defined here as

$$\alpha(\boldsymbol{x}', \boldsymbol{u}' \mid \boldsymbol{x}, \boldsymbol{u}) = \min\left\{1, \frac{r(\boldsymbol{x} \mid \boldsymbol{x}')\varepsilon(\boldsymbol{x}', \boldsymbol{u}')}{r(\boldsymbol{x}' \mid \boldsymbol{x})\varepsilon(\boldsymbol{x}, \boldsymbol{u})}\right\}. \tag{3.9}$$

As in Chapter 2 it is sufficient to show the non self-transition term in this transition density satisfies detailed balance with respect to the target density (3.8) as self-transitions leave any distribution invariant. We have that for $\boldsymbol{x} \neq \boldsymbol{x}'$, $\boldsymbol{u} \neq \boldsymbol{u}'$

$$\begin{aligned}
&\mathrm{t}(\boldsymbol{x}', \boldsymbol{u}' \mid \boldsymbol{x}, \boldsymbol{u})\,\pi(\boldsymbol{x}, \boldsymbol{u}) \\
&\quad = \frac{1}{C}\, r(\boldsymbol{x}' \mid \boldsymbol{x})\, \rho(\boldsymbol{u}')\, \alpha(\boldsymbol{x}', \boldsymbol{u}' \mid \boldsymbol{x}, \boldsymbol{u})\, \varepsilon(\boldsymbol{x}, \boldsymbol{u})\, \rho(\boldsymbol{u}) \\
&\quad = \frac{1}{C}\, \rho(\boldsymbol{u}')\, \rho(\boldsymbol{u})\, \min\{r(\boldsymbol{x}' \mid \boldsymbol{x})\, \varepsilon(\boldsymbol{x}, \boldsymbol{u}), r(\boldsymbol{x} \mid \boldsymbol{x}')\, \varepsilon(\boldsymbol{x}', \boldsymbol{u}')\} \\
&\quad = \frac{1}{C}\, r(\boldsymbol{x} \mid \boldsymbol{x}')\, \rho(\boldsymbol{u})\, \alpha(\boldsymbol{x}, \boldsymbol{u} \mid \boldsymbol{x}', \boldsymbol{u}')\, \varepsilon(\boldsymbol{x}', \boldsymbol{u}')\, \rho(\boldsymbol{u}') \\
&\quad = \mathrm{t}(\boldsymbol{x}, \boldsymbol{u} \mid \boldsymbol{x}', \boldsymbol{u}')\, \pi(\boldsymbol{x}', \boldsymbol{u}'),
\end{aligned} \tag{3.10}$$

and so the transition operator corresponding to Algorithm 8 leaves the target distribution invariant.

We can equivalently consider Algorithm 8 as a standard Metropolis–Hastings transition operator on a target distribution with density (3.8) using a proposal $r(\boldsymbol{x}' \mid \boldsymbol{x})\rho(\boldsymbol{u}')$ i.e. perturbatively updating the $\boldsymbol{x}$ values and indepedently resampling the $\boldsymbol{u}$ values. Substituting this proposal density and target density into the standard Metropolis–Hastings accept ratio recovers the form used in the pseudo-marginal variant,

$$\frac{r(\boldsymbol{x} \mid \boldsymbol{x}')\rho(\boldsymbol{u})\frac{1}{C}\,\varepsilon(\boldsymbol{x}', \boldsymbol{u}')\,\rho(\boldsymbol{u}')}{r(\boldsymbol{x}' \mid \boldsymbol{x})\rho(\boldsymbol{u}')\frac{1}{C}\,\varepsilon(\boldsymbol{x}, \boldsymbol{u})\,\rho(\boldsymbol{u})} = \frac{r(\boldsymbol{x} \mid \boldsymbol{x}')\varepsilon(\boldsymbol{x}', \boldsymbol{u}')}{r(\boldsymbol{x}' \mid \boldsymbol{x})\varepsilon(\boldsymbol{x}, \boldsymbol{u})}. \tag{3.11}$$

Algorithm 9 Auxiliary pseudo-marginal framework.

Input: $(\boldsymbol{x}_n, \boldsymbol{u}_n)$: current target variables – auxiliary variables pair, T_1 : transition operator updating only auxiliary variables $\boldsymbol{u}$ and leaving distribution with density in (3.8) invariant, T_2 : transition operator updating only target variables $\boldsymbol{x}$ and leaving distribution with density in (3.8) invariant.

Output: $(\boldsymbol{x}_{n+1}, \boldsymbol{u}_{n+1})$: new target state – auxiliary variables pair.

$\boldsymbol{u}_{n+1} \sim \mathsf{T}_1(\cdot \mid \boldsymbol{x}_n, \boldsymbol{u}_n)$
$\boldsymbol{x}_{n+1} \sim \mathsf{T}_2(\cdot \mid \boldsymbol{x}_n, \boldsymbol{u}_{n+1})$
return $(\boldsymbol{x}_{n+1}, \boldsymbol{u}_{n+1})$

Algorithm 10 Auxiliary pseudo-marginal MI + MH.

Input: $(\boldsymbol{x}_n, \boldsymbol{u}_n)$: current target – auxiliary variables state pair, ε : estimator function for density of target distribution, ρ : density of estimator's auxiliary variable distribution, r : proposal density for updates to target state.

Output: $(\boldsymbol{x}_{n+1}, \boldsymbol{u}_{n+1})$: new target – auxiliary variables state pair.

$\boldsymbol{u}^* \sim \rho(\cdot)$ ▷ T_1: MI update to auxiliary variables.
$v \sim \mathcal{U}(\cdot \mid 0, 1)$
if $v < \frac{\varepsilon(\boldsymbol{x}_n, \boldsymbol{u}^*)}{\varepsilon(\boldsymbol{x}_n, \boldsymbol{u}_n)}$ **then**
    $\boldsymbol{u}_{n+1} \leftarrow \boldsymbol{u}^*$
else
    $\boldsymbol{u}_{n+1} \leftarrow \boldsymbol{u}_n$
$\boldsymbol{x}^* \sim r(\cdot \mid \boldsymbol{x}_n)$ ▷ T_2: MH update to target variables.
$w \sim \mathcal{U}(\cdot \mid 0, 1)$
if $w < \frac{r(\boldsymbol{x}_n \mid \boldsymbol{x}^*)\, \varepsilon(\boldsymbol{x}^*, \boldsymbol{u}_{n+1})}{r(\boldsymbol{x}^* \mid \boldsymbol{x}_n)\, \varepsilon(\boldsymbol{x}_n, \boldsymbol{u}_{n+1})}$ **then**
    $\boldsymbol{x}_{n+1} \leftarrow \boldsymbol{x}^*$
else
    $\boldsymbol{x}_{n+1} \leftarrow \boldsymbol{x}_n$
return $(\boldsymbol{x}_{n+1}, \boldsymbol{u}_{n+1})$

This formulation highlights a potential source of some of the computational issues with the pseudo-marginal Metropolis–Hastings algorithm. In high-dimensional spaces generally we would expect independent resampling of a subset of the variables in a Markov chain state from their marginal distribution for a proposed Metropolis–Hastings move to perform poorly [31]. Unless the variables being independently resampled have little or no dependency on the rest of the chain state, the marginal distribution will be significantly different from the conditional distribution given the remaining variables and proposed values from the marginal will be often be highly atypical under the conditional and so have a low probability of acceptance.

3.4 AUXILIARY PSEUDO-MARGINAL METHODS

The observation that the pseudo-marginal Metropolis–Hastings update corresponds to a special case of the standard Metropolis–Hastings al-

gorithm with independent proposed updates to the auxiliary random variables suggests the possibility of using alternative transition operators within a pseudo-marginal context. A particularly simple framework is to alternate updates to the target state $\boldsymbol{x}$ given the auxiliary variables $\boldsymbol{u}$ and to the auxiliary variables $\boldsymbol{u}$ given the target state $\boldsymbol{x}$. We refer to this scheme as the *auxiliary pseudo-marginal* (APM) framework and summarise it in Algorithm 9.

A simple example of an APM method is formed by alternating *Metropolis independence* (MI) updates to the auxiliary variables given the target variables using R as the proposal distribution with *Metropolis–Hastings* (MH) updates to the target variables given the current auxiliary variables; this variant is described in Algorithm 10. Following the convention of [30] we name this method APM MI+MH for short and will in general use the form APM [T1]+[T2] to name APM methods where [T1] and [T2] are abbreviations for the types of the transition operators T_1 and T_2 respectively.

The APM MI+MH method retains the black-box nature of the original *pseudo-marginal* (PM) MH algorithm by requiring no explicit knowledge of the auxiliary random variables used in the density estimate providing we can read and write the internal state of the *pseudo-random number generator* (PRNG) used by the estimator. This can be achieved for example using the `.Random.seed` attribute in `R` and the `get_state` and `set_state` methods of a `NumPy` `RandomState` object. We then only need to store the PRNG state associated with each target density estimator evaluation and restore a previous state if we wish to estimate the density at a new target state with the same set of auxiliary variables as used for a previous evaluation.

Any PM MH implementation can easily be converted in to a APM MI+MH method as the two algorithms require exactly the same input objects with the APM MI+MH method simply splitting the original single MH step into two separate propose-accept steps. The APM MI+MH method introduces some overhead by requiring two new evaluations of the target density estimator per overall update (once for the new proposed auxiliary variables and once for the new proposed target variables) compared to the single evaluation required for the PM MH algorithm.

Importantly however the updates to the target variables in APM MI+MH take the form of a standard perturbative MH update. If we use a random-

walk Metropolis update then this means we can automatically tune the step size of the updates by for example appealing to theoretical results suggesting tuning the step size to achieve an average acceptance rate of 0.234 is optimal (in terms of maximising the number of effective samples per computation time) when making perturbative moves in high-dimensions [15]. The tuning can either be done in an initial warm-up phase of the chain with the samples from this initial phase not included in the final Monte Carlo estimates or by using online approaches which use vanishing adaptation [3, 18].

As discussed earlier for particle filtering estimators, under certain simplifying assumptions an alternative average acceptance rate of 0.07 has shown to be optimal for PM MH with a isotropic normal random-walk proposal in high-dimensional target distributions [40]. While this does provide a target for tuning the step-size of a standard PM MH update in the cases where it is relevant, the APM MI+MH update may often be easier to tune in practice. The 0.07 target accept rate is predicated on the variance of the density estimator having been tuned, via the number of Monte Carlo samples, such that log density estimates have a standard deviation of approximately 1.8. In general tuning the density estimator variance can be non-straightforward as in real problems it will typically vary depending on $\boldsymbol{x}$ and it is not clear which value or values to use to measure the variance at, potentially requiring an additional preliminary run to find a suitable $\boldsymbol{x}$ value to tune at. Further the non-constant estimator variances found in practice will tend to give an accept rate which varies in mean and variance across the target space. This gives a noisy signal for adaptive algorithms to tune the step-size by, potentially requiring slow adaptation for stability.

In contrast the APM MI+MH method decouples the MI auxiliary updates, which have an acceptance rate controlled by the variance of the density estimate[2] and so N, and the MH target variables updates which have an acceptance rate which is controlled by the proposal step-size λ. The two distinct accept rates provide independent signals to tune the two free parameters N and λ by, and which individually will generally be less noisy than the single combined accept rate of the PM MH update.

2 During the MI update to the auxiliary variables the target variables $\boldsymbol{x}$ are held fixed and a proposed new set of auxiliary variable values $\boldsymbol{u}^*$ and so density estimate $\hat{p}^* = \varepsilon(\boldsymbol{x}, \boldsymbol{u}^*)$ independently sampled. If the variance of the density estimate tends to zero the ratio of $\hat{p}^*$ to the previous estimate $\hat{p}$ which determines the accept probability of the MI step tends to one.

In density estimators which are simple Monte Carlo averages and the cost of the estimator scales linearly with the number of Monte Carlo samples N such that the results of [39] apply and a choice of $N = 1$ close to optimal, the additional signal provided by the accept rate of the MI updates to the auxiliary variables is of less direct relevance. However as noted previously, in practice often the linear estimator cost assumption will not hold for small N, due to utilisation of parallel computation or high fixed costs. In these cases we may still wish to use the MI accept rate to adjust N so that the accept rate is above some lower threshold: although a low N (and so high estimator variance and low MI step accept probability) may be preferable in the asymptotic regime as the number of samples tends to infinity, in practical settings with finite length chains it can be that an overly high density estimator variance can lead to very low accept rates for the auxiliary variable updates such that in a finite length chain the number of updates to the auxiliary variables is very low (or even zero), potentially leading to biases in the marginal distributions of the sampled target variables.

3.5 PSEUDO-MARGINAL SLICE SAMPLING

Rather than using a MH update to the target variables, the APM framework also makes it simple to apply alternative transition operators to pseudo-marginal inference problems. A particularly appealing option are the linear and elliptical *slice sampling* (SS) algorithms discussed in Chapter 2 (Algorithms 4 and 5); when combined with MI updates to the auxiliary variables we term such methods APM MI+SS. Slice sampling algorithms automatically adapt the scale of proposed moves and so will generally require less tuning than random-walk Metropolis to achieve reasonable performance and also cope better in target distributions where the geometry of the density and so appropriate scale for proposed updates varies across the target variable space.

Slice sampling updates will always lead to a non-zero move of the target variables on each update providing for fixed values of the auxiliary variables the estimator function ε is a smooth function of the target variables. In such cases APM MI+SS chains will not show the 'sticking' artifacts in the traces of the target variables common to PM MH chains. As the auxiliary variables are still being updated using Metropolis independence transitions however they will still be susceptible to having

proposed moves rejected so the accept rate (and traces if available) of the auxiliary variables updates should also be monitored to check for convergence issues.

The APM MI+MH and APM MI+SS methods although offering advantages over the standard PM MH method do not address the issue that proposing new auxiliary variable values for fixed values of the target variables independent of the previous auxiliary variable values can perform poorly in high dimensions. Even weak dependence between the auxiliary variables and target variables will mean that in high-dimensions the typical set of the marginal auxiliary variable distribution R used as the proposal distribution will differ significantly from the typical set of the conditional distribution on the auxiliary variables given the target variables values used to decide acceptances and so the accept probability of proposed updates to the auxiliary variables will be small.

One way of increasing the probability of proposed updates to the auxiliary variables from R being accepted is to increase the number of Monte Carlo samples N used in the estimator. For concreteness we will assume we use the importance sampling estimator (3.4) for inference in a hierarchical latent variable model with a multivariate normal importance distribution $q(\boldsymbol{z} \mid \boldsymbol{x}, \boldsymbol{y}) = \mathcal{N}\left(\boldsymbol{z} \mid \boldsymbol{\mu}, \boldsymbol{L}\boldsymbol{L}^{\mathsf{T}}\right)$ (in general $\boldsymbol{\mu}$ and $\boldsymbol{L}$ will depend on $\boldsymbol{x}$ and $\boldsymbol{y}$ but we leave this dependence implicit for notational simplicity). Using the reparametrisation of the estimator in (3.7), the target density (3.8) on the auxiliary and target variables takes the form

$$\pi(\boldsymbol{x}, \boldsymbol{u}) = \frac{1}{NC} \sum_{n=1}^{N} \frac{\mathrm{p}_{\mathbf{x},\mathbf{y},\mathbf{z}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{L}\boldsymbol{u}^{(n)} + \boldsymbol{\mu})}{\mathcal{N}\left(\boldsymbol{L}\boldsymbol{u}^{(n)} + \boldsymbol{\mu} \mid \boldsymbol{\mu}, \boldsymbol{L}\boldsymbol{L}^{\mathsf{T}}\right)} \prod_{n=1}^{N} \mathcal{N}\left(\boldsymbol{u}^{(n)} \mid \mathbf{0}, \mathbf{I}\right). \tag{3.12}$$

Using that $C = \mathrm{p}_{\mathbf{y}}(\boldsymbol{y})$ and $\mathcal{N}\left(\boldsymbol{L}\boldsymbol{u} + \boldsymbol{\mu} \mid \boldsymbol{\mu}, \boldsymbol{L}\boldsymbol{L}^{\mathsf{T}}\right) = |\boldsymbol{L}|^{-1}\mathcal{N}(\boldsymbol{u} \mid \mathbf{0}, \mathbf{I})$ this can be manipulated into the form

$$\pi(\boldsymbol{x}, \boldsymbol{u}) = \frac{\mathrm{p}_{\mathbf{x}|\mathbf{y}}(\boldsymbol{x} \mid \boldsymbol{y})}{N|\boldsymbol{L}|^{-1}} \sum_{n=1}^{N} \frac{\mathrm{p}_{\mathbf{z}|\mathbf{x},\mathbf{y}}(\boldsymbol{L}\boldsymbol{u}^{(n)} + \boldsymbol{\mu} \mid \boldsymbol{x}, \boldsymbol{y})}{\mathcal{N}\left(\boldsymbol{u}^{(n)} \mid \mathbf{0}, \mathbf{I}\right)} \prod_{n=1}^{N} \mathcal{N}\left(\boldsymbol{u}^{(n)} \mid \mathbf{0}, \mathbf{I}\right).$$

By separating out the terms involving a single auxiliary variable sample $\boldsymbol{u}^{(m)}$, the conditional density on $\boldsymbol{u}^{(m)}$ given the remaining auxiliary variable samples can be shown to take the form of a mixture

$$\begin{aligned}\pi\left(\boldsymbol{u}^{(m)} \mid \boldsymbol{x}, \{\boldsymbol{u}^{(n)}\}_{n \neq m}\right) \propto \\ \mathrm{p}_{\mathbf{z}|\mathbf{x},\mathbf{y}}(\boldsymbol{L}\boldsymbol{u}^{(m)} + \boldsymbol{\mu} \mid \boldsymbol{x}, \boldsymbol{y}) + w\left(\boldsymbol{x}, \{\boldsymbol{u}^{(n)}\}_{n \neq m}\right)\mathcal{N}\left(\boldsymbol{u}^{(m)} \mid \mathbf{0}, \mathbf{I}\right)\end{aligned} \tag{3.13}$$

$$\text{with } w\left(\boldsymbol{x}, \{\boldsymbol{u}^{(n)}\}_{n\neq m}\right) = \sum_{n\neq m}\left(\frac{\mathrm{p}_{\mathbf{z}|\mathbf{x},\mathbf{y}}(\boldsymbol{L}\boldsymbol{u}^{(n)} + \boldsymbol{\mu} \mid \boldsymbol{x}, \boldsymbol{y})}{\mathcal{N}\left(\boldsymbol{u}^{(n)} \mid \mathbf{0}, \mathbf{I}\right)}\right).$$

The sum of the importance weights in w will grow with N (for independent $\boldsymbol{u}^{(n)} \sim \mathcal{N}(\cdot \mid \mathbf{0}, \mathbf{I})$ $\forall n \neq m$ it would have an expected value $(N-1)|\boldsymbol{L}|$) and so for large N the second term in the mixture will increasingly dominate and the conditional density on $\boldsymbol{u}^{(m)}$ will tend to $\mathcal{N}\left(\boldsymbol{u}^{(m)} \mid \mathbf{0}, \mathbf{I}\right)$ and independence from $\boldsymbol{x}$. Therefore as we increase N we would expect independently re-sampling the auxiliary variables from R in a MI step to have an increasing probability of acceptance.

Although non-rigorous, this analysis also gives an intution to why the pseudo-marginal method can provide an advantage over directly performing MCMC in the joint space of $\mathbf{x}$ and $\mathbf{z}$ in hierachical latent variable models: if the conditional density on the local latent variables $\mathrm{p}_{\mathbf{z}|\mathbf{x},\mathbf{y}}$ has a challenging geometry, for example it is multimodal, then MCMC transition operators based on local moves working in the $(\mathbf{x}, \mathbf{z})$ are likely to mix poorly for example by getting stuck in a single mode or only being able to make very small moves per update. If we instead reparameterise in terms of a set of auxiliary variables $\{\boldsymbol{u}^{(n)}\}_{n=1}^{N}$, then we are able to maintain the correct marginal distribution on the target variables $\mathbf{x}$ while working with a distribution on an extended space which becomes increasingly tractable to sample from as we increase N, with the individual auxiliary variable samples $\boldsymbol{u}^{(n)}$ individually having conditional densities which only weakly depend on $\mathrm{p}_{\mathbf{z}|\mathbf{x},\mathbf{y}}$.

While we can always increase N to the point where independently proposing updates to the auxiliary variables from R will have a reasonable probability of acceptance, this will also increase the computational expense of each update. Rather than proposing new values for the auxiliary variables independently of their previous values, an obvious idea is to take a more standard MCMC approach by using local perturbative updates which leave the overall target distribution (3.8) invariant. For $N = 1$ this equivalent to performing MCMC directly in a non-centred reparameterisation [33] of the joint $(\mathbf{x}, \mathbf{z})$ space by alternating updates of the target and auxiliary (latent) variables. For $N > 1$ we potentially gain from the conditional distribution on the auxiliary variables being easier for MCMC algorithms to explore though with an increased computational cost per update.

One option is to use a MH method such as random-walk Metropolis to update the auxiliary variables. While with a well tuned proposal distribution this approach could work well, it adds further tuning burden to the user which might outweigh any efficiency gains. For problems in which we can reparametrise the density estimator as a deterministic function of a vector of standard normal draws so that $\rho(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{u} \mid \mathbf{0}, \mathbf{I})$, an appealing option is to use elliptical slice sampling (Algorithm 5) to update the auxiliary variables. The elliptical slice sampling algorithm has no free parameters for the user to choose and initially proposes moves to points nearly independent of the current values [28] so if the conditional distribution of the auxiliary variables is well approximated by the normal marginal distribution R, elliptical slice sampling should perform similarly to a MI update. Using the adaptive bracket shrinking procedure discussed in Chapter 2 the elliptical slice sampler is also however able to exponentially back-off to smaller proposed moves around the current state if the bold initial proposal is not on the slice. Providing for fixed values of the target variables the target density (3.8) is a smooth function of the auxiliary variables, the slice sampling procedure will always lead to a non-zero update of the auxiliary variables.

If the auxiliary variables are instead marginally distributed as independent standard uniform variables[3] i.e. $\rho(\boldsymbol{u}) = \prod_i \mathcal{U}(u_i \mid 0, 1)$, one option is to reparameterise these as independent standard normal variables which are then mapped through the normal *cumulative distribution function* (CDF). We can then run elliptical slice sampling in the transformed normal space. In general evaluation of the normal CDF is a relatively expensive operation and the distortion induced by pushing through the CDF may in some case map a distribution with a density with relatively simple geometry in the uniform space to a density with more complex geometry in the normal space. An alternative is to therefore perform linear slice sampling directly in the uniform auxiliary variable space.

A small subtlety is that the target distribution on the auxiliary variables will only have support on the unit hypercube in this case. We can adjust Algorithm 4 for this setting by replacing Line 8 in the Algorithm with $\boldsymbol{x}^* \leftarrow \textsc{Reflect}(\boldsymbol{x}_n + \lambda \boldsymbol{v})$ (and the likewise the corresponding equival-

3 The auxiliary variables in this case could for example represent all the standard uniform draws from the PRNG that are used to generate random variables in the estimator using the rejection and transform sampling routines discussed in Chapter 2.

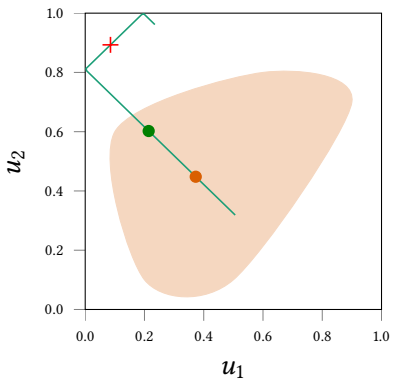


Figure 3.2: Illustration of reflective linear slice sampling in two dimensions. The orange circular marker represents the current state and the light filled orange region the density slice at the sampled slice height (see explanation of Algorithm 4 in Chapter 2 for details). A random slice line direction vector $\boldsymbol{v}$ is sampled from some distribution as in Algorithm 4, for example with elements independently sampled from $\mathcal{N}(0,1)$ or $\mathcal{U}(-1,1)$. This defines a line passing through the current point (green-blue line in Figure), with importantly in this case the line *reflected* at the boundaries of the hypercube (square in this two-dimensional case). An initial bracket of a specified width is randomly placed around the current point on the line. The algorithm then proceeds as in the standard linear slice sampling algorithm by repeatedly proposing a point in the current bracket and accepting if on the slice (in orange region, for example the green circle) or rejecting and shrinking the bracket if off the slice (outside orange region, for example the red cross).

ent expressions in the step-out routine in Lines 17 and 20), where the REFLECT function is defined elementwise by

---

function REFLECT(u)
  $v \leftarrow u \bmod 2$
  return $v\, \mathbb{1}_{[0,1)}(v) + (2 - v)\, \mathbb{1}_{[1,2)}(v)$

The reflection transformation defined by this function has a unit Jacobian determinant and maintains reversibility and so the reflective slice sampling transition leaves the uniform distribution on the slice invariant. An illustrative schematic of a reflective linear slice sampling transition in two dimension is shown in Figure 3.2. Reflective variants of slice sampling are discussed in [32] and [12].

3.6 NUMERICAL EXPERIMENTS

We will now discuss the results of two empirical studies in to the performance of the proposed auxiliary pseudo-marginal methods. Further experiments applying some of the proposed methods in a simulator model inference setting will be discussed in Chapter 4.

3.6.1 Gaussian latent variable model

As a first numerical example we consider inference in a hierarchical Gaussian latent variable model. In particular we assume a model with the factorisation structure shown in Figure 3.1 with

$$\begin{aligned} \mathrm{p}_{\mathbf{x}}(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x} \mid \mathbf{0}, \mathbf{I}), \quad \mathrm{p}_{\mathbf{z}|\mathbf{x}}(\boldsymbol{z} \mid \boldsymbol{x}) = \prod_{m=1}^{M} \mathcal{N}\left(\boldsymbol{z}^{(m)} \mid \boldsymbol{x}, \sigma^2 \mathbf{I}\right), \\ \text{and} \quad \mathrm{p}_{\mathbf{y}|\mathbf{x},\mathbf{z}}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{z}) = \prod_{m=1}^{M} \mathcal{N}\left(\boldsymbol{y}^{(m)} \mid \boldsymbol{z}^{(m)}, \epsilon^2 \mathbf{I}\right). \end{aligned} \tag{3.14}$$

We used $\sigma = 1$ and $\epsilon = 2$ in the experiments and generate $M = 10$ simulated observed values $\{\boldsymbol{y}^{(m)}\}_{m=1}^{M}$, each of dimensionality $D = 10$. We assume we wish to infer plausible values for the D-dimensional vector $\mathbf{x}$ consistent with the observed $\mathbf{y}$ and so the target distribution for inference has density $p(\boldsymbol{x}) = \mathrm{p}_{\mathbf{x}|\mathbf{y}}(\boldsymbol{x} \mid \boldsymbol{y})$. Here because of the self-conjugacy of the Gaussian distribution, the marginalisation over the local latent variables $\mathbf{z}$ can be performed analytically to give

$$\mathrm{p}_{\mathbf{x}|\mathbf{y}}(\boldsymbol{x} \mid \boldsymbol{y}) = \mathcal{N}\left(\boldsymbol{x} \,\middle|\, \frac{1}{M + \sigma^2 + \epsilon^2} \sum_{m=1}^{M} \boldsymbol{y}^{(m)}, \frac{\sigma^2 + \epsilon^2}{N + \sigma^2 + \epsilon^2} \mathbf{I}\right). \tag{3.15}$$

Although exact inference is therefore tractable in this case, we apply pseudo-marginal MCMC methods to allow us to study the performance of the methods in a case where we have a ground-truth for the inferences to check convergence against.

We use an importance sampling estimator of the form given in (3.4) using $\mathrm{P}_{\mathbf{z}|\mathbf{x}}$ as the importance distribution i.e.

$$q\left(\{\boldsymbol{z}^{(m)}\}_{m=1}^{M} \mid \boldsymbol{x}, \{\boldsymbol{y}^{(m)}\}_{m=1}^{M}\right) = \prod_{m=1}^{M} \mathcal{N}\left(\boldsymbol{z}^{(m)} \mid \boldsymbol{x}, \sigma^2 \mathbf{I}\right). \tag{3.16}$$

As this importance distribution does not take in to account the observed values $\{\boldsymbol{y}^{(m)}\}_{m=1}^{M}$ it results in a relatively high-variance importance

sampling estimator of the density with a variance which depends on the values of the target variables $\boldsymbol{x}$. Therefore although exact inference in this example is tractable and the target distribution has a simple isotropic geometry, in this pseudo-marginal formulation the model still has some of the key features which can pose challenges to pseudo-marginal inference algorithms.

For the auxiliary pseudo-marginal methods, we use a reparameterisation of the estimator equivalent to (3.7), using the standard normal variables used to generate samples from $\mathsf{P}_{\mathbf{z}|\mathbf{x}}$ as the auxiliary variables, resulting in an auxiliary variable marginal distribution with density $\rho(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{u} \mid \mathbf{0}, \mathbf{I})$ and an estimator function ε

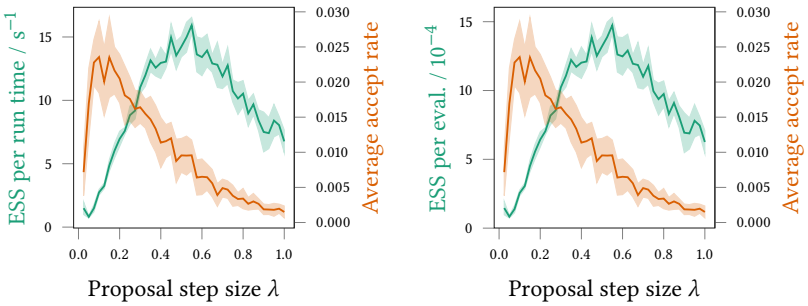
$$\varepsilon(\boldsymbol{x}, \boldsymbol{u}) = \frac{\mathcal{N}(\boldsymbol{x} \mid \mathbf{0}, \mathbf{I})}{N} \sum_{n=1}^{N} \prod_{m=1}^{M} \mathcal{N}\left(\boldsymbol{y}^{(m)} \mid \sigma \boldsymbol{u}^{(n,m)} + \boldsymbol{x}, \epsilon^2 \mathbf{I}\right), \tag{3.17}$$

with $\boldsymbol{u} = \left[\boldsymbol{u}^{(1,1)}; \ldots \boldsymbol{u}^{(1,M)}; \boldsymbol{u}^{(2,1)} \ldots \boldsymbol{u}^{(N,M)}\right] \in \mathbb{R}^{NM}$.

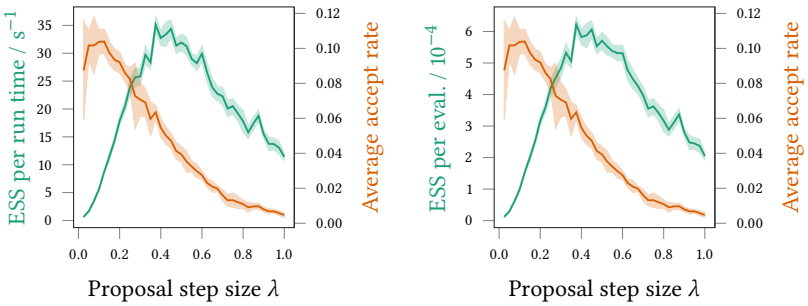
Pseudo-marginal Metropolis–Hastings

We first applied the PM MH algorithm to perform inference in this model, using an isotropic normal random-walk proposal distribution for the updates to the target variables, i.e. $r(\boldsymbol{x}' \mid \boldsymbol{x}) = \mathcal{N}\left(\boldsymbol{x}' \mid \boldsymbol{x}, \lambda^2 \mathbf{I}\right)$. To assess the impact of the choice of the proposal step size parameter λ on sampling efficiency, we ran 10 independent chains initialised from the prior $\mathcal{N}(\mathbf{0}, \mathbf{I})$ for λ values on a equispaced grid of 40 points between 0.025 and 1, running each chain for 50 000 iterations. We ran all experiments for the cases of density estimators using $N = 1$, $N = 8$ and $N = 32$ importance samples, with the logarithm of the density estimate at the value of the target variables $\boldsymbol{x}$ used to generate the observed values $\boldsymbol{y}$ having standard deviation 3.6 for $N = 1$, 1.8 for $N = 8$ and 1.2 for $N = 32$.

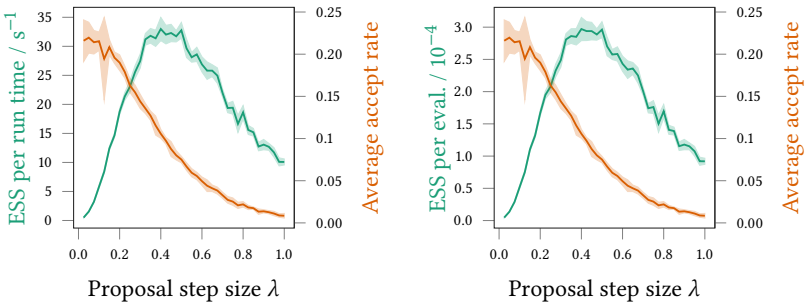
For all combinations of N and λ we estimated the *effective sample size* (ESS) (as defined for a geometrically ergodic Markov chain in Equation 2.25 of Chapter 2) for the posterior mean of each chain using the `R CODA` package [35]. We then derived two overall measures of computational efficiency from these ESS estimates by normalising either by the number of joint density evaluations in the density estimator (which increases per iteration with the number of importance samples N) or the wall clock run time of the chains in seconds. The results are plotted



(a) $N = 1$



(b) $N = 8$



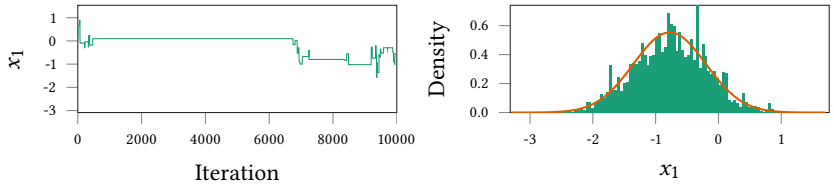
(c) $N = 32$

Figure 3.3: Results of Gaussian latent variable model PM MH chains. The plots in each row show both the estimated ESS normalised by either the compute time (green, left column) or number of density estimator evaluations (green, right column) and average acceptance rate of MH updates (orange), versus the isotropic random-walk proposal step-size λ for the MH updates to the target variables. The top row shows the case for a density estimator using $N = 1$ importance sample, middle row for $N = 8$ and the bottom row for $N = 32$. In all cases the curves show mean values across 10 independent chains initialised from the prior and filled region show ±1 standard deviation.

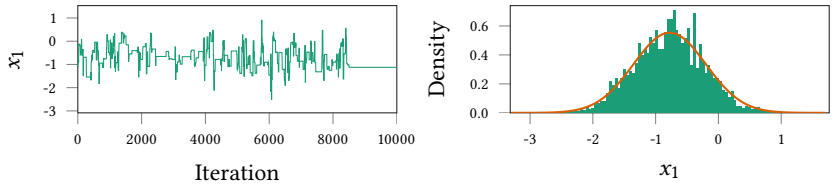
in Figure 3.3. Each pair of plots in a row corresponds to a particular number of importance samples. In each row the left column shows the ESSs normalised by the run time and the right column by the number of density evaluations, with the green curves representing the mean of these values across all the chains and the filled region plus and minus one standard deviation (note the standard deviation rather than standard error of mean was used as in some of the plots the standard error was too small to be easily visible). On each axis as well as the normalised ESS, the average accept rate across the chains is also plotted in orange (with scale shown on the right vertical axis), with again the curves showing the mean value across the chains and the filled regions plus and minus one standard deviation.

The results of [39] suggest that asymptotically using $N = 1$ importance sample should be optimal in this case assuming a linear increase in the cost of generating each sample with N. The measure of computational efficiency used in [39] therefore most closely corresponds to the estimated ESS normalised by the number of density evaluations (which scales linearly with N), and indeed on this measure (green curves in right column of Figure 3.3) we see that the chains using $N = 1$ outperforms the $N = 8$ and $N = 32$ cases.

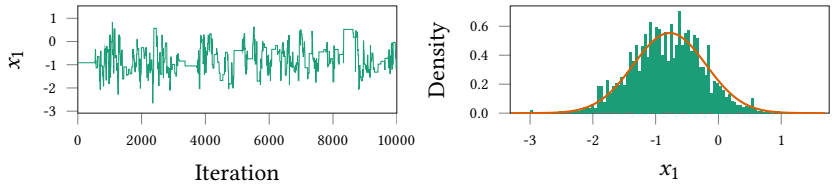
The plots in Figure 3.3a and to a lesser extent 3.3b show a spurious appearing behaviour for the smallest step sizes that the accept rate (orange curve) seems to initially *increase* as the step size is made larger, contrary to what we would reasonably expect. This anomaly can be ascribed to a lack of convergence in the chains with small step sizes due to the sticking behaviour discussed previously. For the $N = 1$ case, because of the relatively high density estimator variance, the chains are prone to getting stuck for thousands of iterations at a time. The estimator variance is dependent on the values of the target variables $\boldsymbol{x}$ and generally seems to be lower for values typical under the posterior. As the chains are initialised from the prior, they tend to therefore initialised in regions in which the estimator variance is higher than typical often leading to long sticking periods near the start of the chain. For the chains with small step sizes the chain is slower to 'warm-up' and converge towards the typical set of the posterior distribution on the target variables and so this propensity for sticking during the initial warm-up period has a larger effect, leading to some chains rejecting nearly all updates even though the step size is very small. This counter



(a) $N = 1, \lambda = 0.550$



(b) $N = 8, \lambda = 0.375$



(c) $N = 32, \lambda = 0.400$

Figure 3.4: Example traces and histograms of PM MH chains in Gaussian latent variable model inference task. In each row a trace of the sampled values for the x_1 target variable in the last 10000 iterations of a PM MH Markov chain using the optimal step size for the relevant N found from Figure 3.3 is shown in the left plot, while the right plot shows a histogram of the samples values from the full chain (green filled region) against the exact marginal posterior density (orange curve). In the histogram plots the number of samples in the chain used to produce the plot have been adjusted to account for the increased number of density evaluations for higher N, so the $N = 1$ plot is of a chain of 3.2×10^6 samples, the $N = 8$ plot is of 8×10^5 samples as the $N = 32$ plot of 1×10^5 samples.

intuitive behaviour of the empirical accept rates for small step sizes and general noisiness of the dependency of the accept rate on the step size, particularly for small N, highlights the difficulty of tuning the PM MH updates: the low accept rates here would intuitively indicate the step size should be made smaller but in some cases this would actually make the measured accept rate even worse.

Figure 3.4 shows example traces of the x_1 variable samples for chains using density estimates with $N = 1$, $N = 8$ and $N = 32$ importance samples. In each case the step size suggested to be optimal by the results in Figure 3.3 (in terms of effective samples per density evaluation) has been used, and the traces shown are the last 10 000 iterations of a longer run. Also shown are histogram estimates of the posterior marginal densities on the x_1 variable using the sampled states from the whole chain, with the total number of samples in each chain adjusted to account for the extra computational cost of using more importance samples, along with a curve showing the true posterior marginal density. The propensity of the chains to stick is clearly visible in the traces particularly for the $N = 1$ case, with long series of thousands of rejected updates at a time. This is also reflected in the noisiness of the marginal density estimates with spurious peaks appearing around the states where the chain gets stuck.

When comparing instead in terms of the estimated ESS normalised by actual chain run time (green curves in left column of Figure 3.3) the results no longer suggest $N = 1$ is optimal, with the $N = 8$ and $N = 32$ cases both performing better on this measure for all step sizes. This can be explained by the non-linear scaling of the computational cost per update with the number of importance samples due to both overhead from the implementation of the rest of the operations in the transition and only partial utilisation of the parallel compute resource available (the CPU used in the experiments had 4 cores). Although the increase in efficiency per actual run time for $N \neq 1$ is implementation and device dependent, a possibly stronger reason suggested by the results to use $N > 1$ is the less brittle nature of the chains behaviour, with the very low accept rates in the $N = 1$ case needing long runs to smooth out the effects of long series of rejections.

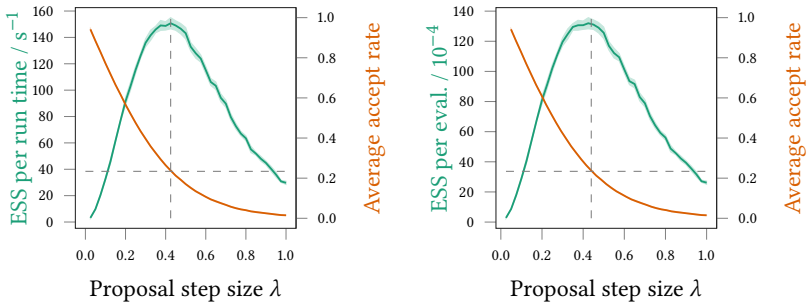
The results in Figure 3.3 also highlight the difficulty of tuning the proposal step size when using a random-walk Metropolis PM MH update. The optimal step size appears to possibly weakly depend on the num-

ber of importance samples used (though the noisiness of the curves make this difficult to determine). Further there is not a clear relationship between the average accept rate and optimal step size. As previously stated the result of [15] that a step size giving an accept rate of 0.234 is close to optimal is not applicable to the update here, with this confirmed empirically by the fact that only the chains with the smallest step sizes for the $N = 32$ case are even able to achieve an accept rate close to 0.234 (and are far from optimal in efficiency). In practice we therefore we do not have an obvious signal to tune the step size by beyond running pilot chains and computing ESS estimates which is likely to add too much cost to justify any gain in efficiency from choosing a better step size for subsequent chains.

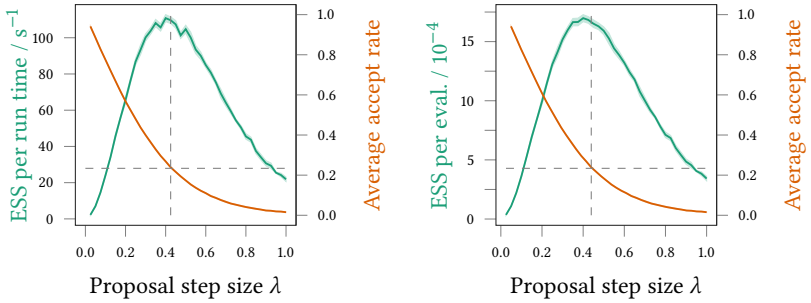
Splitting the update

We next applied the proposed APM MI+MH algorithm to perform inference in the Gaussian latent variable model. From an implementation perspective this simply requires the original combined update to the auxiliary and target variables in the PM MH case to be split in to separate MI updates of the auxiliary variables given fixed target variables and MH updates of the target variables for fixed auxiliary variables. Despite the seemingly minor change to the form of the update, the difference in the results is dramatic.

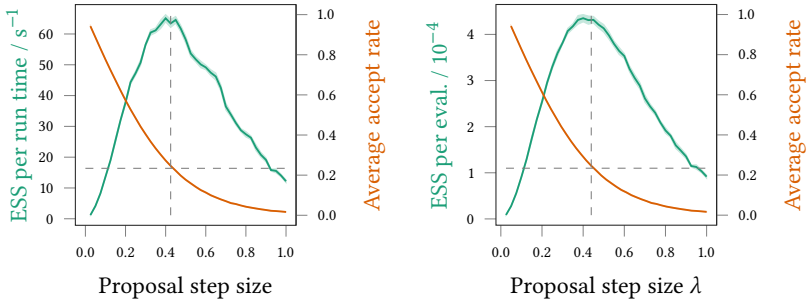
Figure 3.5 shows plots of results of an equivalent series of experiments as used to produce Figure 3.3. In this case the horizontal axes on the plots shows the proposal step size for the MH updates to the target variables which as previously use an random-walk Metropolis proposal $r(\boldsymbol{x}' \mid \boldsymbol{x}) = \mathcal{N}\left(\boldsymbol{x}' \mid \boldsymbol{x}, \lambda^2 \mathbf{I}\right)$. Again 10 independent chains initialised from the prior were run for each step size λ and number of importance samples N pair, with in this case shorter chains of 20 000 iterations used (with the known posterior means and standard deviations used to establish that the chains had adequately converged). Again the estimated ESSs for estimates of the posterior mean were computed for each chain, with the green curves in the left column of plots in Figure 3.5 showing the mean of these estimated ESSs across the chains normalised by the total wall clock run time for the chain, and the right column the ESSs normalised by the number of joint density evaluations. The average accept rate shown by the orange curves in Figure 3.5 is for the MH update to the target variables. A separate average accept rate was



(a) $N = 1$



(b) $N = 8$



(c) $N = 32$

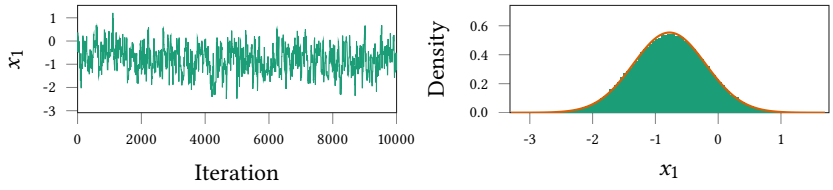
Figure 3.5: Results of Gaussian latent variable model APM MI+MH chains. The plots in each row show both the estimated ESS normalised by either the total compute time (green, left column) or number of density estimator evaluations (green, right column) and average acceptance rate for the MH updates (orange), versus the isotropic random-walk proposal step-size for the MH updates to the target variables. The top row shows the case for a density estimator using $N = 1$ importance sample and the bottom row for $N = 8$. The top row shows the case for a density estimator using $N = 1$ importance sample, middle row for $N = 8$ and the bottom row for $N = 32$. In all cases the curves show mean values across 10 independent chains initialised from the prior and filled region show ±1 standard deviation. The horizontal dashed lines indicate an accept rate of 0.234 and the vertical dashed lines the corresponding proposal step size.

recorded for the MI updates to the auxiliary variables and was found to not show any obvious dependency on the target variable proposal step size, with an average accept rate of approximately 0.025 for chains with $N = 1$ importance sample in the density estimates, an average accept rate of 0.11 for chains with $N = 8$ and an average accept rate of 0.23 for chains using $N = 32$.

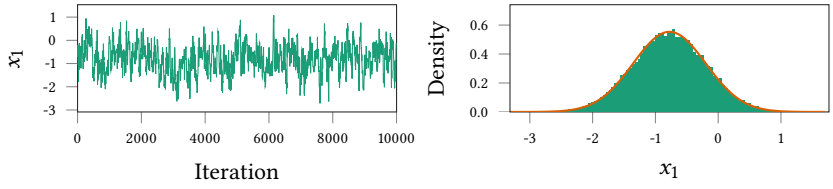
On both the time and density evaluation normalised measures of efficiency the APM MI+MH chains perform significantly better than the PM MH chains. The peak ESS per density evaluation value for the $N = 1$ and $\lambda = 0.425$ case is around a factor of ten higher than the corresponding peak value for the PM MH chains, while in terms of the ESS per run time metric the best APM MI+MH chains show around a factor four improvement over the PM MH chains. While other experiments have suggested this level of improvement is atypical, it seems reasonable to conclude that at least in some cases the extra overhead introduced by requiring two density estimates per overall update is worthwhile.

More importantly perhaps the curves in Figure 3.5 suggest the APM MI+MH update is signicantly easier to tune. The average accept rate of the MH updates to the target variables shows the expected monotonically decreasing behaviour as the step size is increased and in general the measured accept rates are significantly less noisy than the corresponding accept rates for the PM MH updates. The horizontal dashed lines in Figure 3.5 indicate an average accept rate of 0.234 with the corresponding vertical dashed lines showing the estimated proposal step size corresponding to this acceptance rate. As can be seen by both the compute time and density evaluation normalised measures of sampling efficiency, the chains with proposal step sizes giving accept rates near to 0.234 are close to optimal in efficiency, suggesting the theoretical result of [15] holds here as suggested earlier. Further in this model at least, this relationship seems to hold for a range of different numbers of importance samples and so density estimator variances. This suggests it is valid to use standard adaptive approaches which use the average accept rate as a control signal to tune the step size of the target variables MH proposal distribution when using the APM MI+MH update.

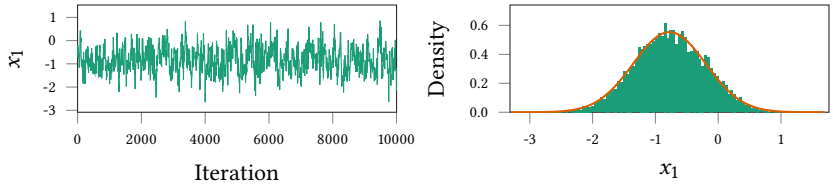
In further contrast to the PM MH results, the results for the APM MI+MH chains seem to unambiguously support using $N = 1$ importance sample. On both the computation time and density evaluation normalised measures of efficiency, the chains using one importance sample dominate



(a) $N = 1, \lambda = 0.425$



(b) $N = 8, \lambda = 0.425$



(c) $N = 32, \lambda = 0.425$

Figure 3.6: Example traces and empirical histograms of APM MI+MH chains in Gaussian latent variable model inference task. In each row a trace of the sampled values for the x_1 target variable in the last 10 000 iterations of a APM MI+MH Markov chain using the optimal step size for the relevant N found from Figure 3.5 is shown in the left plot, while the right plot shows an empirical histogram of the samples values from the full chain (green filled region) against the exact marginal posterior density (orange curve). In the histogram plots the number of samples in the chain used to produce the plot have been adjusted to account for the increased number of density evaluations for higher N and to account for the 2 evaluations per update compared to PM MH to allow fair comparison with Figure 3.4, so the $N = 1$ plot is of a chain of 1.6×10^6 samples, the $N = 8$ plot is of 2×10^5 samples as the $N = 32$ plot of 5×10^4 samples.

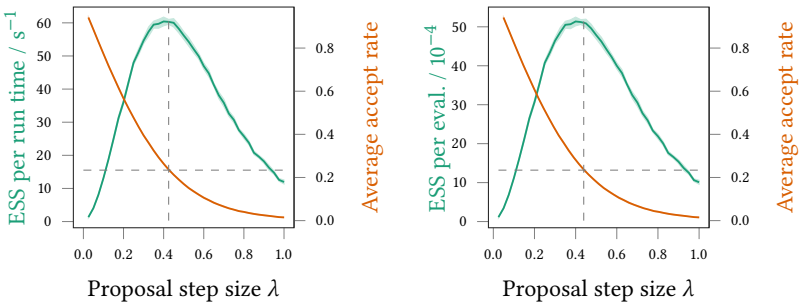


Figure 3.7: Results of Gaussian latent variable model APM SS+MH chains (using $N = 1$ importance sample). The plots in each row show both the estimated ESS normalised by either the total compute time (green, left column) or number of density estimator evaluations (green, right column) and average acceptance rate for the MH updates (orange), versus the isotropic random-walk proposal step-size λ for the MH updates to the target variables. The curves show mean values across 10 independent chains initialised from the prior and filled region show ±1 standard deviation.

over the $N = 8$ and $N = 32$ cases. The APM MI+MH chains using a single importance sample do not show the pathological sticking behaviour evident in the PM MH chains, with an example trace shown for a step size of $\lambda = 0.425$ (which Figure 3.5 suggests is close to optimal) in Figure 3.6a. Unlike the $N = 1$ PM MH trace, over the 10 000 iterations shown the APM MI+MH seems to mix well with no obvious sticking periods. The example traces for the $N = 8$ and $N = 32$ APM MI+MH chains in Figure 3.6 also seem to follow this pattern. Comparing the posterior marginal density estimates for the x_1 target variable shown in the right column of Figure 3.6, the marginal estimates for the $N = 1$ case appear the smoothest, almost indistinguishable from the curve of the true density (to normalise for the additional density evaluations required for the $N = 8$ and $N = 32$ cases the number of samples in the chains used to produce the histograms was reduced accordingly). This again suggests that any improvement in mixing by using $N > 1$ in this case is outweighed by the cost of the additional density evaluations.

Slice sampling the auxiliary variables

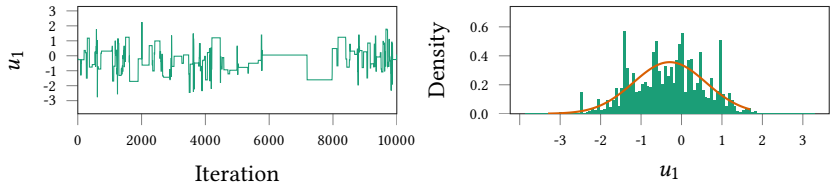
For the APM MI+MH chains discussed in the previous subsection, when using $N = 1$ importance sample the MI updates to the auxiliary variables were only accepted 2.5% of the time. Although this did not appear to impede convergence of the chain in this example, more generally low accept rates for the MI updates to the auxiliary variables may be

a cause for concern as in shorter chains this will mean the auxiliary variables are only updated a small number of times across the chain. As convergence of the distribution on the target variables in the chain state to their marginal target distribution is reliant on the distribution of the auxiliary variables in the chain state also converging, very infrequent updates of the auxiliary variables could potentially lead to difficult to diagnose convergence issues in the chains. Although increasing the number of importance samples in the estimator can increase the MI step accept rate as seen in the APM MI+MH experiments above, there is a diminishing returns behaviour to the increase of acceptance rate with the number of samples.

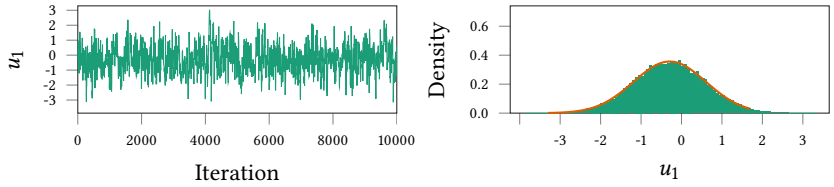
The earlier suggestion to use perturbative updates to the auxiliary variables provides an alternative approach to improve the auxiliary variable mixing. We test specifically here the proposal to use elliptical slice sampling updates to the auxiliary variables, which is a natural choice in this case due to their Gaussian marginal distribution. We use the same MH update to the target variables as in the experiments in the previous two subsections, and again measure sampling efficiency for different proposal step sizes λ. We only run chains using a estimator taking $N = 1$ importance sample in this case as we are mainly interested in using perturbative updates to the auxiliary variables as an alternative to having to increase the number of importance samples to achieve reasonable acceptance rates for MI updates to the auxiliary variables.

Results for an equivalent series of experiments as discussed in the previous two subsections for APM SS+MH chains using elliptical slice sampling updates to the auxiliary variables are shown in Figure 3.7. In this case as the MI updates to the auxiliary variables for the $N = 1$ case seemed to be sufficient to achieve convergence, the elliptical slice sampling updates do not seem to significantly improve mixing of the target variables. The extra overhead from the adaptive slice sampling updates means overall computational efficiency decreases by roughly a factor of two across all proposal step sizes λ compared to the corresponding APM MI+MH results for $N = 1$ in Figure 3.5a, with this consistent across both the density evaulation normalised efficiency metric and run time normalised measure.

Although the slice sampling updates do not help improve the sampling of the target variables here, the resulting auxiliary variables samples are much more representative of their true posterior distribution (which

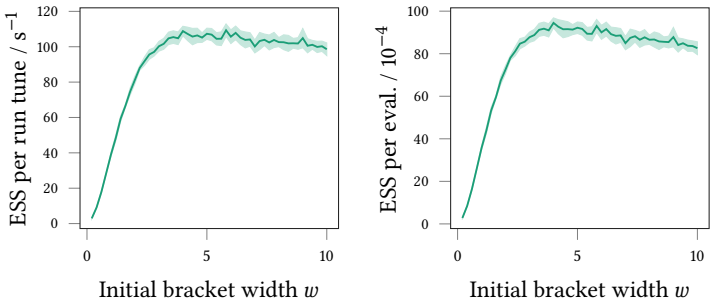


(a) APM MI+MH $N = 1, \lambda = 0.425$

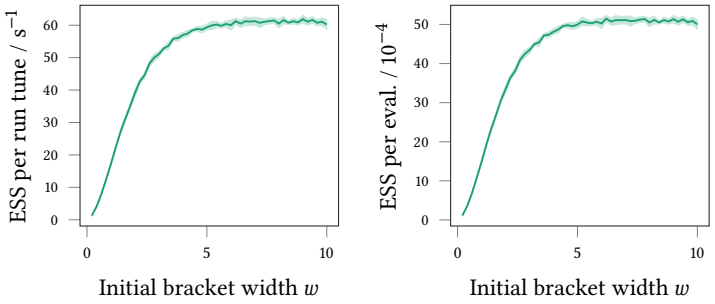


(b) APM SS+MH $N = 1, \lambda = 0.425$

Figure 3.8: Example traces and histograms of an auxiliary variable in APM MI+MH and APM SS+MH chains in Gaussian latent variable model inference task. In each row a trace of the sampled values for the u_1 auxiliary variable in the last 10000 iterations of a Markov chain using the optimal step size $\lambda = 0.425$ and $N = 1$ is shown in the left plot, while the right plot shows a histogram of the sample values from the full chain (green filled region) against the exact marginal posterior density (orange curve). In the histogram plots the number of samples in the chain used to produce the plot have been adjusted to account for the roughly two times increase in the number of density evaluations per sample for the APM SS+MH updates compared to APM MI+MH, so the APM MI+MH plot is of a chain of 10^5 samples and the APM SS+MH plot is of 5×10^4 samples.



(a) APM MI+SS ($N = 1$)



(b) APM SS+SS ($N = 1$)

Figure 3.9: Results of Gaussian latent variable model APM chains using linear SS to update target variables and either MI updates to auxiliary variables (top row) or elliptical SS updates (bottom row). The plots in each row show both the estimated ESS normalised by either the total compute time (left column) or number of density estimator evaluations (right column), versus the slice sampler initial bracket width for the linear SS updates to the target variables. The curves show mean values across 10 independent chains initialised from the prior and filled region show ±1 standard deviation.

again can be found analytically) compared to when using MI updates. Figure 3.8 shows traces and histograms of one of the auxiliary variables for chains computed using both the APM MI+MH and APM SS+MH updates. The slice sampling updates give significantly better mixing of the auxiliary variables than the MI updates which due to the low accept rate remain fixed for many iterations. Although in this case this does seem to translate to an obvious improvement in convergence of the target variables, more generally a factor two increase in run time for the added robustness of significantly improved mixing of the auxiliary variables seems like it will often be a worthwhile tradeoff to avoid possible convergence issues.

Slice sampling the target variables

As a final set of experiments for this model, we explored the use of slice sampling updates to the target variables with an auxiliary pseudo-marginal framework, specifically linear slice sampling updates along an isotropically sampled direction. To test the claim that the efficiency of slice sampling updates is less sensitive to the choice of the free initial bracket width parameter w of the algorithm than random-walk Metropolis updates are to the choice of the proposal step size parameter λ, we ran a similar series of experiments as in the previous sub-sections to analyse the dependency of sampling efficiency on λ by instead varying the initial bracket width w.

For each of 50 initial bracket width w values on an equispaced grid between 0.2 and 10, we ran 10 independent APM MI+SS and APM SS+SS chains (with elliptical slice sampling updates to the auxiliary variables) initialised from the prior of 20 000 iterations each. As previously for each set of chains for a particular w value we computed the estimated ESSs of the chains for the estimate of the posterior mean and normalised this value by both the total wall-clock run time in seconds and total number of joint density evaluations to give two measures of overall efficiency. The means and one standard deviation intervals of these values across the 10 chains are shown for the APM MI+SS chains in Figure 3.9a and for the APM SS+SS chains in Figure 3.9b. In all cases zero linear step-out iterations were used in the slice sampling updates to the target variables.

The peak efficiency achieved by the APM MI+SS chains on this problem is less than that for the best APM MI+MH chains by a factor of around 1.5 on both measures of efficiency. As the slice sampling updates do more work per iteration than the MH updates this is not unexpected as a well-tuned MH update will generally perform better than a slice sampling update when the geometry of the target distribution is simple (as is the case here). Importantly however the slice sampling updates maintain a computational efficiency that is within around 10% of the optimal efficiency across a wide range of intial bracket width values, with values from $w = 2$ to $w = 10$ all seeming to perfom reasonably well in this problem. This is in contrast to the much tighter range of proposal step size values required to get good performance with MH updates to the target variables. The exponential back-off to smaller proposals provided by the adaptive bracket shrinking procedure in the slice sampling trans-

ition means that the penalty for using an overly large scale parameter w is much less severe than the corresponding situation for using an overly large λ in a MH update.

The results for the APM SS+SS show a similar pattern compared to the APM SS+MH results, except for that the best APM SS+MH chains perform almost identically to the best APM SS+SS chains. This is due to the extra overhead introduced by the elliptical slice sampling updates to the auxiliary variables having a larger effect than the slightly more efficient updates to the target variables by the optimally tuned MH updates in this case. This also explains the even slower drop-off in efficiency for the APM SS+SS for large values of the initial bracket width w, with the extra density evaluations this requires on average having a smaller overall effect on efficiency due to the higher baseline number of evaluations because of the elliptical slice sampling updates.

3.6.2 Gaussian process probit regression

As a second experiment we consider a more challenging problem of inferring the parameters of the covariance function of a latent Gaussian process used to model the relationship between pairs of feature vectors and binary target outputs. The use of PM MH for this task was considered in [14] and shown to give signficant improvements over competing MCMC methods.

As an example data set we used the Wisconsin breast cancer prediction data set [24] from the UCI machine learning dataset repository [21] as also used for experiments in [14]. The data $\{\boldsymbol{d}^{(m)}, y_m\}_{m=1}^{M}$ consists of pairs of vectors $\boldsymbol{d}^{(m)}$ of $K = 9$ integer descriptors of individual cells found in a fine needle aspiration biopsy of suspect breast lumps, and a binary class y_m indicating whether the lump was later found to malignant or benign. The original dataset contains 699 data-points, however 17 data-points have missing atrributes so $M = 682$ data-points were used in the experiments here.

To model the unknown relationship between the input descriptors and binary class label output, a zero-mean Gaussian process prior [38] was placed on a set of latent real-valued function values $\mathbf{z} \in \mathbb{R}^M$. A squared exponential covariance function was used with per-feature length scales $\boldsymbol{\ell} \in \mathbb{R}_{>0}^K$ and output scale $\mathrm{s} \in \mathbb{R}_{>0}$, with the covariance function specifically defined as

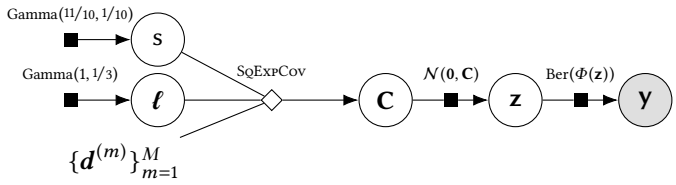


Figure 3.10: Gaussian process probit regression factor graph.

function SqExpCov($\{\boldsymbol{d}^{(m)}\}_{m=1}^{M}, \mathrm{s}, \boldsymbol{\ell}, \epsilon = 10^{-8}$)
  for $i \in \{1 \ldots M\}$ **do**
    $\mathrm{C}_{i,i} \leftarrow \mathrm{s} + \epsilon$
    for $j \in \{1 \,..\, j-1\}$ **do**
      $\mathrm{C}_{i,j} \leftarrow \mathrm{s} \exp\left(-\frac{1}{2}\sum_{k=1}^{D}\left(\frac{d_k^{(i)} - d_k^{(j)}}{\ell_d}\right)^2\right)$
      $\mathrm{C}_{j,i} \leftarrow \mathrm{C}_{i,j}$
  return C

The ϵ value is a 'jitter' parameter to improve numerical stability [38]. This covariance functions represents an assumption that nearby $\boldsymbol{d}^{(m)}$ points correspond to similar $\mathbf{z}$ values, with the typical length-scales in the feature space over which correlations are high determined by the elements of $\boldsymbol{\ell}$. The latent variables $\mathbf{z}$ are assumed to determine the probability of the observed binary class outputs $\mathbf{y}$ being one or zero by a probit link function i.e. given $\mathbf{z} = z$ the binary outputs are modelled as having a Bernoulli distribution $\mathrm{Ber}(\Phi(z))$ where Φ is the standard normal CDF function. Following [14] Gamma prior distributions were placed on both the per-feature length-scales $\boldsymbol{\ell}$ and output scale s covariance function parameters. The overall model is shown as a directed factor graph in Figure 3.10.

For inference we assume we are interested in inferring the posterior distribution on the $\boldsymbol{\ell}$ and s covariance function parameters given the observed input-output pairs, such that we could then use the inferred plausible $\boldsymbol{\ell}$ and s values to make predictions of the outputs corresponding to unlabelled inputs. We define the target variables for inference $\mathbf{x}$ as the logarithms of $\boldsymbol{\ell}$ and s so that the target distribution has support on an unbounded space i.e. $\mathbf{x} = [\log \mathrm{s}; \log \boldsymbol{\ell}]$ and $\mathbf{x} \in \mathbb{R}^{10}$, with a Jacobian determinant factor accounting for the change of variables being included in the transformed prior (marginal) density $\mathrm{p}_{\mathbf{x}}$

$$\mathrm{p}_{\boldsymbol{x}}(\boldsymbol{x}) \propto \exp\left(\frac{11x_1}{10} - \frac{\exp(x_1)}{10}\right)\prod_{i=2}^{10}\exp\left(x_i - \frac{\exp(x_i)}{3}\right). \tag{3.18}$$

The unnormalised target density is then $\tilde{p}(\boldsymbol{x}) = \mathrm{p}_{\mathbf{x},\mathbf{y}}(\boldsymbol{x}, \boldsymbol{y}) = \mathrm{p}_{\mathbf{y}|\mathbf{x}}(\boldsymbol{y} \mid \boldsymbol{x})$. We cannot evaluate $\mathrm{p}_{\mathbf{y}|\mathbf{x}}$ as it involves an intractable marginalisation over the latent function values $\mathbf{z}$

$$\mathrm{p}_{\mathbf{y}|\mathbf{x}}(\boldsymbol{y}|\boldsymbol{x}) = \int_Z \prod_{m=1}^{M} \left(\Phi(z_m)^{y_m}(1-\Phi(z_m))^{1-y_m}\right) \mathcal{N}(z \mid \mathbf{0}, \mathbf{C})\, \mathrm{d}z. \quad (3.19)$$

One option would be to construct a Markov chain on the joint $(\mathbf{x}, \mathbf{z})$ space with an unnormalised target density $\mathrm{p}_{\mathbf{x},\mathbf{y},\mathbf{z}}$, however strong dependencies between the (transformed) covariance function parameters $\mathbf{x}$ and the latent variables $\mathbf{z}$ makes the joint distribution difficult for MCMC dynamics to explore effectively [14]. As an alternative [14] proposes to use the pseudo-marginal framework to construct a Markov chain using an unbiased importance sampling estimator of $\tilde{p}$.

Though a Monte Carlo estimate of (3.19) can be formed by sampling latent values $\mathbf{z}$ from the Gaussian proccess prior $\mathrm{p}_{\mathbf{z}|\mathbf{x}}$, as this ignores the observed output values $\boldsymbol{y}$ this will tend to lead to a density estimator with unusably high variance for the purposes of use in a pseudo-marginal update. A key insight in [14] was that much lower variance density estimates can be computed by using an optimisation-based approximate inference method to fit a Gaussian approximation to $\mathrm{p}_{\mathbf{z}|\mathbf{x},\mathbf{y}}$ (which as discussed previously is the optimal choice for the importance distribution in terms of minimising variance) to use as the importance distribution. In [14] both Laplace's method and expectation propagation are considered within this context; we concentrate on Laplace's method here for simplicity.

As discussed in Appendix C, Laplace's method involves finding the mode of the density being approximated and then evaluating the Hessian matrix of the log density at this point. An efficient and numerically stable implementation of a Newton–Raphson method can be used to find the mode of the latent posterior for this probit regression Gaussian process model [38, §3.4] with the latent posterior density guaranteed to have a unique mode. Each Newton–Raphson step involves computing a Cholesky factorisation of the Hessian of the log density at the current point which has a $O(M^3)$ computational cost. In the experiments around 10 Newton steps were needed to achieve convergence when finding the mode. Evaluating the density of the Gaussian process prior on the latent function values $\mathbf{z}$ also requires computing a Cholesky decomposition of the Gaussian process covariance matrix which again

has $O(M^3)$ cost. As $M = 682$ these cubic cost operations will tend to be the dominant contributor to the overall run time. As the Gaussian process covariance and Laplace approximation to the latent posterior both depend on the value of the covariance function parameters, the cubic operations have to be performed each time a density estimate is computed at a new value for the target variables.

Once an approximate Gaussian latent posterior $\mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{x},\boldsymbol{y}}, \boldsymbol{\Sigma}_{\boldsymbol{x},\boldsymbol{y}}\right)$ has been fitted using Laplace's method, it can then be used as the importance distribution in an importance sampling estimator of the form shown in (3.4). The Cholesky factorisation $\boldsymbol{L}_{\boldsymbol{x},\boldsymbol{y}} = \text{chol}\, \boldsymbol{\Sigma}_{\boldsymbol{x},\boldsymbol{y}}$ is computed as part of the Laplace's method iteration, and so can be reused to efficiently evaluate the importance distribution density at a $O(M^2)$ cost for each importance sample and to generate samples from the importance distribution using $\boldsymbol{z}^{(n)} = \boldsymbol{L}_{\boldsymbol{x},\boldsymbol{y}}\boldsymbol{u}^{(n)} + \boldsymbol{\mu}_{\boldsymbol{x},\boldsymbol{y}}$ where $\boldsymbol{u}^{(n)}$ is a sampled standard normal vector from $\mathcal{N}(\mathbf{0}, \mathbf{I})$, this again having a $O(M^2)$ cost. This same expression can also be used to reparameterise the estimator as a deterministic function of a set of independent standard normal values $\boldsymbol{u} = \left[\boldsymbol{u}^{(1)}; \boldsymbol{u}^{(2)} \ldots \boldsymbol{u}^{(N)}\right]$ as shown previously in (3.7) and as required for the proposed auxiliary pseudo-marginal methods.

Due to the high overhead of the cubic operations the result of [39] that a choice of $N = 1$ is close to optimal does not apply here. In experiments in [39] with a similar Gaussian process classifier model (in their case using a logistic link function and using a dataset with $M = 144$) it was found computational efficiency was approximately maximised by using $N = 200$ importance samples with it noted that this is around the number required for the $O(M^2N)$ cost of sample generation to be of comparable magnitude to the cubic operation cost. In the example of [39] a non-iterative approach is used to find a Gaussian importance distribution hence only a single Cholesky decomposition of the importance distribution covariance matrix is required. The use of an iterative Laplace method approximation for the importance distribution here as proposed in [14] makes it unclear whether a similar choice of the number of importance samples is reasonable here. While the even higher overhead of the multiple cubic operations per estimator evaluation supports possibly using $N \geq M$, part of the justification of using an expensive procedure to fit the importance distribution is that it means fewer importance samples are needed to achieve a low-variance density estimator. In the experiments with the same dataset in [14] $N = 1$ import-

ance sample was used and found to work well, though in that case an isotropic covariance function was used with a single length scale parameter such that the dimensionality of the target space was two rather than ten as here.

In preliminary runs we found that the PM MH update had very low accept rates however small we set the proposal step-size when using $N = 1$ importance sample in the density estimator. Increasing the number of importance samples to $N = 50$ gave a signficant improvement in performance and overall stability with a negligible increase in run time per update. Increasing the number of importance samples further to $N = 500$ gave a further increase in efficiency but also increased the run time in our implmentation by around one third which outweighed the per iteration sampling efficiency gains made. We therefore used $N = 50$ importance samples for the main experiments with all methods; given the limited number of values tested this is unlikely to be optimal but in most practical situations we would be unlikely to perform an exhaustive search for the optimal N. Interestingly the auxiliary pseudo-marginal methods appeared to still be able to mix when using $N = 1$ importance sample with APM MI+MH chains still able to achieve a target accept rate of $\sim 0.15 - 0.3$ for the MH updates to the target variables. Due to the negligible increase in run time however when using $N = 50$ importance samples we performed the experiments for the APM methods with $N = 50$ also.

We generated Markov chains for the model using each of PM MH, APM MI+MH, APM SS+MH and APM SS+SS for the updates. For the MH updates to the target variables in the first three methods we used a Gaussian random-walk Metropolis proposal distribution $r(\boldsymbol{x}' \mid \boldsymbol{x}) = \mathcal{N}\left(\boldsymbol{x}' \mid \boldsymbol{x}, \lambda^2 \mathbf{I}\right)$. To set the proposal step size λ we followed the adaptive approach used in [14], with the step size adjusted over an initial warm-up phase of 2000 iterations, with the average accept rate over every 100 iterations used as a control signal to decide whether to increase or decrease the step size. Also following [14] a target average accept rate range of $[0.15, 0.3]$ was used[4], with the step size made smaller or larger, if the average accept rate is below or above this range respectively during the adaptive phase. As noted in the previous experiments, while a target rate of 0.234 for the MH updates to the target variables in APM methods can be

4 Although in the published version of [14] it is stated a target range of $[0.2, 0.3]$ was used, the code accompanying the paper suggests a range of $[0.15, 0.3]$ was used in the experiments so we follow that instead.

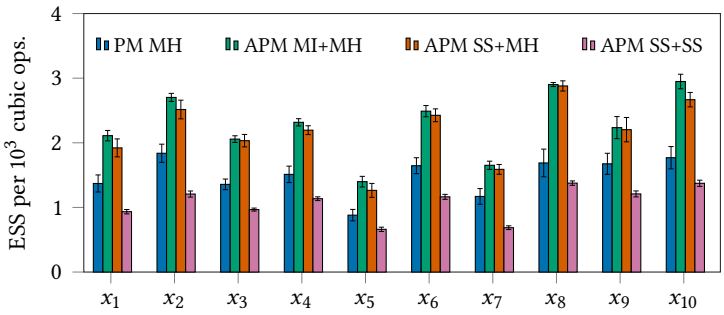
justified theoretically and empirically, it is not clear what the optimal choice is for PM MH updates, with this seeming to be dependent on the estimator variance and so number of importance samples N. While the $[0.15, 0.3]$ target accept rate range therefore seems reasonable for the APM methods it is unclear whether it is a good choice for the pseudo-marginal method, however as it was used with some success in [14] and given a lack of obvious alternative methods for choosing the target rate, we use it for the PM MH updates here.

For the APM SS+MH and APM SS+SS methods we used elliptical slice sampling for the updates to the auxiliary variables. For the slice sampling update to the target variables in the APM SS+SS chains we used linear slice sampling along a random direction vector (sampled isotropically) with a fixed initial bracket width of $w = 4$ and no linear step out iterations. To account for the 2000 adaptive warm up iterations performed before the main PM MH, APM MI+MH and APM SS+MH chains, for the APM SS+SS chains we ran 1000 warm up iterations before the main chain runs. For all four methods, 10 chains independently initialised from the prior were run for 10 000 iterations, with both the total number of cubic operations performed and overall run time recorded to allow for adjustment for different per iteration costs in the results.

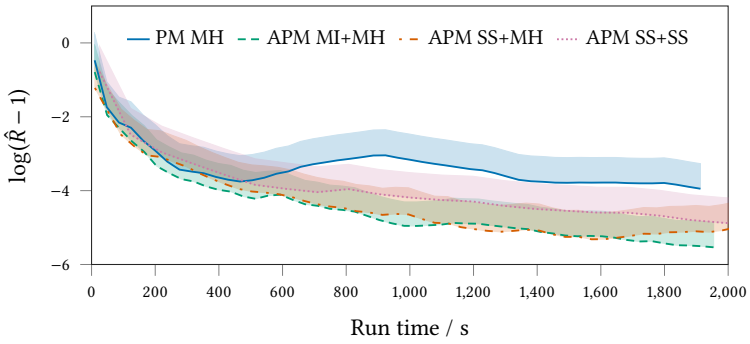
Results of the experiments are summarised in Figure 3.11. As a first measure of performance we consider the relative estimated sampling efficiency of the different methods. For each of the 10 target variables we estimated the ESS for the estimated mean of the variable using `R CODA` [35] and normalised these values by the total number of cubic operations performed in each chain[5]. The means of these values across the 10 chains per method (and standard errors) are shown for each of the target variables in the bar plot in Figure 3.11a.

By this ESS measure of efficiency, the APM MI+MH and APM SS+MH chains both consistently perform better than the PM MH chains, with they performing very similarly to each other, and the APM SS+SS chains perform worse than all other methods. Note that as the updates to the auxiliary variables to do not require any cubic operations (providing the Cholesky factorisations of the Gaussian process prior covariance

5 The mean chain run time per cubic operation performed was 0.0184 ± 0.00007 s for PM MH, 0.0187 ± 0.00014 s for APM MI+MH, 0.0196 ± 0.00012 s for APM SS+MH and 0.0184 ± 0.00013 s for APM SS+SS so using the cubic operation count as a proxy for overall computational cost seems reasonable here and removes the effect of any variable background system processes on the wall-clock run times.



(a) Sampling efficiency. ESS estimates for each of 10 target variables normalised by the number of cubic cost operations performed per chain. The bars show the means values across 10 independent chains with markers for ±1 standard error of mean.



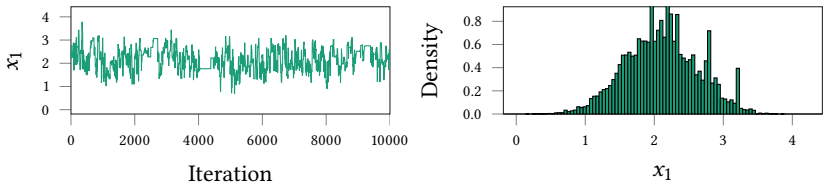
(b) Chain convergence. Plots of PSRF $\hat{R}$ statistic on a log scale computed across 10 independent chains initialised from the prior for increasing number of chain iterations (normalised by mean total run time for each method to adjust for different per iteration run times) for each of four transition operators tested. Curves show median value and filled regions indicate confidence interval to upper 95th percentile of computed estimate. A $\hat{R}$ value of unity is indicative of chains having converged to stationarity, so for the plotted $\log(\hat{R}-1)$ values, more negative values indicate approaching convergence.

Figure 3.11: Gaussian process probit regression results.

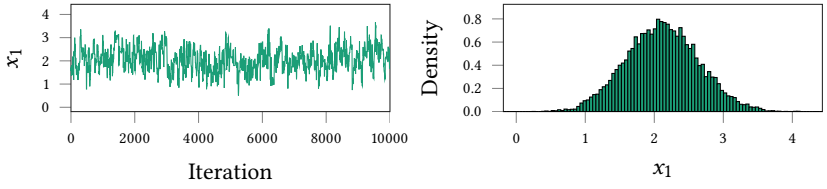
and importance distribution covariance at the current target variable values are cached from the target variable update), there is little effect on the overall run time from using elliptical SS updates to the auxiliary variables as opposed to MI updates, hence the much closer performance here of APM MI+MH and APM SS+MH compared to the previous Gaussian latent variable experiments. The average accept rate of the MI updates to the auxiliary variables in the APM MI+MH chains was 0.24 here suggesting there is probably a limited gain from using elliptical SS updates to the auxiliary variables in this case as the MI updates are likely to be mixing the auxiliary variables sufficiently well.

Although PM MH seems to outperform the APM SS+SS method here, other results suggest the estimated ESS measures of performance should be treated with some caution, with in general estimated ESSs being susceptible to giving misleading results when chains have poorly converged. Figure 3.11b shows plots of the *potential scale reduction factor* (PSRF) convergence diagnostic proposed by Gelman and Rubin in [16], also often termed the $\hat{R}$ statistic. This is a heuristic measure of Markov chain convergence computed from multiple independent chains initialised from a distribution which should be over-dispersed compared to the (common) target distribution (we use the prior here). The diagnostic compares the between-chain and within-chain variance of each variable in the chain state, with a necessary but not sufficient condition for convergence being that these converge to being equal, corresponding to a $\hat{R}$ value of one. We used `CODA` to estimate the $\hat{R}$ values from the 10 independent chains run for each method as a function of an increasing number of iterations in the chain sequences used to compute the $\hat{R}$ estimates. We then accounted for the different per iteration run time of the different methods (in particular the APM SS+SS chains took on average $\sim 2.5\times$ longer per iteration than the other methods) by plotting these $\hat{R}$ values for increasing chain iterations against the estimated run time to complete that number of iterations, the resulting curves shown in Figure 3.11b. The darker coloured curves show the median of the estimated $\hat{R}$ interval and the lighter filled regions of the same colour show the 50th–95th percentile range of the estimate. To allow the curves to be more clearly distinguished, the $\hat{R}$ values are plotted on a shifted log scale i.e. $\log(\hat{R}-1)$, with more negative values therefore corresponding to $\hat{R}$ values closer to one and so indicative of the chains being closer to convergence.

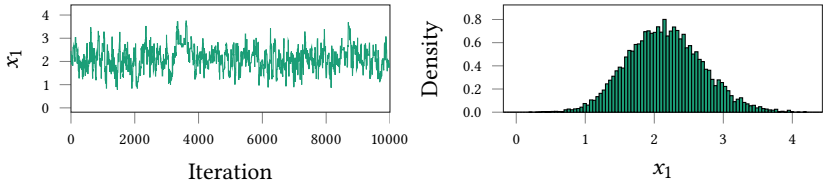
On this measure of performance the PM MH chains seem to perform more poorly, showing a slower convergence rate than the other methods, inclduing the APM SS+SS chains. The non-montonically decreasing behaviour seen in the $\hat{R}$ curve for the PM MH chains seems to be the result of the chains suffering the earlier discussed sticking behaviour, with one of the 10 chains found to have stuck for a run of over 2000 iterations and multiple incidents of sticking periods of hundreds of iterations in all of the chains. An example trace of one of the chains for the x_1 target variable is shown in Figure 3.12a where these sticking periods are clearly visible. The chains run using $N = 500$ importance samples (traces not shown here) also showed sticking behaviour



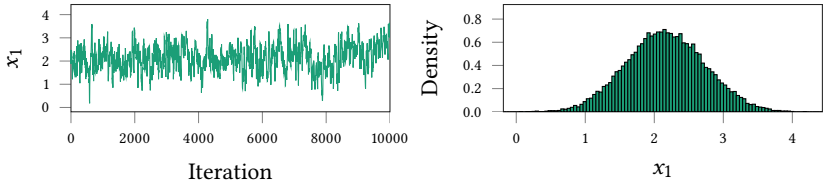
(a) PM MH



(b) APM MI+MH



(c) APM SS+MH



(d) APM SS+SS

Figure 3.12: Example traces and histograms of target variable $x_1 = \log s$ from chains sampled using pseudo-marginal and auxiliary pseudo-marginal approaches in Gaussian process probit regression model inference task. In each row a trace of the sampled values for the x_1 variable for a single 10000 iteration Markov chain is shown in the left plot, while the right plot shows a histogram of the sampled values from all 10 chains. In the histogram plots the number of samples in the chain used to produce the plot have been adjusted to account for the roughly 2.5 times increase in run time for the APM SS+SS chains compared to the other methods.

though somewhat less frequently, suggesting that while increasing the number of importance samples can lessen the impact of these events, it does not seem to necessarily eliminate them. Figure 3.12 also shows example chain traces for the x_1 variable for each of the three other APM methods; in all cases here there are no visible long sticking periods and this was also reflected across the other chains not shown.

The right column of Figure 3.12 shows histograms for the x_1 target variable computed from the samples from all 10 chains for each method (in the case of the APM SS+SS chains only the first 4000 iterations from each chain were included to account for the roughly 2.5 times slower run time per chain in this case). Although we do not have a ground truth for the marginal posterior density here to compare against, it seems reasonable to assume that the spurious peaks in the histogram for the PM MH chains are not a reflection of the true marginal density but instead a result of the long sticking artifacts in the chains causing the states that the chain remains stuck at to be overly represented in the histograms. The APM methods produced much smoother marginal density estimates, with the APM SS+SS chains seeming to give a particularly smooth result here even with the run time adjustment meaning this histogram is computed from less than half the number of samples as used in the other methods. Although by no means conclusive, this provides a further suggestion that the relatively poor standing of the APM SS+SS chains on the ESS measure of performance is not an entirely accurate potrayal of the overall performance of the method.

3.7 DISCUSSION

The auxiliary pseudo-marginal methods discussed in this Chapter are a relatively simple extension to the existing pseudo-marginal MCMC framework which nonetheless offer some important benefits.

The simplest proposed approach of splitting the combined proposed update to both auxiliary and target variables in the standard PM MH algorithm into a separate Metropolis independence updates to the auxiliary variables and Metropolis–Hastings update to the target variables (APM MI+MH) involves changing only a few lines of code in most implementations and adds no further free parameters to tune. Despite involving only a minor change, in the empirical studies performed this adjusted update was found to give significantly better computational

cost normalised sampling efficiency over the standard pseudo-marginal Metropolis–Hastings update, despite in some cases doubling the computational effort per overall chain update. A simple intuition for understanding this improved performance is that for a fixed proposal distribution for the target variables, the accept rate of the MH update to the target variables in the APM MI+MH chains was typically more than double the corresponding accept rate for the overall PM MH update. Therefore the doubling of the number of density estimates needed per iteration was more than outweighed by more than double the number of proposed target variable updates *from the same proposal distribution* (e.g. same Gaussian random-walk step-size) being accepted

The size of the increase in the accept rates for a fixed proposal distribution is dependent on how high the variance of the density estimator is or equivalently how dependent the target and auxiliary variables are under the auxiliary joint target. For high variances cases e.g. when using $N = 1$ importance sample, the increase in accept rates for the target variable updates in APM MI+MH chains over the accept rate of the PM MH updates is higher due to poor performance of making independent proposed updates to the auxiliary variables in the PM MH having a strong deleterious effect on the PM MH accept rate. For example in the Gaussian latent variable model experiments when using $N = 1$ importance sample the accept rate of the MH updates in the APM MI+MH chains was typically around a factor of 20 higher than the accept rate for the corresponding PM MH chains using the same proposal step size. As the variance of the density estimator is decreased by increasing N, the difference in the accept rates for a fixed proposal step size becomes less marked with around a factor five difference for $N = 8$ and around a factor two difference for $N = 2$ between APM MI+MH and PM MH. So with a lower variance estimator the difference in performance between APM MI+MH and PM MH becomes less marked.

However the recommendation of [39] suggests that when the computational cost of each PM MH update scales linearly with N (and when using a density estimator formed as an average of unbiased Monte Carlo estimates) that using $N = 1$ is close to optimal for PM MH despite the higher estimator variance. As the low N, high density estimator variance cases are precisely when we expect to see the largest potential gains from using APM MI+MH over PM MH this suggests when this linear cost scaling argument is valid there will often be a computational

gain from using APM MI+MH. In some cases as we saw in the Gaussian process experiments we can form a much lower variance density estimate by expending some computational effort to fit a good importance distribution. In these cases due to the additional overhead of the fitting procedure the linear cost scaling argument no longer applies. Further the density estimates in this case may be sufficiently low variance for there to be little improvement in accept rates of updates to the target variables by splitting the PM MH in to separate MI and MH updates. However typically in these cases the overhead introduced by separately updating the auxiliary variables in an MI step will also be much less than the cost of the original PM MH update, as for fixed values of the target variables the importance distribution does not need to be refitted and any target variable dependent computations such as Cholesky factorisations of covariance matrices can be cached and reused. Therefore the overall cost per APM MI+MH update will be very close to that of each PM MH update and so even a small improvement in accept rate of the target variable updates can make it worthwhile to split the update.

Perhaps more important than the sampling efficiency gains seen from using APM MI+MH over PM MH in the experiments here was the significantly improved ability to tune the MH updates in the algorithm even when using a high-variance density estimator. By decoupling the dependency of the MH accept rate from the density estimator variance, theoretical guidelines for choosing a proposal step-size based on the average accept rate can be straightforwardly applied to tune APM MI+MH updates. The resulting increased ease of use of the algorithm and decreased requirement for user intervention to get good performance might often make APM MI+MH an attractive choice even when the extra runtime overhead per update negates any sampling efficiency gains. Further the separate MI step accept rate of the APM MI+MH update provide a diagnostic already computed as part of the chain updates which can alert users to issues with poor mixing of the auxiliary variables due to low accept rates of the auxiliary updates. In contrast it will not always be clear if a poor accept rate of a PM MH chain is due to poor choice of the target variables proposal distribution or due to a high density estimator variance, and separately monitoring the density estimator variance as part of the update adds overhead while not being as directly interpretable as the MI step accept rate.

If initial runs using an APM MI+MH method do show a very low accept rate for the updates to the auxiliary variables which might lead to convergence issues, the proposed APM SS+MH approaches offer a simple 'plug-in' solution to improve mixing of the auxiliary variables without having to tune a separate proposal distribution for a MH update to the auxiliary variables. If the auxiliary variables can be naturally represented as being marginally distributed according to the standard normal distribution, then elliptical slice sampling is a straightforward choice, having no free parameters to tune and still initially proposing bold moves to near independent points in the auxiliary space while able to back-off to more conservative updates to ensure a non-zero move to the auxiliary variables under weak smoothness conditions.

Another common case is auxiliary variables which are naturally parameterised as a vector of standard uniform draws, in which case reflective linear slice sampling offers analagous benefits. Although the linear slice sampling algorithm does have a free initial bracket width parameter to be chosen, in general (as seen in the experiments using this algorithm for updates to the target variables in the Gaussian latent variable model experiments) the efficiency of the algorithm is not strongly dependent on the choice of this parameter providing it is set large enough to cover most of the intersection of the slice with the sampled line as the exponential shrinking of the bracket on proposing an off slice point will quickly reduce the bracket to a more appropriate size if set initially too large. For reflective slice sampling in the unit hypercube a fixed initial bracket width of one and a direction vector $\boldsymbol{v}$ sampled from $\mathcal{N}(\mathbf{0}, \mathbf{I})$ was found to work well in experiments applying APM SS+MH methods to inference in a doubly-intractable Ising model problem in [30].

Independently of and concurrently with the original conference publication [30] related to this work, both Dahlin et al. [8] and Deligiannidis et al. [9] considered related frameworks in which the auxiliary random variables of a pseudo-marginal density estimator are updated using a Metropolis–Hastings update leaving the distribution defined by the density (3.8) on the joint auxiliary–target variable space invariant. Both assume a parameterisation in which the auxiliary variables have an isotropic standard normal marginal distribution $\rho(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{u} \mid \mathbf{0}, \mathbf{I})$,

and consider a Metropolis–Hastings update to the auxiliary variables with proposal density

$$r^*(\boldsymbol{u}' \mid \boldsymbol{u}) = \mathcal{N}\left(\boldsymbol{u}' \mid \sqrt{1-\lambda^2}\boldsymbol{u}, \lambda^2 \mathbf{I}\right) \tag{3.20}$$

which can be variously considered as a discretisation of a Ornstein-Uhlenbeck diffusion process, exact update of an AR(1) model or as an fixed step size update on an elliptical path that the elliptical slice sampling algorithm 5 generalises by adaptively setting the step size λ. This fixed step-size Metropolis–Hastings update is more amenable to analysis, with both [8] and [9] giving much more extensive theoretical justifications for using perturbative updates to the auxiliary variables (or equivalents introducing correlations in between the auxiliary variable samples) than the mainly intuition based and empirical arguments made here. These theoretical insights are important for informing future development of these ideas. In practical settings however, though the above MH update with an optimally tuned choice of λ may give better sampling efficiency performance compared to the elliptical slice sampling updates proposed here, we would suggest that the additional tuning burden placed on the user and loss of robustness in cases where the appropriate step size varies across the state space, would suggest that elliptical slice sampling updates to the auxiliary variables are still often a good default choice.

The empirical evidence for using slice sampling updates to the target variables as in the proposed APM MI+SS and APM SS+SS methods is less strong, with in both of the models considered in the experiments here these methods having poorer run-time adjusted efficiency than the well tuned APM MI+MH and APM SS+MH methods respectively. If adapting an existing PM MH algorithm where some effort has already been extended to identify an appropriate proposal distribution for updates to the target variables or other information is available to inform this choice, the additional overhead of the slice sampling updates might not be worthwhile. In cases however where we have less prior knowledge about appropriate scales for updates to the target variables or are more concerned with overall robustness and ease of use, slice sampling updates to the target variables are likely to be more attractive however.

Subsequent to the publication of the conference publication related to this work, Lindsten and Doucet proposed the use of *Hamiltonian*

Monte Carlo (HMC) within an (auxiliary) pseudo-marginal framework [22]. Under the assumption that the joint auxiliary target density (3.8) is defined with respect to the Lebesgue measure, their *pseudo-marginal Hamiltonian Monte Carlo* algorithm proposes jointly updating the auxiliary and target variables using a HMC transition operator. In particular they assume the marginal distribution on the auxiliary variables R is standard norma $\mathcal{N}(\mathbf{0}, \mathbf{I})$ and leverage this to propose an alternative symplectic integrator to the typical leapfrog scheme which reduces the error when multiple auxiliary variable samples are used in the density estimator.

In numerical experiments with a hierarchical model of a diffraction process with a three-dimensional target space, it was found that the proposed pseudo-marginal HMC algorithm gave similar performancy to using a APM SS+MH update when normalised by the computational cost per update. In a second experiment with a generalised linear mixed model with a 13 dimensional target space, the proposed pseudo-marginal HMC algorithm was compared to a APM SS+MH update in which the update to the target variables is formed of a sequential scan of per-dimension random-walk Metropolis updates to each individual target variable. It is reported that attempts to jointly update all target variables in the MH step led to very poor acceptance rates. The traces for the pseudo-marginal HMC chain (Figure 4 in [22]) in this case indicate improved mixing compared to the APM SS+MH update, though as the run-time per sample of the pseudo-marginal HMC method is reported to be approximately 3.5 times higher in the implementation used and the traces do not appear to be run-time adjusted it is not clear what a cost normalised comparison would show. Autocorrelation plots for chains from the two approaches are also shown (Figure 13 in [22]), with the pseudo-marginal HMC method showing quicker decay of the autocorrelations per sample lag compared to APM SS+MH though again it is not clear if the autocorrelation plots are run-time adjusted. Both the pseudo-marginal HMC and APM SS+MH chains appear to mix significantly better than *Particle Gibbs* [2], an auxiliary variable approach based on a particle filter density estimator.

The use of HMC updates with an auxiliary pseudo-marginal framework seems an appealing idea when the required gradients are available due to the improved performance in complex high-dimensional target distributions often offered by HMC methods, and the integrator proposed

by [22] is an elegant approach for exploiting structure in the auxiliary target distribution to give improved performance when the number of auxiliary variable dimensions is large. Though in the experiments in [22] it is not clear how significant the gain in performance is over using random-walk Metropolis updates to the target variables in a APM SS+MH method, it seems plausible that in models with higher-dimensional target space that HMC updates would start to increasingly outperform random-walk Metropolis updates to the target variables. In the next chapter we will discuss related methods which apply HMC updates to perform inference in simulator models; this work was performed concurrently and independently to [22].

BIBLIOGRAPHY

[1] Christophe Andrieu, Arnaud Doucet and Roman Holenstein. 'Particle Markov chain Monte Carlo methods'. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 72.3 (2010), pp. 269–342.

[2] Christophe Andrieu and Gareth O Roberts. 'The pseudo-marginal approach for efficient Monte Carlo computations'. In: *The Annals of Statistics* (2009).

[3] Christophe Andrieu and Johannes Thoms. 'A tutorial on adaptive MCMC'. In: *Statistics and computing* 18.4 (2008), pp. 343–373.

[4] Mark A Beaumont. 'Estimation of population growth or decline in genetically monitored populations'. In: *Genetics* 164.3 (2003), pp. 1139–1160.

[5] Luke Bornn, Natesh S Pillai, Aaron Smith and Dawn Woodard. 'The use of a single pseudo-sample in approximate Bayesian computation'. In: *Statistics and Computing* 27.3 (2017), pp. 583–590.

[6] Nicolas Chopin and Sumeetpal S Singh. 'On particle Gibbs sampling'. In: *Bernoulli* 21.3 (2015), pp. 1855–1883.

[7] H. Cramér. *Mathematical Methods of Statistics*. Princeton University Press, 1946.

[8] Johan Dahlin, Fredrik Lindsten, Joel Kronander and Thomas B Schön. 'Accelerating pseudo-marginal Metropolis-Hastings by correlating auxiliary variables'. In: *arXiv preprint arXiv:1511.05483* (2015).

[9] George Deligiannidis, Arnaud Doucet, Michael K Pitt and Robert Kohn. 'The Correlated Pseudo-Marginal Method'. In: *arXiv preprint arXiv:1511.04992* (2015).

[10] A. Doucet, A. Smith, N. de Freitas and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Information Science and Statistics. Springer New York, 2001. ISBN: 9780387951461. URL: https://books.google.co.uk/books?id=uxX-koqKtMMC.

[11] Arnaud Doucet, MK Pitt, George Deligiannidis and Robert Kohn. 'Efficient implementation of Markov chain Monte Carlo when using an unbiased likelihood estimator'. In: *Biometrika* 102.2 (2015), pp. 295–313.

[12] Oliver B Downs, David JC MacKay and Daniel D Lee. 'The nonnegative Boltzmann machine'. In: *Advances in Neural Information Processing Systems*. 2000, pp. 428–434.

[13] Robert G Edwards and Alan D Sokal. 'Generalization of the Fortuin–Kasteleyn–Swendsen–Wang representation and Monte Carlo algorithm'. In: *Physical review D* 38.6 (1988), p. 2009.

[14] Maurizio Filippone and Mark Girolami. 'Pseudo-marginal Bayesian inference for Gaussian processes'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.11 (2014), pp. 2214–2226.

[15] Andrew Gelman, Walter R Gilks and Gareth O Roberts. 'Weak convergence and optimal scaling of random walk Metropolis algorithms'. In: *The annals of applied probability* 7.1 (1997), pp. 110–120.

[16] Andrew Gelman and Donald B Rubin. 'Inference from iterative simulation using multiple sequences'. In: *Statistical science* (1992), pp. 457–472.

[17] Neil J Gordon, David J Salmond and Adrian FM Smith. 'Novel approach to nonlinear/non-Gaussian Bayesian state estimation'. In: *IEE Proceedings F (Radar and Signal Processing)*. Vol. 140. 2. IET. 1993, pp. 107–113.

[18] Todd L Graves. 'Automatic step size selection in random walk Metropolis algorithms'. In: *arXiv preprint arXiv:1103.5986* (2011).

[19] David M Higdon. 'Auxiliary variable methods for Markov chain Monte Carlo with applications'. In: *Journal of the American Statistical Association* 93.442 (1998), pp. 585–595.

[20] AD Kennedy and Julius Kuti. 'Noise without noise: a new Monte Carlo method'. In: *Physical review letters* 54.23 (1985), p. 2473.

[21] M. Lichman. *UCI Machine Learning Repository*. 2013. URL: http://archive.ics.uci.edu/ml.

[22] Fredrik Lindsten and Arnaud Doucet. 'Pseudo-Marginal Hamiltonian Monte Carlo'. In: *arXiv preprint arXiv:1607.02516* (2016).

[23] Anne-Marie Lyne, Mark Girolami, Yves Atchadé, Heiko Strathmann and Daniel Simpson. 'On Russian roulette estimates for Bayesian inference with doubly-intractable likelihoods'. In: *Statistical science* 30.4 (2015), pp. 443–467.

[24] Olvi L Mangasarian, W Nick Street and William H Wolberg. 'Breast cancer diagnosis and prognosis via linear programming'. In: *Operations Research* 43.4 (1995), pp. 570–577.

[25] Paul Marjoram, John Molitor, Vincent Plagnol and Simon Tavaré. 'Markov chain Monte Carlo without likelihoods'. In: *Proceedings of the National Academy of Sciences* (2003).

[26] Jesper Møller, Anthony N Pettitt, R Reeves and Kasper K Berthelsen. 'An efficient Markov chain Monte Carlo method for distributions with intractable normalising constants'. In: *Biometrika* 93.2 (2006), pp. 451–458.

[27] Iain Murray. 'Advances in Markov chain Monte Carlo methods'. PhD thesis. University College London, University of London, 2007.

[28] Iain Murray, Ryan Prescott Adams and David J.C. MacKay. 'Elliptical slice sampling'. In: JMLR: W&CP 9 (2010), pp. 541–548.

[29] Iain Murray, Zoubin Ghahramani and David J. C. MacKay. 'MCMC for doubly-intractable distributions'. In: *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI-06)*. AUAI Press, 2006, pp. 359–366.

[30] Iain Murray and Matthew Graham. 'Pseudo-marginal slice sampling'. In: *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*. 2016, pp. 911–919.

[31] Peter Neal and Clement Lee. 'Optimal scaling of the independence sampler: Theory and Practice'. In: *arXiv preprint arXiv:1511.04334* (2015).

[32] Radford M Neal. 'Slice sampling'. In: *Annals of statistics* (2003).

[33] Omiros Papaspiliopoulos, Gareth O Roberts and Martin Sköld. 'Non-centered parameterisations for hierarchical models and data augmentation'. In: *Bayesian Statistics 7: Proceedings of the Seventh Valencia International Meeting*. Vol. 307. Oxford University Press, USA. 2003.

[34] Michael Pitt, Ralph Silva, Paolo Giordani and Robert Kohn. 'Auxiliary particle filtering within adaptive Metropolis-Hastings sampling'. In: *arXiv preprint arXiv:1006.1914* (2010).

[35] Martyn Plummer, Nicky Best, Kate Cowles and Karen Vines. 'CODA: Convergence Diagnosis and Output Analysis for MCMC'. In: *R News* 6.1 (2006), pp. 7–11. URL: http://CRAN.R-project.org/doc/Rnews/Rnews_2006-1.pdf.

[36] James Gary Propp and David Bruce Wilson. 'Exact sampling with coupled Markov chains and applications to statistical mechanics'. In: *Random structures and Algorithms* 9.1-2 (1996), pp. 223–252.

[37] C Radhakrishna Rao. 'Information and the accuracy attainable in the estimation of statistical parameters'. In: *Breakthroughs in statistics*. Springer, 1992, pp. 235–247.

[38] C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. Adaptative computation and machine learning series. University Press Group Limited, 2006. ISBN: 9780262182539.

[39] Chris Sherlock, Alexandre Thiery and Anthony Lee. 'Pseudo-marginal Metropolis–Hastings using averages of unbiased estimators'. In: *arXiv preprint arXiv:1610.09788* (2016).

[40] Chris Sherlock, Alexandre H Thiery, Gareth O Roberts and Jeffrey S Rosenthal. 'On the efficiency of pseudo-marginal random walk Metropolis algorithms'. In: *The Annals of Statistics* 43.1 (2015), pp. 238–275.

[41] Martin A Tanner and Wing Hung Wong. 'The calculation of posterior distributions by data augmentation'. In: *Journal of the American statistical Association* 82.398 (1987), pp. 528–540.

[42] David A Van Dyk and Xiao-Li Meng. 'The art of data augmentation'. In: *Journal of Computational and Graphical Statistics* 10.1 (2001), pp. 1–50.