Large-Scale Stochastic Sampling from the Probability Simplex

Jack Baker
STOR-i CDT, Deptment of Mathematics and Statistics
Lancaster University
`j.baker1@lancaster.ac.uk`

Paul Fearnhead
Department of Mathematics and Statistics
Lancaster University

Emily B. Fox
Department of Statistics
University of Washington

Christopher Nemeth
Department of Mathematics and Statistics
Lancaster University

Abstract

Stochastic gradient Markov chain Monte Carlo (SGMCMC) has become a popular method for scalable Bayesian inference. These methods are based on sampling a discrete-time approximation to a continuous time process, such as the Langevin diffusion. When applied to distributions defined on a constrained space, such as the simplex, the time-discretisation error can dominate when we are near the boundary of the space. We demonstrate that while current SGMCMC methods for the simplex perform well in certain cases, they struggle with *sparse simplex spaces*; when many of the components are close to zero. However, most popular large-scale applications of Bayesian inference on simplex spaces, such as network or topic models, are sparse. We argue that this poor performance is due to the biases of SGMCMC caused by the discretization error. To get around this, we propose the stochastic CIR process, which removes all discretization error and we prove that samples from the stochastic CIR process are asymptotically unbiased. Use of the stochastic CIR process within a SGMCMC algorithm is shown to give substantially better performance for a topic model and a Dirichlet process mixture model than existing SGMCMC approaches.

1 Introduction

Stochastic gradient Markov chain Monte Carlo (SGMCMC) has become a popular method for scalable Bayesian inference (Welling and Teh, 2011; Chen et al., 2014; Ding et al., 2014; Ma et al., 2015). The foundation of SGMCMC methods are a class of continuous processes that explore a target distribution—e.g., the posterior—using gradient information; these processes converge to a Markov chain which samples from the posterior distribution exactly. SGMCMC methods replace the costly full-data gradients with minibatch-based stochastic gradients, which provides one source of error. Another source of error arises from the fact that the continuous processes are almost never tractable to simulate; instead, discretizations are relied upon. In the non-SG scenario, the discretization errors are corrected for using Metropolis-Hastings corrections. However, this is not (generically) feasible in the SG setting. The result of these two sources of error is that SGMCMC targets an approximate posterior (Welling and Teh, 2011; Teh et al., 2016; Vollmer et al., 2016).

Preprint. Work in progress.

Another significant limitation of SGMCMC methods is that they struggle to sample from constrained spaces. Naively applying SGMCMC can lead to invalid, or inaccurate values being proposed. The result is large errors near the boundary of the space (Patterson and Teh, 2013; Ma et al., 2015; Wenzhe Li, 2016). A particularly important constrained space is the simplex space, which is used to model discrete probability distributions. A parameter ω of dimension d lies in the simplex if it satisfies the following conditions: $\omega_j \geq 0$ for all $j = 1, \ldots, d$ and $\sum_{j=1}^{d} \omega_j = 1$. Many popular models contain simplex parameters. For example, latent Dirichlet allocation (LDA) is defined by a set of topic-specific distributions on words and document-specific distributions on topics. Probabilistic network models often define a link probability between nodes. More generally, mixture and mixed membership models have simplex-constrained mixture weights; even the hidden Markov model can be cast in this framework with simplex-constrained transition distributions. As models become large-scale, these vectors ω often become *sparse*–i.e., many ω_j are close to zero—pushing them to the boundaries of the simplex. All the models mentioned have this tendency. For example in network data, nodes often have relatively few links compared to the size of the network, e.g., the number of friends the average social network user has will be small compared with the size of the whole social network. In these cases the problem of sampling from the simplex space becomes even *harder*; since many values will be very close to the boundary of the space.

Patterson and Teh (2013) develop an improved SGMCMC method for sampling from the probability simplex: stochastic gradient Riemannian Langevin dynamics (SGRLD). The improvements achieved are through an astute transformation of the simplex parameters, as well as developing a Riemannian (see Girolami and Calderhead, 2011) variant of SGMCMC. This method achieved state-of-the-art results on an LDA model. However, we show despite the improvements over standard SGMCMC, the discretization error of this method still causes problems on the simplex. In particular, it leads to asymptotic biases which dominate at the boundary of the space and causes significant inaccuracy.

To counteract this, we design an SGMCMC method based on the Cox-Ingersoll-Ross (CIR) process. The resulting process, which we refer to as the stochastic Cox-Ingersoll-Ross process (SCIR), has *no discretization error*. This process can be used to simulate from gamma random variables directly, which can then be moved into the simplex space using a well known, standard transformation. The CIR process has a lot of nice properties. One is that the transition equation is known exactly, which is what allows us to simulate from the process without discretization error. We are also able to characterize important theoretical properties of the SCIR algorithm, such as the non-asymptotic moment generating function, and thus its mean and variance.

We demonstrate the impact of this SCIR method on a broad class of models. Included in these experiments is the development of a scalable sampler for Dirichlet processes, based on the slice sampler of Walker (2007); Papaspiliopoulos (2008); Kalli et al. (2011). To our knowledge the application of SGMCMC methods to Bayesian nonparametric models has not been explored, and we consider this a further contribution of the article. All proofs in this article are relegated to the Supplementary Material. All code for the experiments will be made available online, and full details of hyperparameter and tuning constant choices has been detailed in the Supplementary Material.

2 Stochastic Gradient MCMC on the Probability Simplex

2.1 Stochastic Gradient MCMC

Consider Bayesian inference for continuous parameters $\theta \in \mathbb{R}^d$ based on data $\mathbf{x} = \{x_i\}_{i=1}^N$. Denote the density of x_i as $p(x_i|\theta)$ and assign a prior on θ with density $p(\theta)$. The posterior is then defined, up to a constant of proportionality, as $p(\theta|\mathbf{x}) \propto p(\theta) \prod_{i=1}^N p(x_i|\theta)$, and has distribution π. We define $f(\theta) := -\log p(\theta|\mathbf{x})$. Whilst MCMC can be used to sample from π, such algorithms require access to the full data set at each iteration. Stochastic gradient MCMC (SGMCMC) is an approximate MCMC algorithm that reduces this per-iteration computational and memory cost by using only a small subset of data points at each step.

The most common SGMCMC algorithm is stochastic gradient Langevin dynamics (SGLD), first introduced by Welling and Teh (2011). This sampler uses the Langevin diffusion, defined as the solution to the stochastic differential equation

$$d\theta_t = -\nabla f(\theta_t)dt + \sqrt{2}dW_t, \tag{2.1}$$

where W_t is a d-dimensional Wiener process. Similar to MCMC, the Langevin diffusion defines a Markov chain whose stationary distribution is π.

Unfortunately, simulating from (2.1) is rarely possible, and the cost of calculating ∇f is $O(N)$ since it involves a sum over all data points. The idea of SGLD is to introduce two approximations to circumvent these issues. First, the continuous dynamics are approximated by discretizing them, in a similar way to Euler's method for ODEs. This approximation is known as the *Euler-Maruyama* method. Next, in order to reduce the cost of calculating ∇f, it is replaced with a cheap, unbiased estimate. This leads to the following update equation, with user chosen stepsize h

$$\theta_{m+1} = \theta_m - h\nabla \hat{f}(\theta) + \sqrt{2h}\eta_m, \qquad \eta_m \sim N(0,1). \tag{2.2}$$

Here, $\nabla \hat{f}$ is an unbiased estimate of ∇f whose computational cost is $O(n)$ where $n \ll N$. Typically, we set $\nabla \hat{f}(\theta) := -\nabla \log p(\theta) - N/n \sum_{i \in S_m} \nabla \log p(x_i|\theta)$, where $S_m \subset \{1, \ldots, N\}$ resampled at each iteration with $|S_m| = n$. Applying (2.2) repeatedly defines a Markov chain that approximately targets π (Welling and Teh, 2011). There are a number of alternative SGMCMC algorithms to SGLD (Chen et al., 2014; Ding et al., 2014; Ma et al., 2015), based on approximations to other diffusions that also target the posterior distribution.

Recent work has investigated reducing the error introduced by approximating the gradient using minibatches (Dubey et al., 2016; Nagapetyan et al., 2017; Baker et al., 2017; Chatterji et al., 2018). While, by comparison, the discretization error is generally smaller, in this work we investigate an important situation where it degrades performance considerably.

2.2 SGMCMC on the Probability Simplex

We aim to make inference on the simplex parameter ω of dimension d, where $\omega_j \geq 0$ for all $j = 1, \ldots, d$ and $\sum_{j=1}^d \omega_j = 1$. We assume we have categorical data $\mathbf{z}_i$ of dimension d for $i = 1, \ldots, N$, so z_{ij} will be 1 if data point i belongs to category j and z_{ik} will be zero for all $k \neq j$. We assume a Dirichlet prior $\text{Dir}(\alpha)$ on ω, with density $p(\omega) \propto \prod_{j=1}^d \omega_d^{\alpha_j}$, and that the data is drawn from $\mathbf{z}_i \,|\, \omega \sim \text{Categorical}(\omega)$ leading to a $\text{Dir}(\alpha + \sum_{i=1}^N \mathbf{z}_i)$ posterior. An important transformation we will use repeatedly throughout this article, is that if we have d random gamma variables $X_j \sim \text{Gamma}(\alpha_j, 1)$. Then $(X_1, \ldots, X_d)/\sum_j X_j$ will have $\text{Dir}(\alpha)$ distribution, where $\alpha = (\alpha_1, \ldots, \alpha_d)$.

In this simple case the posterior of ω can be exactly calculated. However, in the applications we consider the $\mathbf{z}_i$ are latent variables, and they are also simulated as part of a larger Gibbs sampler. Thus the $\mathbf{z}_i$ will change at each iteration of our algorithm. We are interested in the situation where this is the case, and N is large, so that standard MCMC runs prohibitively slowly. The idea of SGMCMC in this situation is to use sub-samples of the $\mathbf{z}_i$s to propose appropriate local-moves to ω.

Applying SGMCMC to models which contain simplex parameters is challenging due to their constraints. Naively applying SGMCMC can lead to invalid values being proposed. The first work to introduce an SGMCMC algorithm specifically for the probability simplex was Patterson and Teh (2013), the algorithm is a variant of SGLD known as stochastic gradient Riemannian Langevin dynamics (SGRLD). Patterson and Teh (2013) try a variety of transformations for ω which will move the problem onto a space in $\mathbb{R}^d$, where standard SGMCMC can be applied. They also build upon standard SGLD by developing a Riemannian variant (see Girolami and Calderhead, 2011). Riemannian MCMC takes into account the geometry of the space, which assisted with errors at the boundary of the space. The parameterisation Patterson and Teh (2013) find numerically performs the best is $\omega_j = |\theta_j|/\sum_{j=1}^d |\theta_j|$. They use a mirrored gamma prior for θ_j, which has density $p(\theta_j) \propto |\theta_j|^{\alpha_j-1}e^{-|\theta_j|}$. This means the prior for ω remains the required Dirichlet distribution. They calculate the density of $\mathbf{z}_i$ given θ using a change of variables and use an SGLD update to calculate θ.

2.3 SGRLD on Sparse Simplex Spaces

Patterson and Teh (2013) suggested that the boundary of the space is where most problems occur using these kind of samplers. In many popular applications, such as LDA and modeling sparse networks, some of the components ω_j will be close to 0, referred to as a *sparse* space. In other words, there will be many j for which $\sum_{i=1}^N z_{ij} = 0$. In fact, this is their main motivation for introducing the
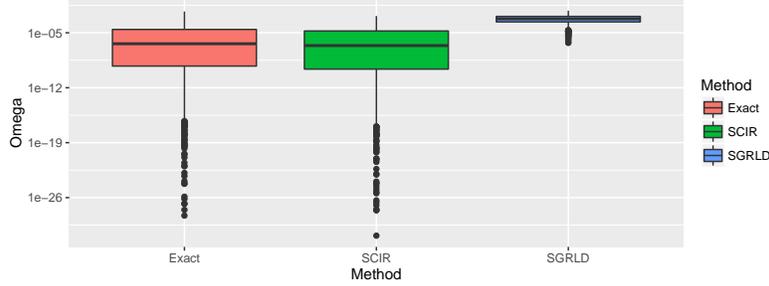
Figure 1: Boxplots of a 1000 iteration sample from SGRLD and SCIR fit to a sparse Dirichlet posterior, compared to 1000 exact independent samples. On the log scale.

Riemannian ideas to their SGLD algorithm. In order to demonstrate the problems with using SGRLD in this case, we provide a similar experiment to Patterson and Teh (2013). We use SGRLD to simulate from a sparse simplex parameter ω of dimension ten with $N = 1000$. We set $\sum_{i=1}^{N} z_{i1} = 800$, $\sum_{i=1}^{N} z_{i2} = \sum_{i=1}^{N} z_{i3} = 100$, and $\sum_{i=1}^{N} z_{ij} = 0$, for $3 < j \leq 10$. The prior parameter α was set to 0.1 for all components. Leading to a highly sparse Dirichlet posterior. We will refer back to this experiment as the *running experiment*. In Figure 1 we provide boxplots from a sample of the fifth component of ω using SGRLD after 1000 iterations with 1000 iterations of burn-in, compared with boxplots from an exact sample. The method SCIR will be introduced later. We can see from Figure 1 that SGRLD rarely proposes small values of ω. This becomes a significant issue for sparse Dirichlet distributions, since the lack of small values leads to a poor approximation to the posterior; as we can see from the boxplots.

We hypothesize that the reason SGRLD struggles when ω_j is near the boundary is due to the discretization by h, and we now try to diagnose this issue in detail. The problem relates to the bias of SGLD, caused by the discretization of the algorithm. We use the results of Vollmer et al. (2016) to characterize this bias for a fixed stepsize h. For similar results when the stepsize scheme is decreasing, we refer the reader to Teh et al. (2016). Proposition 2.1 is a simple application of Vollmer et al. (2016, Theorem 3.3), so we refer the reader to that article for full details of the assumptions. For simplicity of the statement, we assume that θ is 1-dimensional, but the results are easily adapted to the d-dimensional case.

Proposition 2.1. *(Vollmer et al., 2016) Under Vollmer et al. (2016, Assumptions 3.1 and 3.2), assume θ is 1-dimensional. Let θ_m be iteration m of an SGLD algorithm for $m = 1, \ldots, M$, then the asymptotic bias defined by $\lim_{M \to \infty} \left| 1/M \sum_{m=1}^{M} \mathbb{E}[\theta_m] - \mathbb{E}_{\pi}[\theta] \right|$ has leading term $O(h)$.*

While ordinarily this asymptotic bias will be hard to disentangle from other sources of error, as $\mathbb{E}_{\pi}[\theta]$ gets close to zero, h will have to be set prohibitively small to give a good approximation to θ. The crux of the issue is that, while the *absolute error* remains the same, at the boundary of the space the *relative error* is large since θ is small, and biased upwards due to the positivity constraint. To counteract this, in the next section we introduce a method which has no discretization error. This allows us to prove that the asymptotic bias, as defined in Proposition 2.1, will be zero for any choice of stepsize h.

3 The Stochastic Cox-Ingersoll-Ross Algorithm

We now wish to counteract the problems with SGRLD on sparse simplex spaces. First, we make the following observation: rather than applying a reparameterization of the prior for ω; we can model the posterior for θ_j directly and independently as $\theta_j \mid \mathbf{z} \sim \text{Gamma}(\alpha_j + \sum_{i=1}^{N} z_{ij}, 1)$. Then using the gamma reparameterization $\omega = \theta / \sum_j \theta_j$ still leads to the desired Dirichlet posterior. This leaves the θ_j in a much simpler form, and this simpler form enables us to remove all discretization error. We do this by using an alternative underlying process to the Langevin diffusion. The diffusion we use is known as the Cox-Ingersoll-Ross (CIR) process, commonly used in mathematical finance. A CIR

process θ_t with parameter a and stationary distribution $\text{Gamma}(a, 1)$ has the following form

$$d\theta_t = (a - \theta_t)dt + \sqrt{2\theta_t}dW_t. \tag{3.1}$$

The standard CIR process has more parameters, but we found changing these made no difference to the properties of our proposed scalable sampler and so we omit them (for exact details see the Supplementary Material).

The CIR process has many nice properties. One that is particularly useful for us is that the *transition density* is known exactly. Define $\chi^2(\nu, \mu)$ to be the non-central chi-squared distribution with ν degrees of freedom and non-centrality parameter μ. If at time t we are at state ϑ_t, then the probability distribution of θ_{t+h} is given by

$$\theta_{t+h} \,|\, \theta_t = \vartheta_t \sim \frac{1 - e^{-h}}{2}W, \qquad W \sim \chi^2 \left(2a, 2\vartheta_t \frac{e^{-h}}{1 - e^{-h}} \right). \tag{3.2}$$

This transition density allows us to simulate directly from the CIR process with no discretization error. Furthermore, it has been proved that the CIR process is negative with probability zero (Cox et al., 1985), meaning we will not need to take absolute values as is required for the SGRLD algorithm.

3.1 Adapting for Large Datasets

The next issue we need to address is how to sample from this process when the dataset is large. Suppose that z_i is data for $i = 1, \dots, N$, for some large N, and that our target distribution is $\text{Gamma}(a, 1)$, where $a = \alpha + \sum_{i=1}^{N} z_i$. We want to approximate the target by simulating from the CIR process using only a subset of $\mathbf{z}$ at each iteration. A natural thing to do would be at each iteration to replace a in the transition density equation (3.2) with an unbiased estimate $\hat{a} = \alpha + N/n \sum_{i \in S} z_i$, where $S \subset \{1, \dots, N\}$, similar to SGLD. We will refer to a CIR process using unbiased estimates in this way as the stochastic CIR process (SCIR). Fix some stepsize h, which now determines how often $\hat{a}$ is resampled rather than the granularity of the discretization. Suppose $\hat{\theta}_m$ follows the SCIR process, then it will have the following update

$$\hat{\theta}_{m+1} \,|\, \hat{\theta}_m = \vartheta_m \sim \frac{1 - e^{-h}}{2}W, \qquad W \sim \chi^2 \left(2\hat{a}_m, 2\vartheta_m \frac{e^{-h}}{1 - e^{-h}} \right), \tag{3.3}$$

where $\hat{a}_m = \alpha + N/n \sum_{i \in S_m} z_i$.

We can show that this algorithm will approximately target the true posterior distribution in the same sense as SGLD. To do this, we draw a connection between the SCIR process and an SGLD algorithm, which allows us to use the arguments of SGLD to show that the SCIR process will target the desired distribution. More formally, we have the following relationship:

Theorem 3.1. *Let θ_t be a CIR process with transition 3.2. Then $U_t := g(\theta_t) = 2\sqrt{\theta_t}$ follows the Langevin diffusion for a generalized gamma distribution.*

Theorem 3.1, allows us to show that applying the transformation $g(\cdot)$ to the approximate SCIR process, leads to a discretization free SGLD algorithm for a generalized gamma distribution. Similarly, applying $g^{-1}(\cdot)$ to the approximate target of this SGLD algorithm leads to the desired $\text{Gamma}(a, 1)$ distribution. Full details are given after the proof of Theorem 3.1. The result means that similar to SGLD, we can replace the CIR parameter a with an unbiased estimate $\hat{a}$ created from a minibatch of data. Provided we re-estimate a from one iteration to the next using different minibatches, the approximate target distribution will still be $\text{Gamma}(a, 1)$. As in SGLD, there will be added error based on the noise in the estimate $\hat{a}$. However, from the desirable properties of the CIR process we are able to quantify this error more easily than for the SGLD algorithm, and we do this in Section 4.
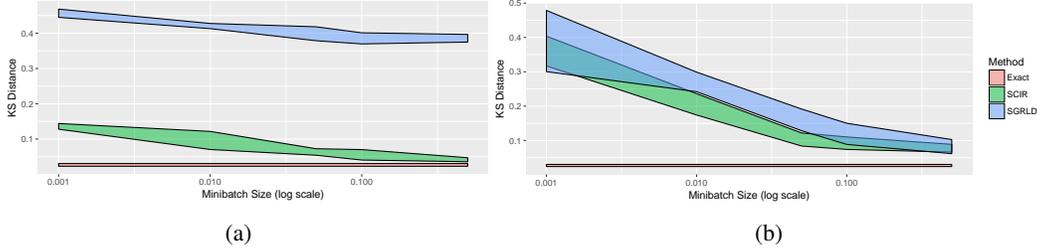
Figure 2: Kolmogorov-Smirnov distance for SGRLD and SCIR at different minibatch sizes when used to sample from (a), a sparse Dirichlet posterior and (b) a dense Dirichlet posterior.

Algorithm 1 below summarizes how SCIR can be used to sample from the simplex parameter $\omega \,|\, \mathbf{z} \sim \mathrm{Dir}(\alpha + \sum_{i=1}^{N} \mathbf{z}_i)$. This can be done in a similar way to SGRLD, with the same per-iteration computational cost, so the improvements we demonstrate later are essentially for free.

Algorithm 1: Stochastic Cox-Ingersoll-Ross (SCIR) for sampling from the probability simplex.

Input: Starting points θ_0, stepsize h, minibatch size n.
Result: Approximate sample from $\omega \,|\, \mathbf{z}$.
for $m = 1$ **to** M **do**
    Sample minibatch S_m from $\{1, \ldots, N\}$
    for $j = 1$ **to** d **do**
        Set $\hat{a}_j \leftarrow \alpha + N/n \sum_{i \in S_m} z_{ij}$.
        Sample $\hat{\theta}_{mj} \,|\, \hat{\theta}_{(m-1)j}$ using (3.3) with parameter $\hat{a}_j$ and stepsize h.
    end
    Set $\omega_m \leftarrow \theta_m / \sum_j \theta_{mj}$.
end

3.2 SCIR on Sparse Data

We test the SCIR process on two synthetic experiments. The first experiment is the running experiment on the sparse Dirichlet posterior of Section 2.3. The second experiment allocates 1000 datapoints equally to each component, leading to a highly dense Dirichlet posterior. For both experiments, we run 1000 iterations of optimally tuned SGRLD and SCIR algorithms and compare to an exact sample. For the sparse experiment, Figure 1 shows boxplots of samples from the fifth component of ω, which is sparse. For both experiments, Figure 2 plots the Kolmogorov-Smirnov distance (d_{KS}) between the approximate samples and the true posterior (full details of the distance measure are given in the Supplementary Material). For the boxplots, a minibatch of size 10 is used; for the d_{KS} plots the proportion of data in the minibatch is varied from 0.001 to 0.5. The d_{KS} plots show the runs of five different seeds, which gives some idea of variability.

The boxplots of Figure 1 demonstrate that the SCIR process is able to handle smaller values of ω much more readily than SGRLD. The impact of this is demonstrated in Figure 2a, the sparse d_{KS} plot. Here the SCIR process is achieving much better results than SGRLD, and converging towards the exact sample at larger minibatch sizes. The dense d_{KS} plot of Figure 2b shows that as we move to the dense setting the samplers have similar properties. The conclusion is that the SCIR algorithm is a good choice of simplex sampler for either the dense or sparse case.

4 Theoretical Analysis

In the following theoretical analysis we wish to target a Gamma$(a, 1)$ distribution, where $a = \alpha + \sum_{i=1}^{N} z_i$ for some data $\mathbf{z}$. We run an SCIR algorithm with stepsize h for M iterations, yielding the sample $\hat{\theta}_m$ for $m = 1, \ldots, M$. We compare this to an exact CIR process with stationary distribution Gamma$(a, 1)$, defined by the transition equation in (3.2). We do this by deriving the moment generating function (MGF) of $\hat{\theta}_m$ in terms of the MGF of the exact CIR process. This allows

us to quantify the moments of $\hat{\theta}_m$ in the analysis to follow. For simplicity, we fix a stepsize h and, abusing notation slightly, set θ_m to be a CIR process that has been run for time mh.

Theorem 4.1. *Let $\hat{\theta}_M$ be the SCIR process defined in (3.3) starting from θ_0 after M steps with stepsize h. Let θ_M be the corresponding exact CIR process, also starting from θ_0, run for time Mh, and with coupled noise. Then the MGF of $\hat{\theta}_M$ is given by*

$$M_{\hat{\theta}_M}(s) = M_{\theta_M}(s) \prod_{m=1}^{M} \left[\frac{1 - s(1 - e^{-mh})}{1 - s(1 - e^{-(m-1)h})} \right]^{-(\hat{a}_m - a)}, \tag{4.1}$$

where we have

$$M_{\theta_M}(s) = \left[1 - s(1 - e^{-Mh}) \right]^{-a} \exp\left[\theta_0 \frac{s e^{-Mh}}{1 - s(1 - e^{-Mh})} \right].$$

The proof of this result follows by induction from the properties of the non-central chi-squared distribution. The result shows that the MGF of the SCIR can be written as the MGF of the exact underlying CIR process, as well as an error term in the form of a product. Deriving the MGF enables us to find the non-asymptotic bias and variance of the SCIR process, which is more interpretable than the MGF itself. The results are stated formally in the following Corollary.

Corollary 4.2. *Given the setup of Theorem 4.1,*

$$\mathbb{E}[\hat{\theta}_M] = \mathbb{E}[\theta_M] = \theta_0 e^{-Mh} + a(1 - e^{Mh}),$$

so that, since $\mathbb{E}_\pi[\theta] = a$, then $\lim_{M \to \infty} |\frac{1}{M} \sum_{m=1}^{M} \mathbb{E}[\hat{\theta}_m] - \mathbb{E}_\pi[\theta]| = 0$ and SCIR is asymptotically unbiased. Similarly,

$$\mathbb{V}\mathrm{ar}[\hat{\theta}_M] = \mathbb{V}\mathrm{ar}[\theta_M] + (1 - e^{-2Mh}) \frac{1 - e^{-h}}{1 + e^{-h}} \mathbb{V}\mathrm{ar}[\hat{a}],$$

where $\mathbb{V}\mathrm{ar}[\hat{a}] = \mathbb{V}\mathrm{ar}[\hat{a}_m]$ for $m = 1, \ldots, M$ and

$$\mathbb{V}\mathrm{ar}[\theta_M] = 2\theta_0(e^{-Mh} - e^{-2Mh}) + a(1 - e^{-Mh})^2.$$

The results show that the approximate process is asymptotically unbiased. We believe this explains the improvements the method has over SGRLD in the experiments of Sections 3.2 and 5. We also obtain the non-asymptotic variance as a simple sum of the variance of the exact underlying CIR process, and a quantity involving the variance of the estimate $\hat{a}$. This is of a similar form to the strong error of SGLD (Sato and Nakagawa, 2014), though without the contribution from the discretization. The variance of the SCIR is somewhat inflated over the variance of the CIR process. Reducing this variance would improve the properties of the SCIR process and would be an interesting avenue for further work. Control variate ideas could be applied for this purpose (Nagapetyan et al., 2017; Baker et al., 2017; Chatterji et al., 2018) and they may prove especially effective since the mode of a gamma distribution is known exactly.

5 Experiments

5.1 Latent Dirichlet Allocation

Latent Dirichlet allocation (LDA, see Blei et al., 2003) is a popular model used to summarize a collection of documents by clustering them based on underlying topics. The data for the model is a matrix of word frequencies, with a row for each document. LDA is based on a generative procedure. For each document l, a discrete distribution over the K potential topics, θ_l, is drawn as $\theta_l \sim \mathrm{Dir}(\alpha)$ for some suitably chosen hyperparameter α. Each topic k is associated with a discrete distribution ϕ_k over all the words in a corpus, meant to represent the common words associated with particular topics. This is drawn as $\phi_k \sim \mathrm{Dir}(\beta)$, for some suitable β. Finally, each word in document l is drawn a topic k from θ_l and then the word itself is drawn from ϕ_k.

LDA is a good example for this method because ϕ_k is likely to be very sparse, there are many words which will not be associated with a given topic at all. The code is an adaption of the code released by Patterson and Teh (2013), which we apply to a dataset of scraped Wikipedia documents. At each
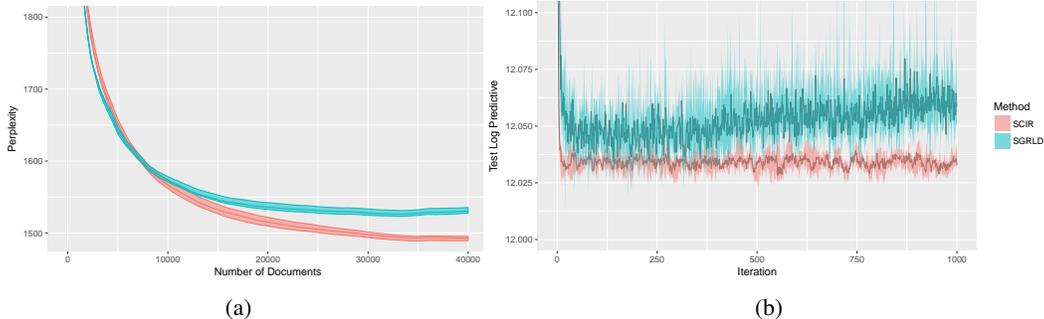
Figure 3: (a) plots the perplexity of SGRLD and SCIR when used to sample from the LDA model of Section 5.1 applied to Wikipedia documents; (b) plots the log predictive on a test set of the anonymous Microsoft user dataset, sampling the mixture model defined in Section 5.2 using SCIR and SGRLD.

iteration a minibatch of 50 documents is sampled in an online manner. We use the same vocabulary set as in Patterson and Teh (2013) which consists of approximately 8000 words. The exponential of the average log-predictive on a held out set of 1000 documents is calculated every 5 iterations to evaluate the model. This quantity is known as the perplexity, and use a document completion approach to calculate it (Wallach et al., 2009). The perplexity is plotted for five runs using different seeds, which gives an idea of variability. Similar to Patterson and Teh (2013), for both methods we use a decreasing stepsize scheme of the form $h_m = h[1 + m/\tau]^{-\kappa}$. The results are plotted in Figure 3a. While the initial convergence rate is similar, SCIR keeps descending past where SGRLD begins to converge. This experiment serves as a good example for the impact that removing the discretization error has for this problem. Further impact would probably be seen if a larger vocabulary size is used, leading to sparser topic vectors. In real-world applications of LDA, it is quite common to use vocabulary sizes above 8000.

5.2 Bayesian Nonparametric Mixture Model

We apply SCIR to sample from a Bayesian nonparametric mixture model of categorical data, based on Dunson and Xing (2009). To the best of our knowledge, the development of SGMCMC methods for Bayesian nonparametric models has not been considered before, so we deem this to be another contribution of the work. In particular, we develop a truncation free, scalable sampler based on SGMCMC for Dirichlet processes (DP, see Ferguson, 1973). For more thorough details of DPs and the stochastic sampler developed, the reader is referred to the Supplementary Material. The model can be expressed as follows

$$\mathbf{x}_i \,|\, \theta, z_i \sim \text{Multi}(n_i, \theta_{z_i}), \qquad \theta, z_i \sim \text{DP}(\text{Dir}(a), \alpha). \tag{5.1}$$

Here $\text{Multi}(m, \phi)$ is a multinomial distribution with m trials and associated discrete probability distribution ϕ; $\text{DP}(G_0, \alpha)$ is a DP with base distribution G_0 and concentration parameter α. The DP component parameters and allocations are denoted by θ and z_i respectively. We define the number of observations N by $N := \sum_i n_i$, and let L be the number of instances of $\mathbf{x}_i$, $i = 1, \dots, L$. This type of mixture model is commonly used to model the dependence structure of categorical data, such as for genetic or natural language data (Dunson and Xing, 2009). The use of DPs (Ferguson, 1973) means we can account for the fact that we do not know the true dependence structure. DPs allow us to learn the number of mixture components in a penalized way during the inference procedure itself.

We apply this model to the anonymous Microsoft user dataset (Breese et al., 1998). This dataset consists of approximately $N = 10^5$ instances of $L = 30000$ anonymized users. Each instance details part of the website the user visits, which is one of $d = 294$ categories (here d denotes the dimension of $\mathbf{x}_i$). We use the model to try and characterize the typical usage patterns of the website. Since there are a lot of categories and only an average of three observations for any one user, these data are expected to be sparse. To infer the model, we use an novel algorithm, which is a minibatched version of the slice sampler (Walker, 2007; Papaspiliopoulos, 2008; Kalli et al., 2011). We assign an uninformative gamma prior on α, and this is inferred similarly to Escobar and West (1995). We minibatch the users at each iteration using $n = 1000$. For multimodal mixture models such as this, SGMCMC methods are known to get stuck in local modes (Baker et al., 2017), so we use a fixed stepsize for both SGRLD

and SCIR. Once again, we plot runs over 5 seeds to give an idea of variability. The results are plotted in Figure 3b. They show that SCIR consistently converges to a lower log predictive test score, and appears to have lower variance than SGRLD. SGRLD also appears to be producing worse scores as the number of iterations increases. We found that SGRLD had a tendency to propose many more clusters than were required. This is probably due to the asymptotic bias of Proposition 2.1, since this would lead to an inferred model that has a higher α parameter than is set, meaning more clusters would be proposed than are needed. In fact, setting a higher α parameter appeared to alleviate this problem, but led to a worse fit, which is more evidence that this is the case.

6 Discussion

We presented an SGMCMC method, the SCIR algorithm, for simplex spaces. We show that the method has no discretization error and is asymptotically unbiased. Our experiments demonstrate that these properties give the sampler improved performance over other SGMCMC methods for sampling from sparse simplex spaces. Many important large-scale models are sparse, so this is an important contribution. A number of useful theoretical properties for the sampler were derived, including the non-asymptotic variance and moment generating function. Finally, we demonstrate the impact of the sampler on a variety of interesting problems including a novel scalable Dirichlet process sampler. An interesting line of further work would be reducing the non-asymptotic variance, which could be done by means of control variates.

7 Acknowledgments

References

Baker, J., Fearnhead, P., Fox, E. B., and Nemeth, C. (2017). Control variates for stochastic gradient MCMC. Available from `https://arxiv.org/abs/1706.05439`.

Blackwell, D. and MacQueen, J. B. (1973). Ferguson distributions via polya urn schemes. *The Annals of Statistics*, 1(2):353–355.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Breese, J. S., Heckerman, D., and Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 43–52.

Chatterji, N. S., Flammarion, N., Ma, Y.-A., Bartlett, P. L., and Jordan, M. I. (2018). On the theory of variance reduction for stochastic gradient Monte Carlo. Available at `https://arxiv.org/abs/1802.05431v1`.

Chen, T., Fox, E., and Guestrin, C. (2014). Stochastic gradient Hamiltonian Monte Carlo. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1683–1691. PMLR.

Cox, J. C., Ingersoll, J. E., and Ross, S. A. (1985). A theory of the term structure of interest rates. *Econometrica*, 53(2):385–407.

Ding, N., Fang, Y., Babbush, R., Chen, C., Skeel, R. D., and Neven, H. (2014). Bayesian sampling using stochastic gradient thermostats. In *Advances in Neural Information Processing Systems 27*, pages 3203–3211.

Dubey, K. A., Reddi, S. J., Williamson, S. A., Poczos, B., Smola, A. J., and Xing, E. P. (2016). Variance reduction in stochastic gradient Langevin dynamics. In *Advances in Neural Information Processing Systems 29*, pages 1154–1162.

Dunson, D. B. and Xing, C. (2009). Nonparametric Bayes modeling of multivariate categorical data. *Journal of the American Statistical Association*, 104(487):1042–1051.

Escobar, M. D. and West, M. (1995). Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588.

Ferguson, T. S. (1973). A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1(2):209–230.

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

Kalli, M., Griffin, J. E., and Walker, S. G. (2011). Slice sampling mixture models. *Statistics and Computing*, 21(1):93–105.

Liverani, S., Hastie, D., Azizi, L., Papathomas, M., and Richardson, S. (2015). PReMiuM: An R package for profile regression mixture models using Dirichlet processes. *Journal of Statistical Software*, 64(7):1–30.

Ma, Y.-A., Chen, T., and Fox, E. (2015). A complete recipe for stochastic gradient MCMC. In *Advances in Neural Information Processing Systems*, pages 2917–2925.

Nagapetyan, T., Duncan, A., Hasenclever, L., Vollmer, S. J., Szpruch, L., and Zygalakis, K. (2017). The true cost of stochastic gradient Langevin dynamics. Available at `https://arxiv.org/abs/1706.02692`.

Papaspiliopoulos, O. (2008). A note on posterior sampling from Dirichlet mixture models. Technical Report. Available at `http://wrap.warwick.ac.uk/35493/1/WRAP_papaspliiopoulos_08-20wv2.pdf`.

Patterson, S. and Teh, Y. W. (2013). Stochastic gradient Riemannian Langevin dynamics on the probability simplex. In *Advances in Neural Information Processing Systems 26*, pages 3102–3110.

Rosenblatt, M. (1952). Remarks on a multivariate transformation. *The Annals of Mathematical Statistics*, 23(3):470–472.

Sato, I. and Nakagawa, H. (2014). Approximation analysis of stochastic gradient Langevin dynamics by using Fokker-Planck equation and Ito process. In *Proceedings of the 31st International Conference on Machine Learning*, pages 982–990. PMLR.

Sethuraman, J. (1994). A constructive definition of Dirichlet priors. *Statistica Sinica*, 4(2):639–650.

Teh, Y. W., Thiéry, A. H., and Vollmer, S. J. (2016). Consistency and fluctuations for stochastic gradient Langevin dynamics. *Journal of Machine Learning Research*, 17(7):1–33.

Vollmer, S. J., Zygalakis, K. C., and Teh, Y. W. (2016). Exploration of the (non-)asymptotic bias and variance of stochastic gradient Langevin dynamics. *Journal of Machine Learning Research*, 17(159):1–48.

Walker, S. G. (2007). Sampling the Dirichlet mixture model with slices. *Communications in Statistics*, 36(1):45–54.

Wallach, H. M., Murray, I., Salakhutdinov, R., and Mimno, D. (2009). Evaluation methods for topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1105–1112. PMLR.

Welling, M. and Teh, Y. W. (2011). Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning*, pages 681–688. PMLR.

Wenzhe Li, Sungjin Ahn, M. W. (2016). Scalable MCMC for mixed membership stochastic blockmodels. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pages 723–731. PMLR.

Zygalakis, K. C. (2011). On the existence and the applications of modified equations for stochastic differential equations. *SIAM Journal on Scientific Computing*, 33(1):102–130.

A Proofs

A.1 Proof of Proposition 2.1

Proof. Define the *local weak error* of SGLD, starting from θ_0 and with stepsize h, with test function ϕ by

$$\mathbb{E}\left|\phi(\theta_1) - \phi(\bar{\theta}_h)\right|,$$

where $\bar{\theta}_h$ is the true underlying Langevin diffusion (2.1), run for time h with starting point θ_0. Then it is shown by Vollmer et al. (2016) that if $\phi : \mathbb{R}^d \to \mathbb{R}$ is a smooth test function, and that SGLD applied with test function ϕ has local weak error $O(h)$, then

$$\mathbb{E}\left|\lim_{M\to\infty} 1/M \sum_{m=1}^{M} \phi(\theta_m) - \mathbb{E}_\pi[\phi(\theta)]\right|$$

is also $O(h)$. What remains to be checked is that using such a simple function for ϕ (the identity), does not cause things to disappear such that the local weak error of SGLD is no longer $O(h)$. The identity function is infinitely differentiable, thus is sufficiently smooth. For SGLD, we find that

$$\mathbb{E}[\theta_1|\theta_0] = \theta_0 + hf'(\theta_0).$$

For the Langevin diffusion, we define the one step expectation using the weak Taylor expansion of Zygalakis (2011), which is valid since we have made Assumptions 3.1 and 3.2 of Vollmer et al. (2016). Define the infinitesimal operator $\mathcal{L}$ of the Langevin diffusion (2.1) by

$$\mathcal{L}\phi = f'(\theta) \cdot \partial_\theta \phi(\theta) + \partial_\theta^2 \phi(\theta).$$

Then Zygalakis (2011) shows that the weak Taylor expansion of Langevin diffusion (2.1) has the form

$$\mathbb{E}[\bar{\theta}_h|\theta_0] = \theta_0 + h\mathcal{L}\phi(\theta_0) + \frac{h^2}{2}\mathcal{L}^2\phi(\theta_0) + O(h^3).$$

This means when ϕ is the identity then

$$\mathbb{E}[\bar{\theta}_h|\theta_0] = \theta_0 + hf'(\theta_0) + \frac{h^2}{2}\left[f(\theta)f'(\theta) + f''(\theta)\right] + O(h^3).$$

Since the terms agree up to $O(h)$ then it follows that even when ϕ is the identity, SGLD still has local weak error of $O(h)$. This completes the proof. $\qquad\square$

A.2 Proof of Theorem 3.1

Proof. Suppose we have a random variable U_∞ following a generalized gamma posterior with data $\mathbf{z}$ and the following density

$$f(u) \propto u^{2(\alpha + \sum_{i=1}^{N} z_i) - 1} e^{-u^2/4}.$$

Set $a := 2(\alpha + \sum_{i=1}^{N} z_i)$, Then $\partial \log f(u) = (2a - 1)/u - u/2$, so that the Langevin diffusion for U_∞ will have the following integral form

$$U_{t+h} \mid U_t = U_t + \int_t^{t+h} \left[\frac{2a - 1}{U_s} - \frac{U_s}{2}\right] ds + \sqrt{2} \int_t^{t+h} dW_t.$$

Applying Ito's lemma to U_t to transform to $\theta_t = g^{-1}(U_t) = U_t^2/4$ (here $g(\cdot)$ has been stated in the proof), we find that

$$\theta_{t+h} \mid \theta_t = \theta_t + \int_t^{t+h} [a - \theta_s] ds + \int_t^{t+h} \sqrt{2\theta_t} dW_t.$$

This is exactly the integral form for the CIR process. This completes the proof. $\qquad\square$

Now we give more details of the connection between SGLD and SCIR. Let us define an SGLD algorithm that approximately targets U_∞, but without the Euler discretization by

$$U_{(m+1)h} \mid U_{mh} = U_{mh} + \int_{mh}^{(m+1)h} \left[\frac{2\hat{a}_m - 1}{U_s} - \frac{U_s}{2}\right] ds + \sqrt{2} \int_{mh}^{(m+1)h} dW_t, \qquad \text{(A.1)}$$

where $\hat{a}_m$ is an unbiased estimate of a; for example, the standard SGLD estimate $\hat{a}_m = \alpha + N/n \sum_{i \in S_m} z_i$; also h is a tuning constant which determines how much time is simulated before resampling $\hat{a}_m$.

Again applying Ito's lemma to U_{mh} to transform to $\theta_{mh} = g(U_{mh}) = U_{mh}^2/4$, we find that

$$\theta_{(m+1)h} = \theta_{mh} + \int_{mh}^{(m+1)h} [\hat{a}_m - \theta_s]\, ds + \int_{mh}^{(m+1)h} \sqrt{2\theta_t}\, dW_t.$$

This is exactly the integral form for the update equation of an SCIR process.

Finally, to show SCIR has the desired approximate target, we use some properties of the gamma distribution. Firstly if $\theta_\infty \sim \text{Gamma}(a, 1)$ then $4\theta_\infty \sim \text{Gamma}(a, \frac{1}{4})$, so that $U_\infty = 2\sqrt{\theta_\infty}$ will have a generalized gamma distribution with density proportional to $h(u) \propto u^{2a-1} e^{-u^2/4}$. This is exactly the approximate target of the discretization free SGLD algorithm (A.1) we derived earlier.

A.3   Proof of Theorem 4.1

First let us define the following quantities

$$r(s) = \frac{se^{-h}}{1 - s(1 - e^{-h})}, \qquad r^{(n)}(s) = \underbrace{r \circ \cdots \circ r}_{n}(s).$$

Then we will make use of the following Lemmas:

Lemma A.1. *For all $n \in \mathbb{N}$ and $s \in \mathbb{R}$*

$$r^{(n)}(s) = \frac{se^{-nh}}{1 - s(1 - e^{-nh})}.$$

Lemma A.2. *For all $n \in \mathbb{N}$, $s \in \mathbb{R}$, set $r^{(0)}(s) := s$, then*

$$\prod_{i=0}^{n-1} \left[1 - r^{(i)}(s)(1 - e^{-h})\right] = \left[1 - s(1 - e^{-nh})\right].$$

Both can be proved by induction, which is shown in Section B.

Suppose that $\theta_1 | \theta_0$ is a CIR process, starting at θ_0 and run for time h. Then we can immediately write down the MGF of θ_1, $M_{\theta_1}(s)$, using the MGF of a non-central chi-squared distribution

$$M_{\theta_1}(s) = \mathbb{E}\left[e^{s\theta_1} | \theta_0\right] = \left[1 - s(1 - e^{-h})\right]^{-a} \exp\left[\frac{s\theta_0 e^{-h}}{1 - s(1 - e^{-h})}\right].$$

We can use this to find $\mathbb{E}\left[e^{s\theta_M} | \theta_{M-1}\right]$, and then take expectations of this with respect to θ_{M-2}, i.e. $\mathbb{E}\left[\mathbb{E}\left[e^{s\theta_M} | \theta_{M-1}\right] | \theta_{M-2}\right]$. This is possible because $\mathbb{E}\left[e^{s\theta_M} | \theta_{M-1}\right]$ has the form $C(s) \exp[\theta_{M-1} r(s)]$, where $C(s)$ is a function only involving s, and $r(s)$ is as defined earlier. Thus repeatedly applying this and using Lemmas A.1 and A.2 we find

$$M_{\theta_M}(s) = \left[1 - s(1 - e^{-Mh})\right]^{-a} \exp\left[\frac{s\theta_0 e^{-Mh}}{1 - s(1 - e^{-Mh})}\right]. \tag{A.2}$$

Although this was already known, we can use the same idea to find the MGF of the SCIR process.

The MGF of SCIR immediately follows using the same logic as before, as well as using the form of $M_{\theta_M}(s)$ and Lemmas A.1 and A.2. Leading to

$$M_{\hat{\theta}_M}(s) = \prod_{m=1}^{M} \left[1 - r^{(m-1)}(s)(1 - e^{-h})\right]^{-\hat{a}_m} \exp\left[\theta_0 r^{(M)}(s)\right]$$

$$= M_{\theta_M}(s) \prod_{m=1}^{M} \left[\frac{1 - s(1 - e^{-mh})}{1 - s(1 - e^{-(m-1)h})}\right]^{-(\hat{a}_m - a)}$$

12

A.4 Proof of Theorem 4.2

Proof. From Theorem 4.1, we have

$$M_{\hat{\theta}_M}(s) = M_{\theta_M}(s) \underbrace{\prod_{m=1}^{M} \left[1 - s(1 - e^{-mh})\right]^{-(\hat{a}_m - a)} \prod_{m=1}^{M} \left[1 - s(1 - e^{-(m-1)h})\right]^{-(a - \hat{a}_m)}}_{e_0(s)} \underbrace{\phantom{\prod_{m=1}^{M} \left[1 - s(1 - e^{-(m-1)h})\right]^{-(a - \hat{a}_m)}}}_{e_1(s)}.$$

We clearly have $M_{\theta_M}(0) = e_0(0) = e_1(0) = 1$. Differentiating we find

$$e_0'(s) = \sum_{i=1}^{M} (\hat{a}_i - a)(1 - e^{-ih}) \left[1 - s(1 - e^{-ih})\right]^{-1} e_0(s),$$

similarly

$$e_1'(s) = \sum_{i=1}^{M} (a - \hat{a}_i)(1 - e^{-(i-1)h}) \left[1 - s(1 - e^{-(i-1)h})\right]^{-1} e_1(s).$$

It follows that, labeling the minibatch noise up to iteration M by $\mathcal{B}_M$, and using the fact that $\mathbb{E}\hat{a}_i = a$ for all $i = 1, \ldots, M$ we have

$$
\begin{aligned}
\mathbb{E}\hat{\theta}_M &= \mathbb{E}\left[\mathbb{E}\left(\hat{\theta}_M | \mathcal{B}_M\right)\right] \\
&= \mathbb{E}\left[M_{\hat{\theta}_M}'(0)\right] \\
&= \mathbb{E}\left[M_{\theta_M}'(0)e_0(0)e_1(0) + M_{\theta_M}(0)e_0'(0)e_1(0) + M_{\theta_M}(0)e_0(0)e_1'(0)\right] \\
&= \mathbb{E}\theta_M.
\end{aligned}
$$

Now taking second derivatives we find

$$e_0''(s) = \sum_{i=1}^{M} (\hat{a}_i - a)(\hat{a}_i - a - 1)(1 - e^{-ih})^2 \left[1 - s(1 - e^{-ih})\right]^{-2} e_0(s)$$

$$+ \sum_{i \neq j} (\hat{a}_i - a)(\hat{a}_j - a)(1 - e^{-ih})(1 - e^{-jh}) \left[1 - s(1 - e^{-ih})\right]^{-1} \left[1 - s(1 - e^{-jh})\right]^{-1} e_0(s).$$

Now taking expectations with respect to the minibatch noise, noting independence of $\hat{a}_i$ and $\hat{a}_j$ for $i \neq j$,

$$\mathbb{E}\left[e_0''(0)\right] = \sum_{i=1}^{M} (1 - e^{-ih})^2 \mathbb{V}\text{ar}(\hat{a}_i).$$

By symmetry

$$\mathbb{E}\left[e_1''(0)\right] = \sum_{i=1}^{M} (1 - e^{-(i-1)h})^2 \mathbb{V}\text{ar}(\hat{a}_i).$$

We also have

$$\mathbb{E}\left[e_0'(0)e_1'(0)\right] = -\sum_{i=1}^{M} (1 - e^{-ih})(1 - e^{-(i-1)h}) \mathbb{V}\text{ar}(\hat{a}_i).$$

13

Now we can calculate the second moment using the MGF as follows, note that $\mathbb{E}(e_0'(0)) = \mathbb{E}(e_1'(0)) = 0$,

$$
\begin{aligned}
\mathbb{E}\hat{\theta}_M^2 &= \mathbb{E}\left[M_{\hat{\theta}_M}''(0)\right] \\
&= \mathbb{E}\left[M_{\theta_M}''(0)e_0(0)e_1(0) + M_{\theta_M}(0)e_0''(0)e_1(0) + M_{\theta_M}(0)e_0(0)e_1''(0) + 2M_{\theta_M}(0)e_0'(0)e_1'(0)\right] \\
&= \mathbb{E}\theta_M^2 + \sum_{i=1}^{M}(1 - e^{-ih})^2\mathbb{V}\mathrm{ar}(\hat{a}_i) + \sum_{i=1}^{M}(1 - e^{-(i-1)h})^2\mathbb{V}\mathrm{ar}(\hat{a}_i) - 2\sum_{i=1}^{M}(1 - e^{-ih})(1 - e^{-(i-1)h})\mathbb{V}\mathrm{ar}(\hat{a}_i) \\
&= \mathbb{E}\theta_M^2 + \mathbb{V}\mathrm{ar}(\hat{a})\left[e^{-2Mh} - 1 + 2\sum_{i=1}^{M}\left(e^{-2(i-1)h} - e^{-(2i-1)h}\right)\right] \\
&= \mathbb{E}\theta_M^2 + \mathbb{V}\mathrm{ar}(\hat{a})\left[e^{-2Mh} - 1 + 2\sum_{i=0}^{2M-1}(-1)^i e^{-ih}\right] \\
&= \mathbb{E}\theta_M^2 + \mathbb{V}\mathrm{ar}(\hat{a})\left[e^{-2Mh} - 1 + \frac{2 - 2e^{-2Mh}}{1 + e^{-h}}\right] \\
&= \mathbb{E}\theta_M^2 + \mathbb{V}\mathrm{ar}(\hat{a})(1 - e^{-2Mh})\left[\frac{1 - e^{-h}}{1 + e^{-h}}\right]
\end{aligned}
$$

$\square$

B   Proofs of Lemmas

B.1   Proof of Lemma A.1

Proof. We proceed by induction. Clearly the result holds for $n = 1$. Now assume the result holds for all $n \le k$, we prove the result for $n = k + 1$ as follows

$$
\begin{aligned}
r^{(k+1)}(s) &= r \circ r^{(k)}(s) \\
&= r\left(\frac{se^{-kh}}{1 - s(1 - e^{-kh})}\right) \\
&= \frac{se^{-kh}}{1 - s(1 - e^{-kh})} \cdot \frac{e^{-h}(1 - s(1 - e^{-kh}))}{1 - s(1 - e^{-kh}) - se^{-kh}(1 - e^{-h})} \\
&= \frac{se^{-(k+1)h}}{1 - s(1 - e^{-(k+1)h})}.
\end{aligned}
$$

Thus the result holds for all $n \in \mathbb{N}$ by induction. $\square$

B.2   Proof of Lemma A.2

Proof. Once again we proceed by induction. Clearly the result holds for $n = 1$. Now assume the result holds for all $n \le k$. Using Lemma A.1, we prove the result for $n = k + 1$ as follows

$$
\begin{aligned}
\prod_{i=0}^{k}\left[1 - r^{(i)}(s)(1 - e^{-h})\right] &= \left[1 - s(1 - e^{-kh})\right]\left[1 - \frac{se^{-kh}(1 - e^{-h})}{1 - s(1 - e^{-kh})}\right] \\
&= \left[1 - s(1 - e^{-kh})\right]\left[\frac{1 - s(1 - e^{-(k+1)h})}{1 - s(1 - e^{-kh})}\right] \\
&= \left[1 - s(1 - e^{-(k+1)h})\right]
\end{aligned}
$$

Thus the result holds for all $n \in \mathbb{N}$ by induction. $\square$

C   CIR Parameter Choice

As mentioned in Section 3, the standard CIR process has more parameters than those presented. The full form for the CIR process is as follows

$$d\theta_t = b(a - \theta_t)dt + \sigma\sqrt{\theta_t}dW_t, \tag{C.1}$$

where a, b and σ are parameters to be chosen. This leads to a Gamma$(2ab/\sigma^2, 2b/\sigma^2)$ stationary distribution. For our purposes, the second parameter of the gamma stationary distribution can be set arbitrarily, thus it is natural to set $2b = \sigma^2$ which leads to a Gamma$(a, 1)$ stationary distribution and a process of the following form

$$d\theta_t = b(a - \theta_t)dt + \sqrt{2b\theta_t}dW_t.$$

Fix the stepsize h, and use the slight abuse of notation that $\theta_m = \theta_{mh}$. The process has the following transition density

$$\theta_{m+1} \mid \theta_m = \vartheta_m \sim \frac{1 - e^{-bh}}{2}W, \qquad W \sim \chi^2\left(2a, 2\vartheta_m\frac{e^{-bh}}{1 - e^{-bh}}\right).$$

Using the MGF of a non-central chi-square distribution we find

$$M_{\theta_M}(s) = \left[1 - s(1 - e^{-Mbh})\right]^{-a}\exp\left[\frac{s\theta_0 e^{-Mbh}}{1 - s(1 - e^{-Mbh})}\right].$$

Clearly b and h are unidentifiable. Thus we arbitrarily set $b = 1$.

D   Stochastic Slice Sampler for Dirichlet Processes

D.1   Dirichlet Processes

The Dirichlet process (DP) (Ferguson, 1973) is parameterised by a scale parameter $\alpha \in \mathbb{R}_{>0}$ and a base distribution G_0 and is denoted $DP(G_0, \alpha)$. A formal definition is that G is distributed according to $DP(G_0, \alpha)$ if for all $k \in \mathbb{N}$ and k-partitions $\{B_1, \ldots, B_k\}$ of the space of interest Ω

$$(G(B_1), \ldots, G(B_k)) \sim \text{Dir}(\alpha G_0(B_1), \ldots, \alpha G_0(B_k)).$$

More intuitively, suppose we simulate $\theta_1, \ldots \theta_N$ from G. Then integrating out G (Blackwell and MacQueen, 1973) we can represent θ_N conditional on θ_{-N} as

$$\theta_N \mid \theta_1, \ldots, \theta_{N-1} \sim \frac{1}{N - 1 + \alpha}\sum_{i=1}^{N-1}\delta_{\theta_i} + \frac{\alpha}{N - 1 + \alpha}G_0,$$

where δ_θ is the distribution concentrated at θ.

An explicit construction of a DP exists due to Sethuraman (1994), known as the *stick-breaking construction*. The slice sampler we develop in this section is based on this construction. For $j = 1, 2, \ldots$, set $V_j \sim \text{Beta}(1, \alpha)$ and $\theta_j \sim G_0$. Then the stick breaking construction is given by

$$\omega_j := V_j\prod_{k=1}^{j-1}(1 - V_k) \tag{D.1}$$

$$G \sim \sum_{j=1}^{\infty}\omega_j\delta_{\theta_j}, \tag{D.2}$$

and we have $G \sim DP(G_0, \alpha)$.

D.2   Slice sampling Dirichlet process mixtures

We focus on sampling from Dirichlet process mixture models defined by

$$X_i \mid \theta_i \sim F(\theta_i)$$
$$\theta_i \mid G \sim G$$
$$G \mid G_0, \alpha \sim DP(G_0, \alpha).$$

A popular MCMC algorithm for sampling from this model is the *slice sampler*, originally developed by Walker (2007) and further developed by Papaspiliopoulos (2008); Kalli et al. (2011). The slice sampler is based directly on the stick-breaking construction (D.2), rather than the sequential (Pólya urn) formulation of (D.1). This makes it a more natural approach to develop a stochastic sampler from; since the stochastic sampler relies on conditional independence assumptions. The slice sampler can be extended to other Bayesian nonparametric models quite naturally, from their corresponding stick breaking construction.

We want to make inference on a Dirichlet process using the stick breaking construction directly. Suppose the mixture distribution F, and the base distribution G_0 admit densities f and g_0. Introducing the variable z, which determines which component x is currently allocated to, we can write the density as follows

$$p(x|\omega, \theta, z) \propto \omega_z f(x|\theta_z).$$

Theoretically we could now use a Gibbs sampler to sample conditionally from z, θ and ω. However this requires updating an infinite number of weights, similarly z is drawn from a categorical distribution with an infinite number of categories. To get around this Walker (2007) introduces another latent variable u, such that the density is now

$$p(x|\omega, \theta, z, u) \propto \mathbf{1}(u < \omega_z) f(x|\theta_z),$$

so that the full likelihood is given by

$$p(\mathbf{x}|\omega, \theta, \mathbf{z}, \mathbf{u}) \propto \prod_{i=1}^{N} \mathbf{1}(u_i < \omega_{z_i}) f(x_i|\theta_{z_i}). \tag{D.3}$$

Walker (2007) shows that in order for a standard Gibbs sampler to be valid given (D.3), the number of weights ω_j that needs to be sampled given this new latent variable is now finite, and given by k^*, where k^* is the smallest value such that $\sum_{j=1}^{k^*} \omega_j > 1 - u_i$.

The Gibbs algorithm can now be stated as follows, note we have included an improvement suggested by Papaspiliopoulos (2008), in how to sample v_j.

- Sample the slice variables $\mathbf{u}$, given by $u_i \mid \omega, \mathbf{z} \sim U(0, \omega_{z_i})$ for $i = 1, \ldots, N$. Calculate $u^* = \min \mathbf{u}$.

- Delete or add components until the number of current components k^* is the smallest value such that $u^* < 1 - \sum_{j=1}^{k^*} \omega_j$.

- Draw new component allocations z_i for $i = 1, \ldots, N$, using
  $p(z_i = j|x_i, u_i, \omega, \theta) \propto \mathbf{1}(\omega_j > u_i) f(x_i|\theta)$.

- For $j \leq k^*$, sample new component parameters θ_j from
  $p(\theta_j|\mathbf{x}, \mathbf{z}) \propto g_0(\theta_j) \prod_{i \,:\, z_i = j} f(x_i|\theta_j)$

- For $j \leq k^*$ calculate simulate new stick breaks v from
  $v_j \mid \mathbf{z}, \alpha \sim \text{Beta}\left(1 + m_j, \alpha + \sum_{l=j+1}^{k^*} m_l\right)$. Here $m_j := \sum_{i=1}^{N} \mathbf{1}_{z_i=j}$.

- Update ω using the new v: $\omega_j = v_j \prod_{l<j}(1 - v_j)$.

D.3 Stochastic Sampler

The conditional independence of each update of the slice sampler introduced in Section D.2 makes it possible to adapt it to a stochastic variant. Suppose we update θ and v given a minibatch of the $\mathbf{z}$ and $\mathbf{u}$ parameters. Then since the $\mathbf{z}$ and $\mathbf{u}$ parameters are just updated from the marginal of the posterior, only updating a minibatch of these parameters at a time would leave the posterior as the invariant distribution. Our exact MCMC procedure is similar to that in the R package PReMiuM (Liverani et al., 2015), though they do not use a stochastic sampler. First define the following: $Z^* = \max \mathbf{z}$; $S \subset \{1, \ldots, N\}$ is the current minibatch; $u^* = \min \mathbf{u}_S$; k^* is the smallest value such that $\sum_{j=1}^{k^*} \omega_j > 1 - u^*$. Then our updates proceed as follows:

- Recalculate Z^* and S (note this can be done in $O(n)$ time since only n $\mathbf{z}$ values changed).

- For $j = 1, \ldots, Z^*$ sample v_j stochastically with SCIR from
  $v_j \,|\, \mathbf{z}, \alpha \sim \text{Beta}(1 + \hat{m}_j, \alpha + \sum_{l=j+1}^{k^*} \hat{m}_l)$. Here $\hat{m}_j = N/n \sum_{i \in S} \mathbf{1}_{z_i=j}$.

- Update ω_j using the new v: $\omega_j = v_j \prod_{l<j}(1 - v_j)$.

- For $j = 1, \ldots, Z^*$ sample θ_j stochastically with SGMCMC from
  $p(\theta_j | \mathbf{x}, \mathbf{z}) \propto g_0(\theta_j) \prod_{S_j} f(x_i | \theta_j)$. Here $S_j = \{i : z_i = j \text{ and } i \in S\}$.

- For $i \in S$ sample the slice variables $u_i \,|\, \omega, \mathbf{z} \sim U(0, \omega_{z_i})$.

- Sample α if required. Using Escobar and West (1995), for our example we assume a Gamma(b_1, b_2) prior so that $\alpha \,|\, v_{1:Z^*} \sim \text{Gamma}(b_1 + Z^*, b_2 - \sum_{j=1}^{K^*} \log(1 - v_j))$.

- Recalculate u^*. Sample additional ω_j from the prior, until k^* is reached. For $j = (Z^* + 1), \ldots, k^*$ sample additional θ_j from the prior.

- For $i \in S$, sample z_i, where $\mathbb{P}(z_i = j | u_i, \omega, \theta, \mathbf{x}) \propto \mathbf{1}(\omega_j > u_i) f(x_i | \theta_j)$.

Note that for our particular example, we have the following conditional update for θ (ignoring minibatching for simplicity):

$$\theta_j \,|\, z_j, \mathbf{x} \sim \text{Dirichlet}\left(a + \sum_{i \in S_j} x_{i1}, \ldots, a + \sum_{i \in S_j} x_{id}\right).$$

E Experiments

E.1 Synthetic

We now fully explain the distance measure used in the synthetic experiments. Suppose we have random variables X taking values in $\mathbb{R}$ with cumulative density function (CDF) F. We also have an approximate sample from X, $\hat{X}$ with empirical density function $\hat{F}$. The Kolmogorov-Smirnov distance d_{KS} between X and $\hat{X}$ is defined by $d_{KS}(X, \hat{X}) = \sup_{x \in \mathbb{R}} \left\|\hat{F}(x) - F(x)\right\|$. However the Dirichlet distribution is multi-dimensional, so we measure the average Kolmogorov-Smirnov distance across dimensions by using the Rosenblatt transform (Rosenblatt, 1952).

Suppose now that X takes values in $\mathbb{R}^d$. Define the conditional CDF of $X_k = x_k | X_{k-1} = x_{k-1}, \ldots, X_1 = x_1$ to be $F(x_k | \mathbf{x}_{1:(k-1)})$. Suppose we have an approximate sample from X, which we denote $\mathbf{x}^{(m)}$, for $m = 1, \ldots M$. Define $\hat{F}_j$ to be the empirical CDF defined by the samples $F(x_j^{(m)} | \mathbf{x}_{1:(j-1)}^{(m)})$. Then Rosenblatt (1952) showed that if $\hat{X}$ is a true sample from X then $\hat{F}_j$ should be the uniform distribution and independent of $\hat{F}_k$ for $k \neq j$. This allows us to define a Kolmogorov-Smirnov distance measure across multiple dimensions as follows

$$d_{KS}(X, \hat{X}) = \frac{1}{K} \sum_{j=1}^{K} \sup_{x \in \mathbb{R}} \left\|\hat{F}_j(x) - F_j(x)\right\|.$$

Where here applying Rosenblatt (1952), $F_j(X)$ is just the uniform distribution.

The full posterior distributions for the sparse and dense experiments are as follows:

$$\omega_{\text{sparse}} \,|\, \mathbf{z} \sim \text{Dir}\left[800.1, 100.1, 100.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1\right],$$
$$\omega_{\text{dense}} \,|\, \mathbf{z} \sim \text{Dir}\left[112.1, 119.1, 92.1, 98.1, 95.1, 96.1, 102.1, 92.1, 91.1, 103.1\right].$$

For each of the five random seeds, we pick the stepsize giving the best d_{KS} for SGRLD and SCIR from the following options:

Method	h							
SCIR	1.0	5e-1	1e-1	5e-2	1e-2	5e-3	1e-3	
SGRLD	5e-1	1e-1	5e-2	1e-2	5e-3	1e-3	5e-4	1e-4

Table 1: Stepsizes for the synthetic experiment

E.2 Latent Dirichlet Allocation

As mentioned in the main body, we use a decreasing stepsize scheme of the form $h_m = h(1+m/\tau)^{-\kappa}$. We do this to be fair to SGRLD, where the best performance is found by using this decreasing scheme (Patterson and Teh, 2013; Ma et al., 2015); and this will probably reduce some of the bias due to the stepsize h. We find a decreasing stepsize scheme of this form also benefits SCIR, so we use it as well. Notice that we find similar optimal hyperparameters for SGRLD to Patterson and Teh (2013). Table 2 fully details the hyperparameter settings we use for the LDA experiment.

Method	h	τ	κ	α	β	K	n	Gibbs Samples
CIR	0.5	10.	.33	0.1	0.5	100	50	200
SGRLD	0.01	1000.	.6	0.01	0.0001	100	50	200

Table 2: Hyperparameters for the LDA experiment

E.3 Bayesian Nonparametric Mixture

For details of the stochastic slice sampler we use, please refer to Section D. Figure 3 details full hyperparameter settings for the Bayesian nonparametric mixture experiment. Note that h_θ corresponds to the stepsizes assigned for sampling the θ parameters; while h_{DP} corresponds to the stepsizes assigned for sampling from the weights ω for the Dirichlet process.

Method	h_θ	h_{DP}	a	K	n
CIR	0.1	0.1	0.5	20	1000
SGRLD	0.001	0.005	0.001	30	1000

Table 3: Hyperparameters for the Bayesian nonparametric mixture experiment