LETTER Communicated by Emery Brown

Markov Chain Monte Carlo Methods for State-Space Models with Point Process Observations

Ke Yuan
ky08r@ecs.soton.ac.uk
School of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, U.K.

Mark Girolami
m.girolami@ucl.ac.uk
Department of Statistical Science, Centre for Computational Statistics and Machine Learning, University College London, London, WC1E 6BT, U.K.

Mahesan Niranjan
mn@ecs.soton.ac.uk
School of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, U.K.

This letter considers how a number of modern Markov chain Monte Carlo (MCMC) methods can be applied for parameter estimation and inference in state-space models with point process observations. We quantified the efficiencies of these MCMC methods on synthetic data, and our results suggest that the Reimannian manifold Hamiltonian Monte Carlo method offers the best performance. We further compared such a method with a previously tested variational Bayes method on two experimental data sets. Results indicate similar performance on the large data sets and superior performance on small ones. The work offers an extensive suite of MCMC algorithms evaluated on an important class of models for physiological signal analysis.

1 Introduction

Latent processes in the brain during the processing of controlled stimuli manifest as multiple neural spike trains that are obtained via extracellular recordings, followed by some preprocessing such as spike sorting. In several applications (e.g., brain-computer interface), it is of interest to infer these latent processes from recordings of spike trains using data-driven methods. Traditional approaches to modeling spike trains involve treating the interspike intervals as continuous signals followed by the application of signal processing techniques (Jolivet et al., 2008; Ivanov et al., 1996). Such

treatment, however, ignores the obvious structure in spike train signals, which are discrete processes in time. Smith and Brown (2003) address this concern, formulating a state-space model with point process observations (SSPP). In this model, an underlying first-order autoregressive process defines an evolving system state that modulates an approximate Bernoulli process using a parameterized intensity function.

In potential applications that motivate such data-driven modeling, the inferred latent space can be viewed as an approximation to the responses in the brain that serve to process any applied stimuli. This could potentially be used as input to some control software in a brain-computer interface setting. Further, parameters estimated by fitting the model to observed data may be used as features in a statistical pattern classification setting to automatically separate between classes of stimuli.

In introducing this model, Smith and Brown (2003) derived an approximate expectation-maximization (EM) algorithm for parameter estimation and state inference. In subsequent work, it was shown that the corresponding expected log complete data likelihood (also known as the Q-function) was unimodal and highly nongaussian (skewed) with respect to its parameters (Yuan & Niranjan, 2010). This nongaussian nature of the likelihood motivates a Bayesian treatment with the objective of avoiding the mismatch between maximum likelihood estimates and posterior means. Starting from this, Zammit Mangion, Yuan, Kadirkamanathan, Niranjan, and Sanguinetti (2011) proposed a variational Bayes (VB) method for an SSPP model that provides a computationally efficient way of approximating the joint posterior based on the mean-field method. A limitation of such an approach is that it builds on an unrealistic assumption of independence between states and parameters. The resulting posteriors, which are obtained by minimizing the Kullback-Leibler (KL) divergence between the true posterior of the unknowns and its gaussian or other approximations within the conjugate exponential family, are not exact solutions to the inference task. Since such solutions are often used to offer important insights into the underlying biology, it is of interest to ask how far they might be from the true posteriors.

In this contribution, we use the more powerful Markov chain Monte Carlo (MCMC) methods (see Neal, 1993), which offer asymptotically exact posteriors, to explore different approximate schemes of inference for SSPP models. For this, we consider a number of variants of MCMC methods suitable for SSPP models, thereby enriching the array of tools for inference and parameter estimation. In particular, we examine two recently advanced MCMC methods—the particle marginal Metropolis-Hastings (PMMH) algorithm (Andrieu, Doucet, & Holenstein, 2010) and the Riemann manifold Hamiltonian Monte Carlo (RMHMC) method (Girolami & Calderhead, 2011)—as well as the traditional Hamiltonian Monte Carlo (HMC) method (Duane, Kennedy, Pendleton, & Roweth, 1987). These methods are

demonstrated on a synthetic data set, showing significant efficiency improvement when compared with a commonly used a single-site update Gibbs sampler. In these simulations, RMHMC outperforms the others with high efficiency scores and comparable computational costs. We also consider two case studies using RMHMC and VB methods, the first being a neural representation of various taste stimuli in rat (Di Lorenzo & Victor, 2003), and second, the response variability in marmoset parvocellular neurons (Victor, Blessing, Forte, Buzás, & Martin, 2007). Our results show that posteriors obtained by RMHMC and VB are in general quite similar; in particular, RMHMC shows an advantage when dealing with data sets that are short time records and sparse in the number of spikes.

2 Model Description

Consider an observation interval $(0, T]$, where C channels of events are recorded. We let $Y^c(t)$ denote the counting function of events in each channel c. A point-process model over those events can be fully characterized using its conditional intensity function (CIF) (Daley & Vere-Jones, 2003), where for each channel c, $\lambda^c(t)$, which is also known as the instantaneous rate function of the events, has the expression

$$\lambda^c(t) = \lim_{\Delta \to 0} \frac{\Pr(Y^c(t+\Delta) - Y^c(t) = 1 | x(t), H(t))}{\Delta},$$

where $x(t)$ denotes an underlying state variable and $H(t)$ represents history information. In order to obtain a discrete time model, we choose a large K to divide $(0, T]$ into K bins with equal widths $\Delta = T/K$. For each channel per time slot $k\Delta$, let y_k^c represent an observed event, such that $y_k^c = 1$ if a spike is present and 0 otherwise. Δ is sufficiently small such that there is only one spike per interval Δ. Following Smith and Brown (2003), we give the discretized CIF function a parametric form, defined as

$$\lambda_k^c = \exp(\mu + \beta_c x_k), \tag{2.1}$$

where μ is a background firing rate, assumed to be the same for all channels. The states modulate firing via the multiplicative terms β_c. Note that for different applications, the CIF can take various functional forms (Ergün, Barbieri, Eden, Wilson, & Brown, 2007; Wang, Paiva, Príncipe, & Sanchez, 2009). The probability of an event in $k\Delta$ in the cth channel given the hidden system states x_k and parameters is defined as an approximated Bernoulli probability mass function (see the detailed the derivation in Brown, Barbieri,

Eden, & Frank, 2002; Smith & Brown, 2003):

$$p\left(y_k^c|x_k, \mu, \beta_c\right) = \left(\lambda_k^c\Delta\right)^{y_k^c} \exp\left(-\lambda_k^c\Delta\right). \tag{2.2}$$

The discretized latent state variable x_k follows an AR(1) transition model, for $k = 1, \ldots, K$,

$$x_k = \rho x_{k-1} + \alpha I_k + \varepsilon_k, \tag{2.3}$$

where ε_k are gaussian noise from $\mathcal{N}(0, \sigma_\varepsilon^2)$. I_k is 1 if there is an external stimulus at $k\Delta$ and 0 otherwise. We assume an initial state $x_0 \sim \mathcal{N}(0, \sigma_\varepsilon^2/(1-\rho^2))$. Equations 2.1 to 2.3 define a SSPP model of interest in this letter.

Further, let $x_{0:K} = \{x_k\}_{k=0}^K$, $y_k = \{y_k^c\}_{c=1}^C$ and $y_{0:K} = \{y_k\}_{k=1}^K$, and a parameter ensemble $\boldsymbol{\theta} = \{\rho, \alpha, \mu, \beta_{1:C}\}$. With these, the joint likelihood of states and observations can be written as

$$p(y_{1:K}, x_{0:K}|\boldsymbol{\theta}) = \prod_{k=1}^K \prod_{c=1}^C p\left(y_k^c|x_k, \mu, \beta_c\right) p(x_0) \prod_{k=1}^K p\left(x_k|x_{k-1}, \rho, \alpha, \sigma_\varepsilon^2\right).$$

The log-joint likelihood is

$$\begin{aligned}
\mathcal{L}(y_{1:K}, x_{0:K}|\boldsymbol{\theta}) = &-\frac{K+1}{2}\log 2\pi - (K+1)\log\sigma_\varepsilon^2 \\
&- \sum_{k=1}^K \frac{(x_k - \rho x_{k-1} - \alpha I_k)^2}{2\sigma_\varepsilon^2} \\
&+ \frac{1}{2}\log(1-\rho^2) - \frac{x_0^2(1-\rho^2)}{2\sigma_\varepsilon^2} \\
&+ \sum_{k=1}^K \sum_{c=1}^C \left[y_k^c(\mu + \beta_c x_k + \log\Delta) - \exp(\mu + \beta_c x_k)\Delta\right].
\end{aligned}$$

An issue of identifiability relating to this model exists. This arises from the fact that parameter β appears in the likelihood only via the product $\beta_c x_k$, and the term α multiplies a binary stimulus that is nonzero only at sparse points in time. This makes α and β difficult to estimate, as Smith and Brown (2003) and Zammit Mangion et al. (2011) noted. In practice, we fix β_c and σ_ε^2 to ensure a strong, identifiable model, as with previous work.

3 Markov Chain Monte Carlo for State-Space Models

We start with a brief presentation of MCMC in the context of general state-space models before delving into variants we introduce for the SSPP model. A detailed review on this subject can be found in Fearnhead (2010). From the Bayesian perspective, inference in a general state-space model targets the joint posterior distribution of parameters and hidden states, denoted as $p(\boldsymbol{\theta}, x_{0:K}|y_{1:K})$. A Gibbs sampler, iteratively drawing samples from $p(x_{0:K}|y_{1:K}, \boldsymbol{\theta})$ and $p(\boldsymbol{\theta}|x_{0:K}, y_{1:K})$, is the most popular method to sample from such a posterior distribution. In practice, sampling from $p(\boldsymbol{\theta}|x_{0:K}, y_{1:K})$ is often easy, whereas designing a sampler for $p(x_{0:K}|y_{1:K}, \boldsymbol{\theta})$ is trickier due to the fact that the states are highly correlated and can have a large variation in scale.

The simplest implementation of such a sampling approach is a single-site update Gibbs sampler for both hidden states and parameters, where the components of $x_{0:K}$ and $\boldsymbol{\theta}$ are updated one at a time (see Geweke & Tanizaki, 2001, for details). For sampling states, a sequential sampler that updates each state conditioning on all the rest of the states is used. Such an approach is easy to implement, since the conditional distribution of each state given all the others reduces to one conditioning only on its two adjacent states: $p(x_k|y_{1:K}, x_{k-1}, x_{k+1}, \boldsymbol{\theta})$. However, due to the severe correlation between states, such a sampler may lead to slow mixing (such slow mixing is evident in the SSPP; empirical results are shown in section 7).

To overcome this, Shephard and Pitt (1997) propose a block Gibbs sampler in which instead of single-site updating, the states are grouped into many blocks and updated simultaneously. In this case, the conditionals on states change to the density of each block of states given the two neighboring states of the block: $p(x_{k:s}|y_{1:K}, x_{k-1}, x_{s+1})$, where $k < s < K$. Ideally, one needs the block to be as large as possible; however, when the block size is too large, it is hard to sample from the conditional in most general state-space models. If the block is not large enough, the sampler still suffers from state dependency issues. A balance between the extremes is often difficult to strike.

In the case of block size set equal to the total time points in the model, the state sequences are updated simultaneously from $p(x_{0:K}|y_{1:K}, \boldsymbol{\theta})$. Such updates can be performed exactly only in the linear gaussian models using the Kalman filter (Carter & Kohn, 1994) and discrete hidden Markov model using the forward-backward method (Scott, 2002). However, recent developments in MCMC provide flexible means for updating the whole state sequence for more general state-space models. In the following sections, we introduce several such efficient sampling schemes that can be applied to the SSPP model.

4 Particle Marginal Metropolis-Hastings Algorithm

Andrieu et al. (2010) propose a particle marginal Metropolis-Hastings (PMMH) algorithm that not only jointly samples states but also updates parameters simultaneously with the states. We first review this method.

One may use a proposal mechanism joint in states and parameters as below:

$$q(\{\boldsymbol{\theta}^*, x_{0:K}{}^*\}|\{\boldsymbol{\theta}, x_{0:K}\}) = q(\boldsymbol{\theta}^*|\boldsymbol{\theta})p(x_{0:K}{}^*|y_{1:K}, \boldsymbol{\theta}^*),$$

where the superscript $*$ denotes for proposed variables. Such a proposal mechanism requires an efficient sampling approach for the states, so that the proposed $x^*_{0:K}$ is linked to the proposed $\boldsymbol{\theta}^*$ in a "deterministic" fashion. The only remaining degree of freedom is in the parameter proposal process. Thus, the MH acceptance ratio reduces to

$$\frac{p(x^*_{0:K}, \boldsymbol{\theta}^*|y_{1:K})q(\{\boldsymbol{\theta}, x_{0:K}\}|\{\boldsymbol{\theta}^*, x^*_{0:K}\})}{p(x_{0:K}, \boldsymbol{\theta}|y_{1:K})q(\{\boldsymbol{\theta}^*, x^*_{0:K}\}|\{\boldsymbol{\theta}, x_{0:K}\})} = \frac{p(y_{1:K}|\boldsymbol{\theta}^*)p(\boldsymbol{\theta}^*)q(\boldsymbol{\theta}|\boldsymbol{\theta}^*)}{p(y_{1:K}|\boldsymbol{\theta})p(\boldsymbol{\theta})q(\boldsymbol{\theta}^*|\boldsymbol{\theta})}. \tag{4.1}$$

There are two key issues with this algorithm: how to directly draw samples from the smoothing distribution $p(x_{0:K}|y_{1:K}, \boldsymbol{\theta})$ and how to evaluate the marginal likelihood $p(y_{1:K}|\boldsymbol{\theta})$. For SSPP and many general state-space models, exact computation of the marginal likelihood is not possible, and one needs to perform approximations.

The PMMH algorithm, by employing the sequential Monte Carlo (SMC) approach (see Doucet, de Freitas, & Gordon, 2001), provides an integrated solution to both of the above problems. It is straightforward to use SMC for sampling hidden states of general state-space models. Moreover, SMC also estimates the marginal likelihood by importance sampling.

The marginal likelihood $p(y_{1:K}|\boldsymbol{\theta})$ can be decomposed as

$$p(y_{1:K}|\boldsymbol{\theta}) = p(y_1|\boldsymbol{\theta}) \prod_{k=2}^{K} p(y_k|y_{1:k-1}, \boldsymbol{\theta}), \tag{4.2}$$

where each component takes the form

$$p(y_k|y_{1:k-1}, \boldsymbol{\theta}) = \int p(y_k|x_k, \boldsymbol{\theta})p(x_k|y_{1:k-1}, \boldsymbol{\theta})dx_k. \tag{4.3}$$

With the SMC algorithm, one can simply add up the unnormalized weights of each particle for time k to obtain an estimate of $p(y_k|y_{1:K}, \boldsymbol{\theta})$. Further, multiplying all components yields an estimate of $p(y_{1:K}|\boldsymbol{\theta})$.

The PMMH algorithm can be described in pseudocode as follows:

Algorithm 1: The PMMH algorithm (Andrieu et al., 2010).

Input: Initial $\boldsymbol{\theta}^{(0)}$, ${x_{0:K}}^{(0)}$ and $\hat{p}(y_{1:K}|\boldsymbol{\theta}^{(0)})$;

for $i \leftarrow 1$ **to** M **do**

  draw $\boldsymbol{\theta}^*$ from $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(i-1)})$;

  use SMC to sample ${x_{0:K}}^* \sim p(x_{0:K}|y_{1:K}, \boldsymbol{\theta}^*)$ and compute $\hat{p}(y_{1:K}|\boldsymbol{\theta}^*)$;

  compute acceptance ratio

$$a = \frac{\hat{p}(y_{1:K}|\boldsymbol{\theta}^*)p(\boldsymbol{\theta}^*)q(\boldsymbol{\theta}^{(i-1)}|\boldsymbol{\theta}^*)}{\hat{p}(y_{1:K}|\boldsymbol{\theta}^{(i-1)})p(\boldsymbol{\theta}^{(i-1)})q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(i-1)})};$$

  draw $u \sim \text{Uniform}[0, 1]$;

  if $u < a$ **then**

    set $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^*$, ${x_{0:K}}^{(i)} = {x_{0:K}}^*$ and $\hat{p}(y_{1:K}|\boldsymbol{\theta}^{(i)}) = \hat{p}(y_{1:K}|\boldsymbol{\theta}^*)$;

  else

    set $\boldsymbol{\theta}^{(i)} = \boldsymbol{\theta}^{(i-1)}$, ${x_{0:K}}^{(i)} = {x_{0:K}}^{(i-1)}$ and $\hat{p}(y_{1:K}|\boldsymbol{\theta}^{(i)}) = \hat{p}(y_{1:K}|\boldsymbol{\theta}^{(i-1)})$;

5 Riemann Manifold Hamiltonian Monte Carlo

The PMMH provides a mathematically rigorous sampling approach. Its computational scaling is $\mathcal{O}(NTM)$, where N is the number of the particles used in SMC and T and M are the total numbers of time points and MCMC iterations, respectively. For neural spike train modeling with SSPP models, the length of time series is often long. Moreover, in order to achieve acceptable performance of SMC, thousands of particles are needed. As a result, computational considerations may be high for the PMMH algorithm.

An alternative class of efficient MCMC methods consists of gradient-based methods, in which the gradient of the underlying distribution is used to assist large moves. A representative of this class is the Hamiltonian Monte Carlo (HMC) method (Duane et al., 1987). HMC employs a Hamiltonian dynamical system as a proposal mechanism, with the proposed variables adjusted by a Metropolis step (see a recent review in Neal, 2010). However, the effective use of HMC requires a high level of tuning, which is not feasible with high-dimensional problems. Girolami and Calderhead (2011), by considering the manifold structure of the distribution of interest, propose a novel algorithm, the Riemann manifold Hamiltonian Monte Carlo (RMHMC) method, to automatically tune HMC. We first introduce RMHMC on a general problem setting.

Assume we are interested in sampling from a probability density function $p(\mathbf{x})$, where $\mathbf{x} \in \mathbb{R}^D$, $\mathcal{L}(\mathbf{x})$ denotes the logarithm of $p(\mathbf{x})$. By introducing an auxiliary variable $\mathbf{p} \in \mathbb{R}^D$ with density $p(\mathbf{p}) = \mathcal{N}(\mathbf{0}, \mathbf{G}(\mathbf{x}))$, we can write

the negative joint log density of $p(\mathbf{x}, \mathbf{p})$ as

$$H(\mathbf{x}, \mathbf{p}) = -\mathcal{L}(\mathbf{x}) + \frac{1}{2}\log\left((2\pi)^D|\mathbf{G}(\mathbf{x})|\right) + \frac{1}{2}\mathbf{p}^T\mathbf{G}(\mathbf{x})^{-1}\mathbf{p}. \tag{5.1}$$

Following Duane et al. (1987), $H(\mathbf{x}, \mathbf{p})$ can be interpreted as a Hamiltonian in physics, which consists of the sum of a potential energy function $-\mathcal{L}(\mathbf{x})$ at position $\mathbf{x}$ and a kinetic energy function $\frac{1}{2}\mathbf{p}^T\mathbf{G}(\mathbf{x})^{-1}\mathbf{p}$ with momentum variable $\mathbf{p}$ and a mass matrix $\mathbf{G}(\mathbf{x})$. In the traditional HMC paradigm, the mass matrix is a constant, $\mathbf{M}$, which needs to be tuned for good performance, often simply set to the identity matrix. Clearly, when the dimensionality of $\mathbf{x}$ is high, tuning the elements in $\mathbf{M}$ is difficult, and using the identity matrix may lead to poor performance.

In the RMHMC method, the target distribution $p(\mathbf{x})$ is to be defined on a Riemann manifold. The mass matrix $\mathbf{G}(\mathbf{x})$ becomes a metric tensor on the manifold. Assume we have a conditional density function of data, $\mathbf{z}$, given parameters $\mathbf{x}$, $p(\mathbf{z}|\mathbf{x})$. The metric tensor is the expected Fisher information matrix:

$$\mathbf{G}(\mathbf{x}) = -\mathbb{E}_{p(\mathbf{z}|\mathbf{x})}\left[\frac{\partial^2}{\partial\mathbf{x}^2}\log\left(p(\mathbf{z}|\mathbf{x})\right)\right] = \mathrm{cov}\left[\frac{\partial}{\partial\mathbf{x}}\log\left(p(\mathbf{z}|\mathbf{x})\right)\right]. \tag{5.2}$$

Such an idea was initially proposed in Rao (1945) and triggered intensive studies on the use of Riemann geometry in statistical inference afterward (Amari & Nagaoka, 2000; Kass, 1989).

The Hamiltonian dynamical system, based on equation 5.1, is therefore given by

$$\begin{aligned}
\frac{dx_i}{d\tau} &= \frac{\partial H}{\partial p_i} = \{\mathbf{G}(\mathbf{x})^{-1}\mathbf{p}\}_i \\
\frac{dp_i}{d\tau} &= -\frac{\partial H}{\partial x_i} = \frac{\partial\mathcal{L}}{\partial x_i} - \frac{1}{2}\mathrm{tr}\left(\mathbf{G}(\mathbf{x})^{-1}\frac{\partial\mathbf{G}(\mathbf{x})}{\partial x_i}\right) + \frac{1}{2}\mathbf{G}(\mathbf{x})^{-1}\frac{\partial\mathbf{G}(\mathbf{x})}{\partial x_i}\mathbf{G}(\mathbf{x})^{-1}\mathbf{p}.
\end{aligned} \tag{5.3}$$

The system of partial differential equations, equation 5.3, is solved by a generalized leapfrog integrator, such that the properties of volume preservation and reversibility are maintained:

$$\mathbf{p}\left(\tau + \frac{\varepsilon}{2}\right) = \mathbf{p}(\tau) - \frac{\varepsilon}{2}\nabla_{\mathbf{x}}H\left(\mathbf{x}(\tau), \mathbf{p}\left(\tau + \frac{\varepsilon}{2}\right)\right) \tag{5.4}$$

$$\begin{aligned}
\mathbf{x}(\tau + \varepsilon) = \mathbf{x}(\tau) + \frac{\varepsilon}{2}\Big(&\nabla_{\mathbf{p}}H\left(\mathbf{x}(\tau), \mathbf{p}\left(\tau + \frac{\varepsilon}{2}\right)\right) \\
&+\nabla_{\mathbf{p}}H\left(\mathbf{x}(\tau + \varepsilon), \mathbf{p}\left(\tau + \frac{\varepsilon}{2}\right)\right)\Big)
\end{aligned} \tag{5.5}$$

$$\mathbf{p}(\tau+\varepsilon)=\mathbf{p}\left(\tau+\frac{\varepsilon}{2}\right)-\frac{\varepsilon}{2}\nabla_{\mathbf{x}}H\left(\mathbf{x}\left(\tau+\tau\right),\mathbf{p}\left(\tau+\frac{\varepsilon}{2}\right)\right). \tag{5.6}$$

These properties of the Hamiltonian system leave the target distribution invariant, thereby ensuring a correct MCMC algorithm.

Solutions to equations 5.4 to 5.6, which are obtained by fixed-point iterations in practice, yield a trajectory of position variable $\mathbf{x}$ and momentum variable $\mathbf{p}$. Let $\mathbf{x}^*$ and $\mathbf{p}^*$ denote the end of the trajectory, with $\mathbf{x}^*$ becoming the newly proposed variable. Let $\mathbf{x}^{(i-1)}$ and $\mathbf{p}$ be the starting pair of the trajectory, with $\mathbf{x}^{(i-1)}$, the previous sample. Then $\mathbf{x}^*$ is accepted or rejected according to the ratio

$$\min\left[1,\exp\left(-H(\mathbf{x}^*,\mathbf{p}^*)+H(\mathbf{x}^{(i-1)},\mathbf{p})\right)\right].$$

Note that when the metric tensor is not a function of the position $\mathbf{x}$, the generalized leapfrog integrator reduces to the standard leapfrog integrator of the HMC method. In this scenario, the RMHMC is the same as an HMC with an optimally tuned mass matrix.

For our application of sampling from the joint posterior $p(x_{0:K},\boldsymbol{\theta}|y_{1:K})$ of the SSPP model, we adopt the general Gibbs sampler paradigm, where RMHMC is applied in states sampling (which jointly updates the whole states sequence) and parameter sampling, respectively. The metric tensors in the two sampling stages have two different forms, discussed in the following subsections.

5.1 Metric Tensor for States. For sampling the states, the metric tensor of the likelihood is a diagonal matrix in which the entries on the diagonal are $\sum_{c=1}^{C}\beta_c^2\exp(\mu+\beta_c x_k)\Delta$. The negative Hessian of the log prior has the same form as stochastic volatility models. Therefore, the metric tensor G is a tridiagonl matrix whose diagonal elements are $\left[\frac{1}{\sigma_\varepsilon^2},\sum_{c=1}^{C}\beta_c^2\exp(\mu+\beta_c x_1)\Delta+\frac{1+\rho^2}{\sigma_\varepsilon^2},\ldots,\sum_{c=1}^{C}\beta_c^2\exp(\mu+\beta_c x_{K-1})\Delta+\frac{1+\rho^2}{\sigma_\varepsilon^2},\right.$ $\left.\sum_{c=1}^{C}\beta_c^2\exp(\mu+\beta_c x_K)\Delta+\frac{1}{\sigma_\varepsilon^2}\right]$. Elements on the superdiagonal and subdiagonal are $-\frac{1}{\sigma_\varepsilon^2}$.

Further, we integrate out the states, obtaining a constant metric tensor for sampling states. Therefore, the generalized leapfrog algorithm reduces to the standard one in HMC. The formulation of the metric tensor changes accordingly, in particular, the likelihood terms on the diagonal changes to $\sum_{c=1}^{C}\beta_c^2\exp(\mu+\beta_c\mathbb{E}[x_k]+\frac{\beta_c^2}{2}\mathrm{Var}[x_k])\Delta$, where $\mathbb{E}[x_k]$ and $\mathrm{Var}[x_k]$ denote the mean and variance of x_k and are obtained by equations 5.7 and 5.8, respectively.

5.2 Metric Tensor for Parameters. We consider only three parameters, ρ,α, and μ, while β_c and σ_ε^2 are fixed to ensure strong identifiability.

To constrain the AR process to be stable, ρ is subject to the transformation $\rho = \tanh(\gamma)$. We first obtain the expected value of states $\mathbb{E}[x_k]$ and $\mathrm{Var}[x_k]$:

$$\mathbb{E}[x_k] = \alpha(I_k + \rho I_{k-1} + \cdots + \rho^{k-1} I_1), \tag{5.7}$$

$$\mathrm{Var}[x_k] = \frac{\sigma_\varepsilon^2}{1-\rho^2}. \tag{5.8}$$

Hence, the nonzero terms of the metric tensor, equation 5.2, can be derived as

$$\mathbb{E}\left[\frac{\partial^2 \mathcal{L}}{\partial \gamma^2}\right] = -2\rho^2 - K(1-\rho^2) - \frac{1-\rho^2}{\sigma_\varepsilon^2} \sum_{k=1}^{K} \mathbb{E}[x_{k-1}]^2,$$

$$\mathbb{E}\left[\frac{\partial^2 \mathcal{L}}{\partial \gamma \alpha}\right] = -\frac{1-\rho^2}{\sigma_\varepsilon^2} \sum_{k=1}^{K} \mathbb{E}[x_{k-1}] I_k,$$

$$\mathbb{E}\left[\frac{\partial^2 \mathcal{L}}{\partial \alpha^2}\right] = -\sum_{k=1}^{K} \frac{I_k^2}{\sigma_\varepsilon^2},$$

$$\mathbb{E}\left[\frac{\partial^2 \mathcal{L}}{\partial \mu^2}\right] = -\sum_{k=0}^{K} \sum_{c=1}^{C} \exp(\mu + \beta_c \mathbb{E}[x_k] + \frac{1}{2}\beta_c^2 \mathrm{Var}[x_k])\Delta.$$

The derivatives of the above metric tensor terms with regard to each parameter, needed in the generalized leapfrog algorithm, are straightforward to carry out.

6 Numerical Results

In this section, we compare the MCMC methods for the SSPP model on three data sets—one synthetic and two real. The two real data sets used were obtained from the public repository neurodatabase.org, a resource funded by the Human Brain Project. All simulations were carried out with Matlab on an IntelCore 2 Quad Q6600 2.40 GHZ with 4 GB RAM computer.

6.1 Synthetic Data Set. First, we examine the efficiency of the three MCMC methods, PMMH, HMC, and RMHMC, with a benchmark method: single-site Gibbs sampler on a synthetic data set. Later, the best method in terms of standard efficiency measures will be compared with VB on experimental data sets.

The parameter settings used for generating the synthetic data set are shown in Table 1. We chose an observation length of $T = 20$ s, and time resolution $\Delta = 0.01$ s. The external stimulus was applied at regular intervals

Table 1: True Parameter Setting for Generating the Synthetic Data Set.

ρ	α	σ_ε^2	μ	$\beta_1, \ldots, \beta_{10}$	Channels	K
0.8	4	0.04	0	$0.9, \ldots, 1.1$	10	2,000

of 1 s. To ensure strong identifiability, we fixed β_c and σ_ε^2 to their true values. Hence, the inference task is focused on the states and parameters ρ, α, and μ. In addition, each of the three parameters is assigned a flat prior.

The implementation details of the four methods are as follows:

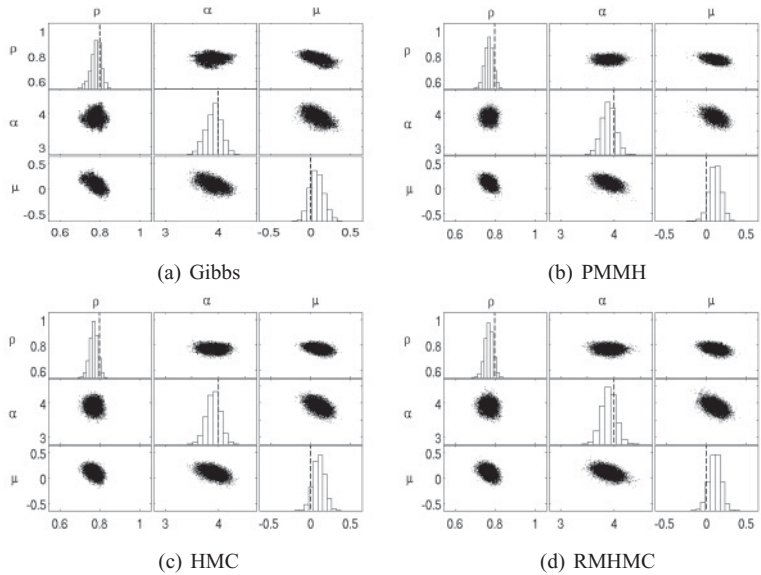
- Single-site Gibbs uses the state transition density proposal for each state and random walk proposals for the parameters, in particular, $\mathcal{N}(\theta^{(i-1)}, 0.01^2)$ for ρ and $\mathcal{N}(\theta^{(i-1)}, 0.1^2)$ for both α and μ. (For more details on the conditional distributions, see appendix C in Zammit Mangion et al., 2011.)
- PMMH uses the same proposals for parameters as the single-site Gibbs sampler. The particles of SMC algorithm are proposed by the state transition density with a population of 1000.
- HMC uses an identity mass matrix that is further scaled by step sizes. Specifically, in the state sampling stage, we employ 34 integration steps with a step size of 0.03 . For the parameters, 67 integration steps with each step size of 0.015 are chosen. On top of those settings, we also use random integration directions to ensure reversibility.
- RMHMC uses a step size of 0.2 and 25 integration steps for the states and a step size of 0.8 and 5 integration steps for parameters. Again, a random integration direction is applied at each generalized leapfrog loop.

HMC is tuned in the light of making a trade-off between acceptance rate and number of the leapfrog steps within each Monte Carlo iteration. In other words, we aim to integrate over a certain distance with a small number of integration steps, without rejecting too many proposals. We tuned RMHMC in the same spirit. In addition, simulations show that, RMHMC, benefiting from the use of local geometric structure, with the same number of integration steps, is able to make much larger moves while maintaining a high acceptance rate, consistent with the findings in Girolami and Calderhead (2011). Based on this, one can achieve fast mixing with fewer integration steps. With the above settings, as expected, the acceptance rates of HMC and RMHMC shown in Table 2 are much higher than the other two random walk proposal-based methods.

Figure 1 shows the posterior distributions obtained by each of the four MCMC methods. While PMMH, HMC, and RMHMC faithfully capture the posterior distributions, the single-site Gibbs sampler produces some ad

Table 2: Acceptance Rates of All Five Methods.

Gibbs	PMMH	HMC	RMHMC
30%−60%	40%−55%	80%−90%	85%−99%



(a) Gibbs (b) PMMH

(c) HMC (d) RMHMC

Figure 1: Full posterior distribution of parameters obtained by four methods, where the true value of each parameter is indicated by a dashed line. There are 20,000 (after 1000 burn-in) posterior samples for each of the three parameters.

hoc shapes within the clouds of samples, implying that the chosen burn-in period is not sufficiently long.

In addition to the posterior profile, by comparing the $\sqrt{\hat{R}}$ statistic from Gelman and Rubin (1992), we further assess each method on the time for convergence to the stationary distribution in Figure 2. This test is carried out by considering five chains with different initializations. Since we have 2004 variables, we show only the statistics for parameters that capture the overall convergence status well. We observe that the Markov chain obtained by a single-site Gibbs sampler is poorly mixed in ρ, whereas RMHMC consistently shows the fastest convergence performance.

Table 3 shows relative performances of the MCMC methods in terms of effective sample size (ESS) and processing time. The appendix (see Table 4) gives general estimates of the comuptational complexities of the algorithms. Such criteria (ESS and processing time) were also used in Girolami and

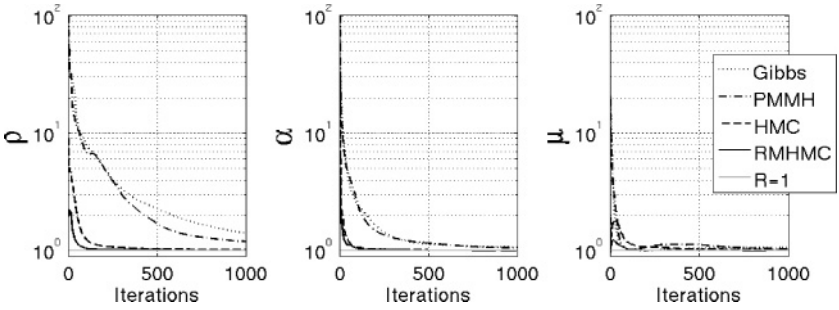


Figure 2: Logarithm $\sqrt{\hat{R}}$ statistics (see Gelman & Rubin, 1992) for ρ, α, and μ. Convergence corresponds to an $\sqrt{\hat{R}}$ value close to 1.

Table 3: ESS and Processing Time Comparison Based on 20,000 Posterior Samples (1000 Burn-In) of States and Parameters Obtained by Single-Site Gibbs, PMMH, HMC, and RMHMC on Synthetic Data Set.

Methods	ESS (ρ, α, μ)	States ESS (Min, Median, Max)	Time(s)
Gibbs	16, 94, 38	46, 98, 210	2594
PMMH	458, 939, 1055	567, 2132, 5129	11,5341
HMC	340, 1590, 930	1152, 4045, 10,979	4225
RMHMC	1072, 1593, 2326	4060, 20,000, 20,000	3136

Note: Each attribute is averaged over 10 runs.

Calderhead (2011) (Liu, 2001, for more details on ESS). In order to make a fair assessment, each method is run 10 times on the same data set and averages tabulated. We note that RMHHC shows the highest ESS scores for both states and parameters and ranks second in processing speed. HMC shows the second-highest ESS score on states, yet the parameter ESS (in ρ and μ) is similar to PMMH. Further, all methods show significant improvement on ESS when compared to the baseline single-site Gibbs sampler. Finally, results on autocorrelation function (ACF) performance in Figure 3 also lend additional supports to the findings.

In summary, from the comparisons carried out on a synthetic data set, we conclude that RMHMC is a clear winner in terms of its sampling and computational efficiencies. The performance of PMMH can be further improved by increasing the number of particles used in the SMC stage or adding sophisticated tricks like auxiliary variables (Pitt & Shephard, 1999) and resample-move algorithm (Gilks & Berzuini, 2001). The computational costs of PMMH, however, higher by a factor of three in the current setting, make the PMMH strand less appealing. Such costs could be even higher

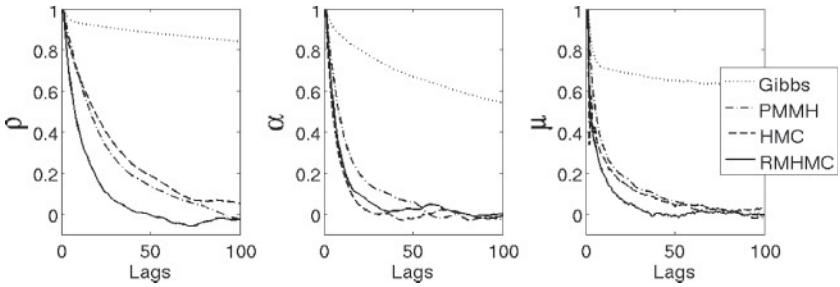


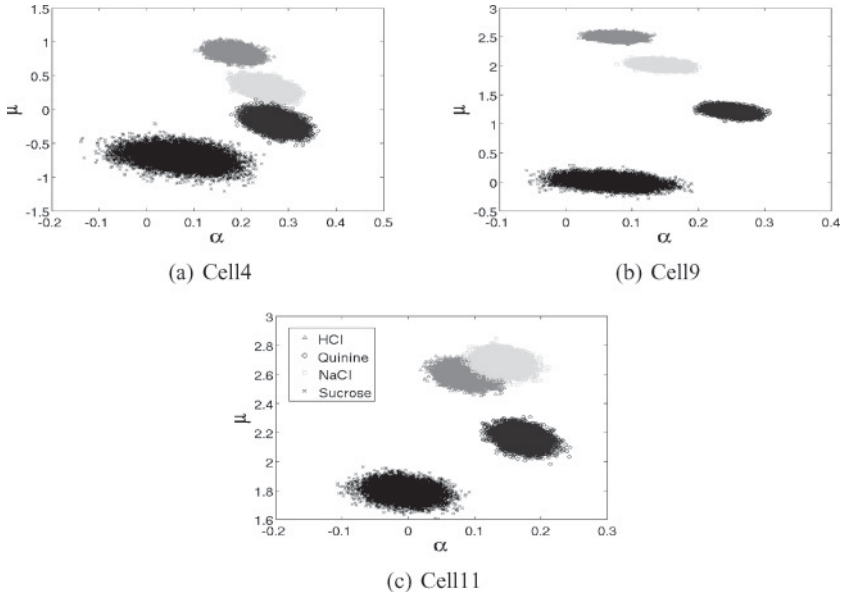
Figure 3: The first 100 lags of autocorrelation values of different MCMC methods for each parameter. RMHMC outperforms other methods in ρ and μ, whereas in α, HMC drops faster than others, indicating that a unit tensor in α may be appropriate.

with real applications, in which the length of data records may be substantial in comparison to the synthetic data we have used.

6.2 Modeling Taste Response. This section is a study of the performance of RMHMC on a rat spike train data set in which the firing pattern of a single cell has been measured under different taste stimuli. The details of the experiment can be found in Di Lorenzo and Victor (2003). Briefly, four taste stimuli are considered: NaCl, sucrose, quinine HCl, and HCl, inducing salty, sweet, sour, and bitter tastes. Under each stimulus, recording trials are separated by a 20 s rinsing and a 1.5 min wait, and each trial consists of a 10 s baseline period with no stimulus, 5 s presentation of stimulus, and 5 s wait.

Recently Zammit Mangion et al. (2011) examined this data set with the SSPP model with an online VB inference framework and showed the ability to detect sudden changes on the model parameters in response to changes in stimuli. The preprocessing steps follow Zammit Mangion et al. (2011), where the 10 s baseline period is considered in the analysis and trials associated with each tastant are concatenated to form one contiguous spike train. Due to the response latency and a linear increase on firing rate for the first 250 ms after each stimulus in the data set, which is noted in both Di Lorenzo and Victor (2003) and Zammit Mangion et al. (2011), a temporal rectangular window of 250 ms is applied at the beginning of each 10 s segment. The time resolution is set to 10 ms, which resulted in a small number of bins containing more than one spike. This was adjusted by moving the spike to the nearest empty bin forward in time. The resulting data contained 23,000 time points in cell 9 and 16,000 time points in cells 4 and 11.

As Zammit Mangion et al. (2011) showed, the input gain α and background firing rate μ play a role as characteristic features for the classification of different tastants. Together with the hidden states, these attributes



(a) Cell4 (b) Cell9

(c) Cell11

Figure 4: Posterior distributions of α and μ given the observed spike trains in cells 4, 9, and 11. The parameter space shows good separation of the four tastes.

dynamically separate the firing rate into two major contributors: background noise and underlying neural dynamics, which is driven by the external stimulus. Such a separation makes classification easier when the firing rate in itself cannot discriminate between the tastants. Therefore, the inference we target is the posterior distributions of α, μ, and the underlying states, given the observed spike train, with the other parameters fixed at : $\rho = 0.97$, $\sigma_\varepsilon^2 = 0.05$, and $\beta = 0.5$. Figure 4 shows results of this, and it can be seen that the separation we aimed for is convincingly achieved by the model.

We also assess the model goodness of fit in Figures 5 and 6 using the time-rescaled theorem-based KS test (for details, see Brown, Barbieri, Ventura, Kass, & Frank, 2001) and find that while RMHMC obtained a slightly better fit in cell 4, the results are similar to VB in cells 9 and 11.

In Figure 7, we show the expected spiking probability over states and parameter posteriors obtained from VB (online and offline) and RMHMC. Note the increases expected probability synchronous with the appearence of spikes. This comparison suggests that in data with a significant number of spikes, VB appears to perform better and the MCMC approach is better suited to data in which the spikes are sparse. Our intuition on this is that when spike count is low, the uncertainty within the posterior is relatively high (see Figures 4a and 4b). MCMC therefore is more flexible to handle

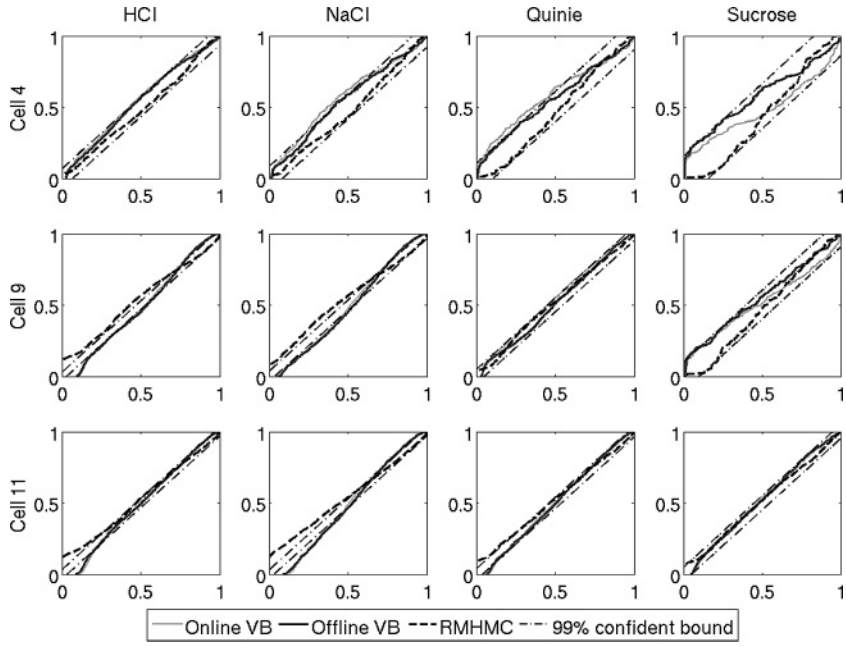


Figure 5: Q-Q plot based on time rescaling theorem (Brown et al., 2001) of inferred model by RMHMC, offline VB, and online VB. The x-axis shows the quantiles, and the y-axis shows an empirical cumulative rate function. Ninety-nine percent confident intervals are indicated by the dashed line in each figure. A 45 degree line indicates a perfect match.

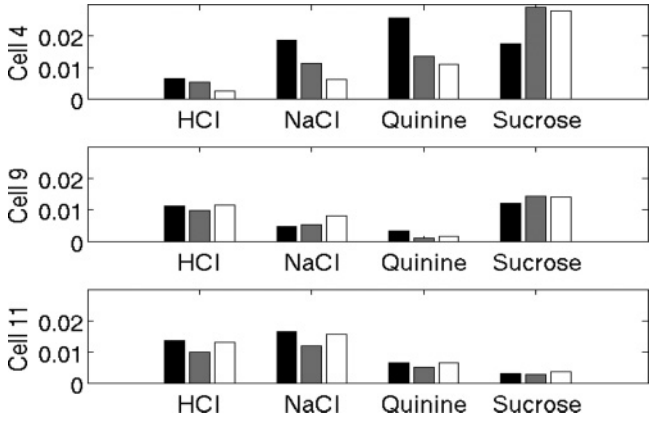


Figure 6: Maximum KS distance for each cell with each taste stimulus. Each block of vertical bars corresponds to offline VB, online VB, and RMHMC, from left to right.

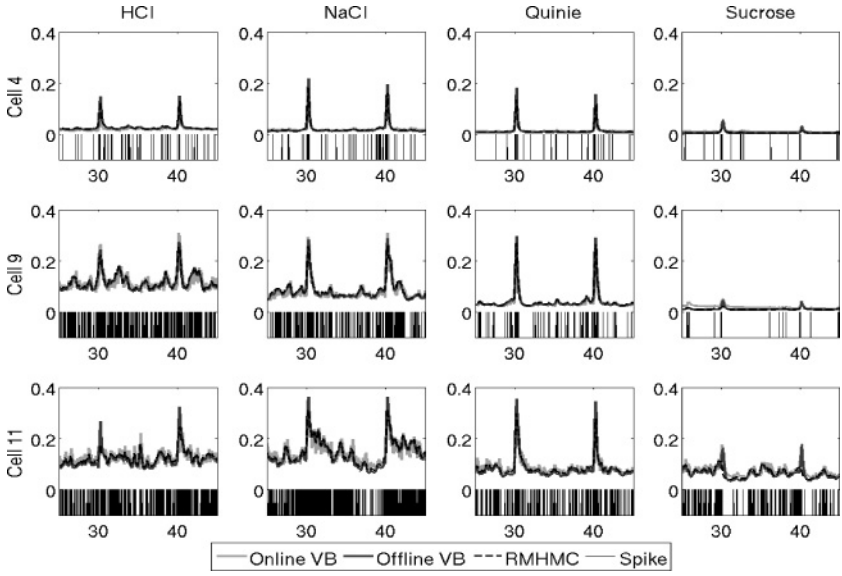


Figure 7: A 20 s segment expected spiking probability with respect to state and parameter posteriors obtain by RMHMC, offline VB, and online VB (graphs overlap because the differences among the methods are small). For each panel, the x-axis denotes time with unit in seconds, and the y-axis denotes the expected spiking probability measure. The observed spike train is also shown in black bars.

the uncertainty. VB methods, on the other hand, often underestimate the uncertainty within the state transition process due to the independent assumption of the mean field approximation (Turner & Sahani, 2010). Note that the number of data points is large in this data set, and given the fact that the posteriors are unimodal, it is reasonable to expect MCMC and VB to show similar performance.

The next section shows results from a different data set in which the data record is much shorter in time and spiking is sparse.

6.3 Parvocellular Neuron Data Set. We now consider another data set from Victor et al. (2007), where the response variability of marmoset parvocellular neurons under drifting sinusoidal luminance gratings stimulus is considered. Single cell spiking activities are recorded, where the luminance modulation (LUM) stimuli are presented at 10 different ascending contrast levels.[1] Each contrast is repeated 13 times within a 3.5 s period for three

[1] 0, 0.0156, 0.0312, 0.0625, 0.0937, 0.125, 0.25, 0.375, 0.5 and 1. Data are from cell MY107.

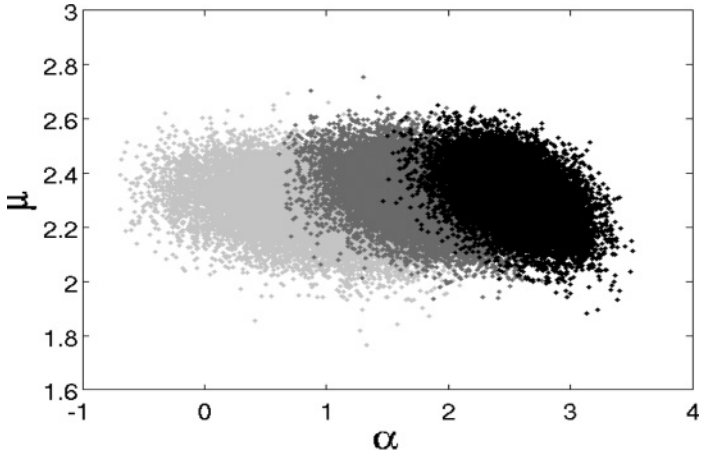


Figure 8: Joint α and μ posteriors. Clusters from right to left correspond to contrast values of 1, 0.5 and 0.35.

trials. We treat the three trials as three parallel channels of spike trains driven by the same stimulus. The time resolution is set to 0.002 s, which guarantees one spike per time bin and yields 1750 time points for each channel.

Similar to the previous example, we use RMHMC to target the posterior distributions of α, μ, and hidden states given observed spike trains. We fix $\rho = 0.8$, $\sigma_\varepsilon^2 = 0.05$, and $\beta = 1$ for each channel.

As shown in Figure 8, the resulting posteriors overlap heavily and therefore are not easy to distinguish between trials with different stimulus types. However, the inferred model is still able to characterize the data quite well according to the KS test results, as shown in Figures 9 and 10. In this case, there are significantly fewer data than in the taste response data set considered previously. RMHMC consistently outperforms both EM and VB in terms of the model goodness of fit across each of the 10 contrast levels. Finally, the expected spiking probabilities are consistent with the data (see Figure 11).

7 Discussion

In this work, we study Bayesian inference and learning in a state-space model with point process observations (SSPP) with a wide range of state-of-the-art MCMC methods. While all methods we considered converge and produce the correct inference, their efficiencies differ significantly, with RMHMC outperforming the others. The reason is that by using the gradient of the posterior and benefiting from the volume preservation properties of

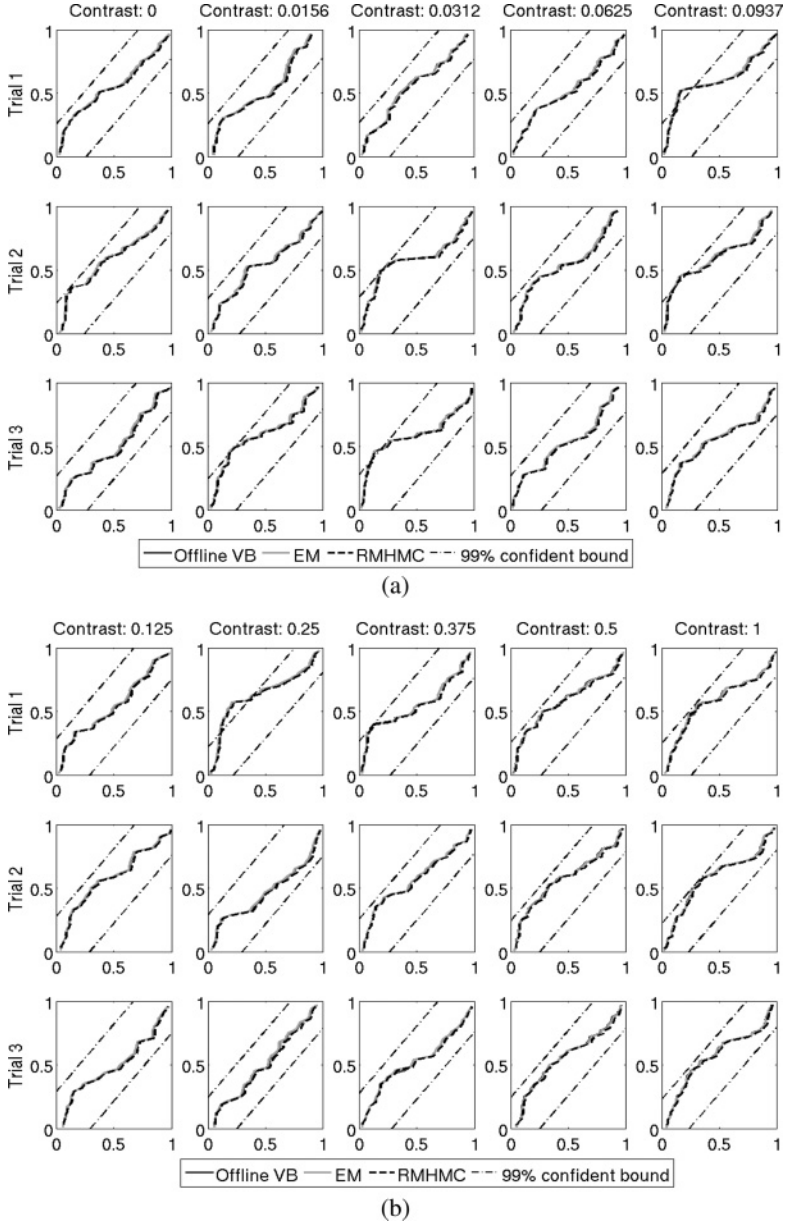


Figure 9: Q-Q plot based on time rescaling theorem. (a, b) Results on contrast from 0 to 0.0937 and 0.125 to 1, respectively. Offline VB, EM, and RMHMC are drawn and overlap heavily. The 99% confidence intervals are shown as dashed lines.

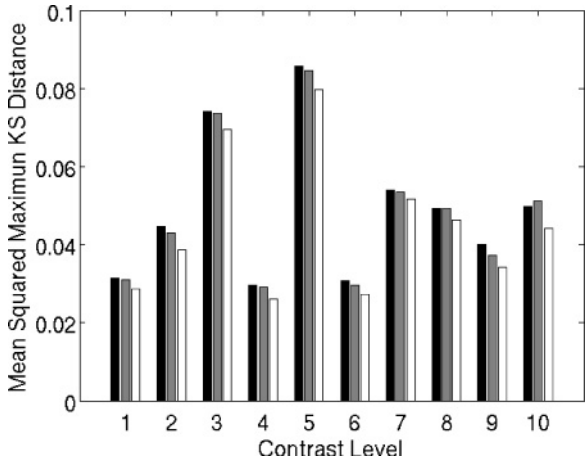


Figure 10: The mean squared maximum KS distance of each contrast level. Contrast levels 1 to 10 denote 0.0156 to 1 in the data set. Each block of vertical bars corresponds to EM, offline VB, and RMHMC from left to right.

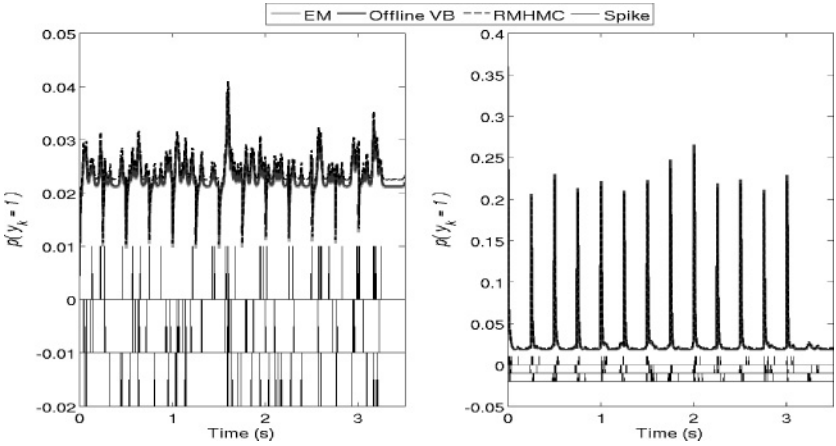


Figure 11: Expected spiking probability with respect to state and parameter posteriors obtained by RMHMC, EM, and offline VB. The left panel and right panels correspond to contrasts of 0 and 1.

the Hamiltonian dynamic system, RMHMC is able to propose large moves while maintaining a high acceptance rate. These moves are guided by a metric tensor, which takes advantage of the underlying manifold structure of the posterior distributions. As for the state-space models and SSPP in

particular, the metric tensor takes the form of an expected Fisher information matrix, which is analytically available. Moreover, due to the previously noted unimodality property (Yuan & Niranjan, 2010) of the SSPP model, such a metric tensor is guaranteed to be positive definite, which justifies the suitability of using RMHMC.

In addition to the synthetic example, we compared the performance of the RMHMC with variational Bayes (VB) on a rat taste stimuli data set and monkey parvocellular neuron data set. Results on these experiments suggest that the advantage gained using RMHMC is pronounced on short-duration data sets with small numbers of observed spikes. This may be of interest in analyzing nonstationary data using short windows in time. For long time series, VB methods seem preferable, since they offer an attractive balance between computational cost and estimation accuracy. However, one should note that the SSPP model may be customized for different modeling tasks. Different choices for the conditional intensity function may lead to VB being intractable. MCMC methods, on the other hand, have the flexibility of adapting to such complex scenarios.

Relating the framework described in this letter, another popular model for characterizing neural spike trains is the point process generalized linear model (Truccolo, Eden, Fellows, Donoghue, & Brown, 2005; Okatan, Wilson, & Brown, 2005), in which the stimuli are treated as canonical parameters in a likelihood model similar to the one considered in SSPP. For inference on this model, Paninski (2004) provides a maximum likelihood formulation, whereas for the Bayesian perspective, both the MCMC and VB approaches have recently been studied and shown good results on decoding the spike trains (Ahmadian, Pillow, & Paninski, 2011; Chen, Kloosterman, Wilson, & Brown, 2010). The SSPP differs from such a paradigm by assuming a latent dynamic process that takes the stimuli as input sources, resulting in a physiologically plausible parameterization. Its inference frameworks including EM (Smith & Brown, 2003), VB (Zammit Mangion et al., 2011), and MCMC discussed in this work, also show good performance on decoding the spike train, while the inferred parameters may serve as discriminant attributes of different physiological states.

For spike train classification, Salimpour, Soltanian-Zadeh, Salehi, Emadi, and Abouzari (2011) show an interesting approach of using the likelihood based on filtered estimates as a discriminator for spike trains responding to various stimuli. Their model treats parameters of the CIF as states in deriving an extended Kalman filter estimator. This differs from the model we considered, which has a separate underlying dynamical state process. Algorithmically, Salimpour et al.'s (2011) work has similarities to the Laplace approximation-based adaptive filters (Smith & Brown, 2003; Eden, Frank, Barbieri, Solo, & Brown, 2004; Koyama, Castellanos Pérez-Bolde, Shalizi, & Kass, 2010).

These adaptive filters for state estimation in the SSPP model are based on the idea of sequentially performing Laplace approximations to the

Table 4: General Computational Cost Comparison.

Gibbs	PMMH	HMC	RMHMC
$\mathcal{O}((K+1)M)$	$\mathcal{O}(NKM)$	$\mathcal{O}((L_1K+L_2)M)$	$\mathcal{O}((L_1^*K+L_2^*)M)$

filtering density. Together with other nonlinear filtering methods like unscented Kalman filter (UKF) (Julier & Uhlmann, 1997; Wan & Van Der Merwe, 2000), they have the potential for constructing efficient proposals for state updating in PMMH. Such an idea has been explored in the context of SMC (Ergün et al., 2007; Wang et al., 2009). Further, these filtering methods can be used to obtain Laplace approximation and variational lower bound of the marginal likelihood, resulting in approximate sampling methods.

As another future extension, instead of the random walk proposals, it is possible to use gradient-based proposals to improve the acceptance rate in PMMH, for example, the Metropolis adjusted Langevin algorithm (MALA), manifold MALA method (Girolami & Calderhead, 2011), HMC, and RMHMC. Despite the severe computational overheads, these ideas are algorithmically attractive, since they offer highly efficient proposal mechanisms to tackle problems with high correlations between states and parameters. We are currently pursuing these avenues.

Appendix: Computation Complexity

Here we give estimates of the orders of computational complexities of the different sampling algorithms used. Let K and M be the time points and the number of Monte Carlo iterations, respectively. N denotes the number of particles used in PMMH. L_1 and L_2 are the number of leapfrog steps for states and parameters in HMC. The superscript $*$ indicates that the number of leapfrog steps in RMHMC is different from those in HMC. In addition, we assume the cost of each parameter updating and other inner calculations to be 1. With these notations, Table 4 shows the estimated complexities.

Acknowledgments

K.Y. is supported by a studentship from the University of Southampton. M.G. acknowledges EPSRC Advanced Fellowship EP/E052029/2, project grants EP/E032745/2 and EP/F009429/2, and the BBSRC project grant: The Silicon Trypanasome. We are grateful for constructive comments from two anonymous reviewers.

References

Ahmadian, Y., Pillow, J. W., & Paninski, L. (2011). Efficient Markov chain Monte Carlo methods for decoding neural spike trains. *Neural Computation*, *23*(1), 46–96.

Amari, S., & Nagaoka, H. (2000). *Methods of information geometry*. New York: Oxford University Press.

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

Brown, E. N., Barbieri, R., Eden, U. T., & Frank, L. M. (2002). Likelihood methods for neural data analysis. In J. Feng (Ed.), *Computational neuroscience: A Comprehensive approach* (pp. 253–286). London: CRC.

Brown, E. N., Barbieri, R., Ventura, V., Kass, R. E., & Frank, L. M. (2001). The time-rescaling theorem and its application to neural spike train data analysis. *Neural Computation*, *14*(14), 325–346.

Carter, C. K., & Kohn, R. (1994). On Gibbs sampling for state space models. *Biometrika*, *81*(3), 541–553.

Chen, Z., Kloosterman, F., Wilson, M. A., & Brown, E. N. (2010). Variational Bayesian inference for point process generalized linear models in neural spike trains analysis. In *Proceedings of the 2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)* (pp. 2086–2089). Piscataway, NJ: IEEE.

Daley, D., & Vere-Jones, D. (2003). *An introduction to the theory of point process* (2nd ed.). New York: Springer-Verlag.

Di Lorenzo, P., & Victor, J. (2003). Taste response variability and temporal coding in the nucleus of the solitary tract of the rat. *Journal of Neurophysiology*, *90*, 1418–1431.

Doucet, A., de Freitas, N., & Gordon, N. (2001). *Sequential Monte Carlo methods in practice*. New York: Springer-Verlag.

Duane, S., Kennedy, A. D., Pendleton, B. J., & Roweth, D. (1987). Hybrid Monte Carlo. *Physics Letters B*, *195*(2), 216–222.

Eden, U. T., Frank, L. M., Barbieri, R., Solo, V., & Brown, E. N. (2004). Dynamic analysis of neural encoding by point process adaptive filtering. *Neural Computation*, *16*(5), 971–998.

Ergün, A., Barbieri, R., Eden, U. T., Wilson, M. A., & Brown, E. N. (2007). Construction of point process adaptive filter algorithms for neural systems using sequential Monte Carlo methods. *IEEE Transactions on Biomedical Engineering*, *54*(3), 419–428.

Fearnhead, P. (2010). MCMC for state space models. In S. Brooks, A. Gelman, G. Jones, & X.-L. Meng (Eds.), *Handbook of Markov chain Monte Carlo*. Boca Raton, FL: CRC.

Gelman, A., & Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences (with discussion). *Statistical Science*, *7*(4), 457–472.

Geweke, J., & Tanizaki, H. (2001). Bayesian estimation of state-space models using the Metropolis-Hasting algorithm within Gibbs sampling. *Computational Statistics and Data Analysis*, *37*(2), 151–170.

Gilks, W., & Berzuini, C. (2001). Following a moving target: Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *63*(1), 127–146.

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

Ivanov, P., Rosenblum, M., Peng, C., Mietus, J., Havlin, S., Stanley, H., et al. (1996). Scaling behaviour of heartbeat intervals obtained by wavelet-based time-series analysis. *Nature*, *383*, 323–327.

Jolivet, R., Kobayashi, R., Rauch, A., Naud, R., Shinomoto, S., & Gerstner, W. (2008). A benchmark test for a quantitative assessment of simple neuron models. *Journal of Neuroscience Methods*, *169*, 417–424.

Julier, S., & Uhlmann, J. (1997). A new extension of the Kalman filter to nonlinear systems. In *Proceedings of the Int. Symp. Aerospace/Defense Sensing, Simul. and Controls* (vol. 3, p. 26). Bellingham, WA: SPIE.

Kass, R. E. (1989). The geometry of asymptotic inference. *Statistical Science*, *4*, 188–234.

Koyama, S., Castellanos Pérez-Bolde, L., Shalizi, C. R., & Kass, R. E. (2010). Approximate methods for state-space models. *Journal of the American Statistical Association*, *105*(489), 170–180.

Liu, J. S. (2001). *Monte Carlo strategies in scientific computing*. New York: Springer-Verlag.

Neal, R. M. (1993). *Probabilistic inference using Markov chain Monte Carlo methods* (Techn. Rep. No. CRG-TR-93-1). Toronto: University of Toronto.

Neal, R. M. (2010). MCMC using Hamiltonian dynamics. In S. Brooks, A. Gelman, G. Jones, & X.-L. Meng (Eds.), *Handbook of Markov chain Monte Carlo*. Boca Raton, FL: CRC.

Okatan, M., Wilson, M. A., & Brown, E. N. (2005). Analyzing functional connectivity using a network likelihood model of ensemble neural spiking activity. *Neural Computation*, *17*(9), 1927–1961.

Paninski, L. (2004). Maximum likelihood estimation of cascade point-process neural encoding models. *Network: Computation in Neural Systems*, *15*(4), 243–262.

Pitt, M., & Shephard, N. (1999). Filtering via simulation: Auxiliary particle filters. *Journal of the American Statistical Association*, *94*, 590–599.

Rao, C. R. (1945). Information & accuracy attainable in the estimation of statistical parameters. *Bull. Calc. Math. Soc.*, *37*, 81–91.

Salimpour, Y., Soltanian-Zadeh, H., Salehi, S., Emadi, N., & Abouzari, M. (2011). Neuronal spike train analysis in likelihood space. *PLoS ONE*, *6*(6), e21256.

Scott, S. L. (2002). Bayesian methods for hidden Markov models: Recursive computing in the 21st century. *Journal of the American Statistical Association*, *97*(457), 337–351.

Shephard, N., & Pitt, M. K. (1997). Likelihood analysis of non-gaussian measurement time series. *Biometrika*, *84*(3), 653–667.

Smith, A. C., & Brown, E. N. (2003). Estimating a state-space model from point process observations. *Neural Computation*, *15*(5), 965–991.

Truccolo, W., Eden, U. T., Fellows, M. R., Donoghue, J. P., & Brown, E. N. (2005). A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. *Journal of Neurophysiology*, *93*(2), 1074–1089.

Turner, R., & Sahani, M. (2010). Two problems with variational expectation maximisation for time-series models. In D. Barber, A.-T. Cemgil, & S. Chiappa (Eds.), *Inference and learning in dynamic models*. Cambridge: Cambridge University Press.

Victor, J., Blessing, E., Forte, J., Buzás, P., & Martin, P. (2007). Response variability of marmoset parvocellular neurons. *Journal of Physiology*, *579*(1), 29–51.

Wan, E., & Van Der Merwe, R. (2000). The unscented Kalman filter for nonlinear estimation. In *Proceedings of the Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000* (pp. 153–158). Piscataway, NJ: IEEE.

Wang, Y., Paiva, A.R.C., Príncipe, J. C., & Sanchez, J. C. (2009). Sequential Monte Carlo point-process estimation of kinematics from neural spiking activity for brain-machine interfaces. *Neural Computation*, *21*(10), 2894–2930.

Yuan, K., & Niranjan, M. (2010). Estimating a state-space model from point process observations: A note on convergence. *Neural Computation*, *22*(8), 1993–2001.

Zammit Mangion, A., Yuan, K., Kadirkamanathan, V., Niranjan, M., & Sanguinetti, G. (2011). Online variational inference for state-space models with point process observations. *Neural Computation*, *23*(8), 1967–1999.

Received July 9, 2011; accepted November 29, 2011.