Online MCMC based Bayesian Inference

Aditya Vikram
14040

Shubham Sharma
14807629

Rahul Gupta
14807519

1 Abstract

In recent years there has been a tremendous increase in amount of data generated and collected. One of the major forms of data is text and their is a need to process it for drawing automated inferences. The sources of data range from internet traffic and network data, computer vision, natural language processing, to bio-informatics. Many models have been constructed using these large scale data which are used to solve a plethora of complex problems. Traditional Monte Carlo Markov Chain(MCMC)[1] Methods can not be applied to these large datasets since it require computations over the whole dataset. Recently Online Machine Learning has been extensively used on these datasets in which data is available in mini batches which are used to update the model parameters as opposed to batch learning in which the entire training data is used at once. Stocastic Gradient Descent(SGD)[2] is one such iterative method which is applied on mini batches of data to find the optimal parameters. Recently, Stochastic Gradient Langevin Dynamics (SGLD)[3] method has been proposed for scaling up Monte Carlo computations to large data problems. However, the method cannot be readily used in constrained domain setting. An extension of SGLD has been proposed to be used on probability simplices, which is called as Stochastic Gradient Riemannian Langevin Dynamics (SGRLD)[4]. This project aims at doing a full-literature survey of SGLD and SGRLD methods and applying it to Latent Dirchlet Allocation(LDA) defined over simplices.

2 Markov Chain Monte Carlo (MCMC) Methods

MCMC consists of class of algorithms which is used to generate samples from some intractable target distribution $p(\theta) = \frac{\hat{p}(\theta)}{Z}$ with a assumption that $\hat{p}(\theta)$ is computable. The algorithms constructs the markov chain which has desired distribution as its equilibrium distribution. Samples are generated from desired distribution by simply observing the chain for large number of steps. The more the steps the closed the samples from desired distribution.

2.1 Metropolis-Hastings (MH) Sampling using Gradient information

MH algorithm constructs a proposal distribution $q(\theta^*|\theta^\tau)$ using the following information

$$\theta^* = \theta^\tau + \frac{\eta}{2}\nabla_z[\log p(D|\theta) + logp(\theta)]|_{\theta^\tau}$$

It then generates a sample from this proposal distribution and accept it with probability

$$A(\theta^*, \theta^\tau) = min(1, \frac{\hat{p}(\theta^*)q(\theta^\tau|\theta^*)}{\hat{p}(\theta^\tau)q(\theta^*|\theta^\tau)})$$

The problem with this approach is that in order to get the new proposal distribution and to decide the acceptance of θ^* we need to calculate the posterior $p(\theta)$ and $q(\theta)$ over whole dataset which is infeasible in general.

3 Stochastic Gradient Langevin Dynamics

3.1 Background

Given a large dataset, Vanilla SGD algorithms (Robbins-Monro type) converge to the MAP or MLE estimate depending on the choice of the loss function. Given input data $X = \{x_1, \cdots, x_N\}$, at each timestep, SGD algorithm randomly selects a small batch $X_t = \{x_{t1}, \cdots, x_{tn}\}$ of data and uses the following update equation:

$$\Delta\theta_t = \frac{\epsilon_t}{2}\left(\nabla \log p(\theta_t) + \frac{N}{n}\sum_{i=1}^{n} \nabla \log p(x_{ti} \,|\, \theta_t)\right)$$

In 2011, Welling and Teh [3] proposed the Stochastic Gradient Langevin Dynamics(SGLD) algorithm, which is aimed at modifying the vanilla SGD algorithm in order to sample points from the full posterior instead of just the MAP solution. It does so by injecting noise, drawn from a zero mean gaussian with defined variance, to the update equation of SGD. The idea is inspired from the work of Neal [5] wherein MCMC algorithms were enhanced using ideas from the solution of a simple kind of Langevin dynamics equation from Physics, namely the Langevin ito diffusion. Given a probability distribution function $p(X)$ over a random variable X, the overdamped Langevin It diffusion equation, driven by a standard Brownian motion W is

$$\dot{X} = \nabla \log p(X) + \sqrt{2}\dot{W}$$

This equation can be solved numerically using the Euler-Maruyama method[6]:

$$X_{k+1} = X_k + \tau\nabla \log p(X_k) + \sqrt{2}\xi_k$$
$$\xi_k \sim \mathcal{N}\left(\mathbf{0}, \tau\mathbf{I}\right)$$

3.2 The SGLD Algorithm

The authors have modified the vanilla SGD update equation to incorporate the solution of Langevin dynamics:

$$\Delta\theta_t = \frac{\epsilon_t}{2}\left(\nabla \log p(\theta_t) + \frac{N}{n}\sum_{i=1}^{n} \nabla \log p(x_{ti} \,|\, \theta_t)\right) + \eta_t$$
$$\eta_t \sim \mathcal{N}\left(0, \epsilon_t\right)$$

The injected noise is linked to the step size because step size is the variance of the Gaussian from which we inject noise. There is still one problem: the MH acceptance step requires evaluation of likelihood and the prior over the entire dataset. But this problem is automatically solved in SGLD by letting the step lengths $\epsilon_t \to 0$. This means that change in θ will be negligible in later stages and so, the MH acceptance rate $\to 1$ after the step lengths become small enough. Thus, there is no need to evaluate probabilities over the entire dataset. We can simply skip the acceptance step. Next, we see the proof of convergence of this algorithm.

3.3 Proof of Convergence

It suffices to show that any sub-sequence of parameters $\theta_{t_1}, \theta_{t_2}, \cdots$ such that $t_1 < t_2 < \cdots$ converges to the true posterior. The step sizes follow the Robbins-Monro conditions:

$$\sum_{t=1}^{\infty} \epsilon_t = \infty \qquad \sum_{t=1}^{\infty} \epsilon_t^2 < \infty$$

Since the step sizes are finite and the *Bolzano Weierstrass theorem* says that every bounded sequence has a convergent sub-sequence, we have:

$$\sum_{t=t_s+1}^{t_{s+1}} \epsilon_t \to \epsilon_0 \quad \text{as } s \to \infty \text{ where } 0 < \epsilon_0 \ll 1$$

We now compare the total noise injected by us and the noise due to stochasticity of the gradient. Total injected noise between time t_s and t_{s+1} is given by

$$\left\| \sum_{t=t_s+1}^{t_{s+1}} \eta_t \right\|_2 \le \sqrt{\sum_{t=t_s+1}^{t_{s+1}} c\epsilon_t} = O\left(\sqrt{\epsilon_0}\right)$$

with high probability where we've used the fact that injected noises at each time step are independent and that a Gaussian random variable is bounded within $c{*}Var$ with high probability. Now, we upper bound the noise due to stochasticity of the gradient batches. The random variable h_t defined below captures randomness of the stochastic minibatch gradient, and has zero mean and finite variance:

$$h_t(\theta) = \frac{N}{n} \sum_{i=1}^{n} \nabla \log p(x_{ti} \mid \theta_t) - \sum_{i=1}^{N} \nabla \log p(x_i \mid \theta_t)$$

Let $g(\theta)$ denote the *true gradient* of the dataset under consideration. Under the Lipschitz assumption for gradients, i.e. $g(\theta_1) - g(\theta_2) \le c(\theta_1\theta_2)$, the stochastic gradient is given by:

$$\begin{aligned}
\sum_{t=t_s+1}^{t_{s+1}} \frac{\epsilon_t}{2}\left(g(\theta_t) + h(\theta_t)\right) &= \frac{\epsilon_0}{2} g(\theta_{ts}) + \left(\sum_{t=t_s+1}^{t_{s+1}} \frac{\epsilon_t}{2} (g(\theta_t) - g(\theta_{ts})) \right) + \sum_{t=t_s+1}^{t_{s+1}} \frac{\epsilon_t}{2} h(\theta_t) \\
&\le \frac{\epsilon_0}{2} g(\theta_{ts}) + \left(\sum_{t=t_s+1}^{t_{s+1}} \frac{\epsilon_t}{2} c(\theta_t - \theta_{ts}) \right) + \sum_{t=t_s+1}^{t_{s+1}} \frac{\epsilon_t}{2} h(\theta_t) \\
&= \frac{\epsilon_0}{2} g(\theta_{ts}) + O(\epsilon_0) + O\left(\sqrt{\sum \epsilon_t^2}\right)
\end{aligned}$$

where the third term captures the variance of $\sum_{t=t_s+1}^{t_{s+1}} \frac{\epsilon_t}{2} h(\theta_t)$, which is $\sum_{t=t_s+1}^{t_{s+1}} \frac{\epsilon_t^2}{4}$.
Thus, the noise due to stochasticity of the gradients is $O(\epsilon_0)$, while that due to the injected noise is $(O(\sqrt{\epsilon_0}))$. So, as the algorithm starts, step sizes are large and the noise due to stochasticity of the gradients dominates the other noise and the algorithm essentially behaves like a vanilla SGD algorithm. As the algorithm proceeds, the step lengths become significantly lower than 1 and the injected noise starts dominating. The algorithm now imitates a MALA, except the acceptance step, thus converging to the true posterior. The authors claim that the transition between these two phases is smooth.
For the algorithm to work, we need to make sure that step sizes decrease to 0 to lower the mixing rate near convergence. However, we don't want the step sizes to be very small otherwise we get a MAP solution. The solution here is to keep the step size constant once it has decreased below a threshold, where that MH rejection rate is negligible.

3.4 Limitations

Due to lack of a proper pre-conditioner, mixing rate in SGLD is unnecessarily slow. This forces SGLD to take large steps in directions of small variance and small steps in directions of large variance, thereby hindering the convergence of the Markov chain. [7] presents an improved algorithm: *Stochastic Gradient Fisher Scoring* for this problem. Besides, SGLD provides no room for constrained optimization, eg: optimization over a probability simplex. Also, gradients required for SGLD maybe inversely proportional to entries in parameters θ (as we'll see in LDA), thereby blowing up when close to zero. Thus, we move on to SGRLD, which tries to overcome these limitations.

4 Riemannian Langevin Dynamics

Langevin dynamics has an isotropic proposal distribution leading to slow mixing if the components of θ have very different scales or if they are highly correlated. A recent approach, the Riemann manifold Metropolis adjusted Langevin algorithm uses a user chosen matrix $G(\theta)$ to precondition in a locally adaptive manner. The Riemannian manifold is the family of probability distributions $p(x|\theta)$ parameterised by θ, for which the expected *Fisher Information Matrix* $\mathcal{I}\theta$ defines a natural Riemannian metric tensor. Any positive definite G defines a Riemannian manifold: a space locally similar to the euclidean space and equipped with a metric on tangents spaces at each point, defined by $G(\theta)$. Tangent spaces can be though of as a generalization of vector spaces for manifolds. For example: Expected Fisher Information matrix $\mathcal{I}\theta$ defines a "natural" Riemannian metric tensor (inspired from *natural* gradients) for a family of probability distributions $p(x|\theta)$. However $\mathcal{I}\theta$ can't be used always because rank problems. In order to solve this problem we can define our own $G(\theta)$.

As in Langevin dynamics, RLD consists of a Gaussian proposal $q(\theta^*|\theta)$, along with a Metropolis Hastings correction step. The update in θ is given by

$$\theta^* = \theta + \frac{\epsilon}{2}\mu(\theta) + G^{\frac{1}{2}}(\theta)\zeta \qquad \zeta \sim \mathcal{N}(0, \epsilon\mathcal{I})$$

where, the j^{th} component of $\mu(\theta)$ is given by:

$$\mu(\theta)_j = \Big(G(\theta)^{-1}\Big(\nabla_\theta \log(p(\theta)) + \sum_{i=1}^{N} \log(p(x_i|\theta))\Big)\Big)_j$$

$$-2\sum_{k=1}^{D}\Big(G(\theta)^{-1}\frac{\partial G(\theta)}{\partial \theta_k}G(\theta)^{-1}\Big)_{jk} + \sum_{k=1}^{D} G(\theta)^{-1}_{jk} Tr\Big(G(\theta)^{-1}\frac{\partial G(\theta)}{\partial \theta_k}\Big)$$

First term in the above equation is the *natural Gradient* of the log posterior, while the other two terms take into account the geometry of the Riemannian manifold in question. The above update gives us the direction of steepest descent taking into account the geometry implied by $G(\theta)$.

5 Stochastic Gradient Riemannian Langevin Dynamics

In the Riemannian Langevin dynamic, the proposal distribution requires calculation of the gradient of the log likelihood w.r.t. θ, which means processing all N items in the data set. SGRLD algorithm replaces the calculation of gradient over the full data set, with a stochastic approximation based on a subset of data $\frac{N}{|D_t|}\sum_{i \in D_t} \nabla_\theta \log p(x_i|\theta)$. Also, SGRLD does not use a Metropolis-Hastings correction step similar to SGLD.

6 SGRLD on the probability simplex

Let us consider a K dimensional probability vector π with Dirichlet prior $p(\pi) \propto \prod_k^K \pi_k^{\alpha_k - 1}$, and data $x = x_1, ..., x_N$ with $p(x_i = k|\pi) = \pi_k$. Then, the posterior is also Dirichlet with $p(\pi|x) \propto \prod_k^K \pi_k^{n_k + \alpha_k - 1}$ where $n_k = \sum_{i=1}^N \delta(x_i = k)$ There are various possible ways to parameterise the probability simplex, and the performance of Langevin Monte Carlo depends strongly on the choice of parameterisation.

Reduced-Mean. In this approach π is directly considered as parameters, however as we can see that the π is constrained and running RLD on K dimensional π will give us result that will be off the simplex, so we will use the first $K - 1$ components as the parameters θ, and setting $\pi_k = 1 - \sum_{k=1}^{K-1} \pi_k$. However it can be noted that Reduced-Mean is still violating the constraint $0 < \pi_k < 1$.

Expanded-Mean. Boundary considerations are simplified by using a redundant parameterisation $\theta_k \in R_+^k$ with prior a product of independent $Gamma(\alpha_k, 1)$ distribution $p(\theta) \propto \prod_{k=1}^K \theta_k^{\alpha_k - 1} \exp(-\theta_k)$ π is then given as $\pi_k = \frac{\theta_k}{\sum_k \theta_k}$, so the prior on π is still Dirichlet. The boundary conditions $0 < \theta_k$ can be handled by simply taking the absolute value of the proposed parameterization.

Reduced Natural. In this parameterization, $\pi_k = \frac{\exp(\theta_k)}{1 + \sum_k \exp(\theta_k)}$ for $k = 1, .., K - 1$. The prior on θ is still Dirichlet. There are no boundary constraints as the range of θ_k is R.

Expanded Natural. In this parameterization, $\pi_k = \frac{\exp(\theta_k)}{\sum_k \exp(\theta_k)}$ for $k = 1, .., K$. As in the expanded-mean parameterisation, we use a product of Gamma priors, in this case for $\exp(\theta_k)$, so that the prior for π remains Dirichlet.

Parameterisation	Reduced-Mean	Reduced-Natural	Expanded-Mean	Expanded-Natural
θ	$\theta_k = \pi_k$	$\theta_k = \log \frac{\pi_k}{1-\sum_k^{K-1} \pi_k}$	$\pi_k = \frac{\lvert\theta_k\rvert}{\sum_{k=1} \lvert\theta_k\rvert}$	$\pi_k = \frac{\mathrm{e}^{\theta_k}}{\sum_{k=1} \mathrm{e}^{\theta_k}}$
$\nabla_\theta \log p(\theta\|\mathbf{x})$	$\frac{n+\alpha}{\theta} - \mathbf{1}\frac{n_K+\alpha-1}{\pi_K}$	$n+\alpha-(n_.+K\alpha)\pi$	$\frac{n+\alpha-1}{\theta} - \frac{n_.}{\theta_.} - \mathbf{1}$	$n+\alpha-n_.\pi-\mathrm{e}^\theta$
$G(\theta)$	$n_. \left(\text{diag}(\theta)^{-1} + \frac{1}{1-\sum_k \theta_k}\mathbf{11}^T\right)$	$\frac{1}{n_.}\left(\text{diag}(\pi) - \pi\pi^T\right)$	$\text{diag}\left(\theta\right)^{-1}$	$\text{diag}\left(\mathrm{e}^\theta\right)$
$G^{-1}(\theta)$	$\frac{1}{n_.}\left(\text{diag}(\theta) - \theta\theta^T\right)$	$n_. \left(\text{diag}(\pi)^{-1} + \frac{1}{1-\sum_k \pi_k}\mathbf{11}^T\right)$	$\text{diag}\left(\theta\right)$	$\text{diag}\left(\mathrm{e}^{-\theta}\right)$
$\sum_{k=1}^D \left(G^{-1}\frac{\partial G}{\partial\theta_k}G^{-1}\right)_{jk}$	$K\theta_j - 1$	$\frac{1}{\pi_j^2} - \frac{K-1}{(1-\sum_k \pi_k)^2}$	-1	$\mathrm{e}^{-\theta_j}$
$\sum_{k=1}^D \left(G^{-1}(\theta)\right)_{jk} \text{Tr}\left(G^{-1}(\theta)\frac{\partial G}{\partial\theta_k}\right)$	$K\theta_j - 1$	$\frac{1}{\pi_j^2} - \frac{K-1}{(1-\sum_k \pi_k)^2}$	-1	$\mathrm{e}^{-\theta_j}$

Table 1: Parameterisation Details

The details for each parameterisation are summarized in Table 1. In all cases we are interested in sampling from the posterior distribution on π, while θ is the specific parameterisation being used. For the mean parameterizations, the θ^{-1} term in the gradient of the log-posterior means that for components of which are close to zero, the proposal distribution for Langevin dynamics has a large mean, resulting in unstable proposals with a small acceptance probability. Due to the form of $G(\theta)^{-1}$, the same argument holds for the RLD proposal distribution for the natural parameterizations. This leaves us with three possible combinations, RLD on the expanded mean parameterisation and Langevin dynamics on each of the natural parameterizations. To demonstrate RLD, we have focused on the former for our experiments.

7 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is a hierarchical Bayesian model. It allows sets of observations to be explained by unobserved groups that explain why some parts of the data are similar. It is therefore most frequently used to model topics arising in collections of text documents where topics are represented by a probability distribution over vocabulary. The model consists of K topics π_k, drawn from a symmetric Dirichlet prior with hyper-parameter β. A document d is then modelled by a mixture of topics, with mixing proportion η_d, drawn from a symmetric Dirichlet prior with hyper-parameter α. The model corresponds to a generative process where documents are produced by drawing a topic assignment z_{di} i.i.d. from η_d for each word w_{di} in document d, and then drawing the word w_{di} from the corresponding topic $\pi_{z_{di}}$.

Now, in our case, applying Riemannian Langevin dynamics to this model by using the Expanded Mean parameterisation, gives us the following updates for the parameter θ

$$\theta^*_{kw} = \left|\theta_{kw} + \frac{\epsilon}{2}\left(\beta - \theta_{kw} + \frac{|D|}{D_t}\sum_{d\in D_t} E_{z_d|w_d,\theta,\alpha}[\eta_{dkw} - \pi_{kw}\eta_{dk}]\right) + (\theta_{kw}^{1/2})\zeta_{kw}\right|$$

where $\zeta \sim \mathcal{N}(0, \epsilon\mathcal{I})$. To calculate this expectation we use Gibbs sampling on the topic assignments in each document separately, using the conditional distributions

$$p(z_{di} = k|w_d, \theta, \alpha) = \frac{\left(\alpha + \eta_{dk}^{\backslash i}\right)\theta_{kw_{di}}}{\sum_k \left(\alpha + \eta_{dk}^{\backslash i}\right)\theta_{kw_{di}}}$$

8 Experiment

We have experimented with SGRLD [4] algorithm for LDA and compared it with Online Variational Bayes [8] for LDA. For our experiments, we used Simple English Wikipedia dump from wikimedia as our dataset. It contains about 127k random articles from Wikipedia. The batch size for both the experiments were kept at 50 articles. The length of vocabulary used is 8000 words which is derived from words in Gutenberg texts excluding words with less than 3 characters because such words are common and do not belong to any subject in particular. SGRLD and OVB implementations were used from [9] and [10] and were modified for our input setting. The parameters of LDA were also kept the same.

We used Perplexity as the standard evaluation metric for these experiments. Perplexity (exponentiated cross entropy) is used as the metric. Perplexity for a document d is given by

$$\text{perp}(d \,|\, W, \alpha, \beta) = \exp\left(-\frac{\sum_{i=1}^{N_d} \log p\left(w_{di} \,|\, W, \alpha, \beta\right)}{N_d}\right)$$

We evaluated Perplexity over a holdout test set having 1000 documents. For each test document, η_d is estimated using all words except every 10^{th} word (90% of the document). Perplexity is calculated on every 10^{th} word of that holdout test document.

9 Results

As expected from theory, SGRLD beats OVB in terms of both runtime and number of iterations for convergence. SGRLD converges at around 20k docs while OVB doesn't converged even after 50000 docs. We found that 1 iteration of SGRLD (50 articles) takes about 0.1 sec, while that of OVB takes $1 - 2$ seconds on our computer.

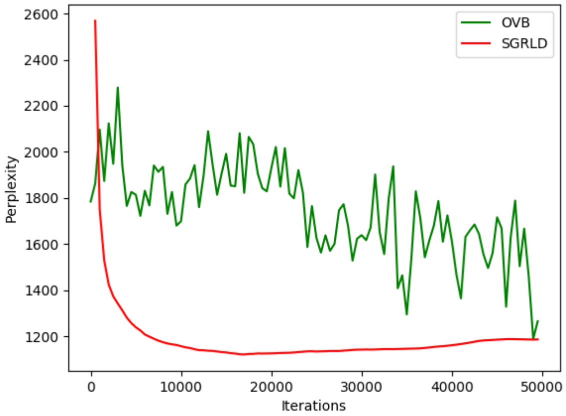

Figure 1: Test Perplexities on Wikipedia Corpus

10 Concepts/Tools Learnt

In this project we have explored SGLD and SGRLD and applied it to LDA. While doing the full literature survey, we have understood how the concept of physics Stochastic Differential Equations is applied in MCMC algorithms. This was our first encounter with the online models. We have learnt how to re-parametrize the distribution to fit into the existing models. We have explored and understood the existing implementation of LDA using SGRLD and OVB models.

11 Positivity constraints: A Discussion

We wanted to derive similar updates for positivity constraints instead of an probability simplex. Suppose there are Gamma priors on some of the parameters, say

$$\theta_j \sim \text{Gamma}(\theta_j \,|\, a_j, b_j) \quad \forall j \in \mathcal{P}$$

where $\mathcal{P}$ is the set of indices of all parameters that require positivity constraints. We use a similar reparameterization trick by exponentiation of a unconstrained variable. But, we couldn't find an appropriate distribution such that its exponentiation leads to a Gamma distribution. We looked at other transform like squaring and taking absolute value, but in vain. So we approximate gamma distribution using exponentiation of a Gaussian random variable by matching the expectations of the approximation to the original Gamma distribution.

$$\theta_j \approx \exp(\frac{w_j}{\sigma^2})$$

$$w_j \sim \mathcal{N}(w_j \,|\, \frac{a_j}{b_j} + 1, \sigma^2)$$

Then, we can similarly write the update equations for this parameterization.

12 Conclusion and Future Work

We set out to do a thorough survey of MCMC methods for online environments. We saw how SGLD borrows concepts from the theory of Stochastic Differential Equations to modify the SGD algorithm and get viable updates for SGLD over mini-batches of data. Then we looked at limitations of SGLD algorithm and how those limitations are overcome in SGRLD. Then, we applied SGRLD to a LDA model and presented the results. SGRLD beats Online Variational Bayes in terms of both runtime and number of iterations for convergence. Finally, we also gave our own parameterization for positive constrained variables with Gamma priors. In the future, we'd like to test how this parameterization works and apply the reparameterization trick to other commonly used models to speed up inference for those models.

References

[1] W. K. Hastings. "Monte Carlo sampling methods using Markov chains and their applications". In: *Biometrika* 57.1 (1970), pp. 97–109. DOI: 10.1093/biomet/57.1.97. eprint: /oup/backfile/content_public/journal/biomet/57/1/10.1093_biomet_57.1.97/1/57-1-97.pdf. URL: http://dx.doi.org/10.1093/biomet/57.1.97.

[2] "A Stochastic Approximation Method". In: *The Annals of Mathematical Statistics* 22.3 (1951), pp. 400–407. ISSN: 00034851. URL: http://www.jstor.org/stable/2236626.

[3] Max Welling and Yee W Teh. "Bayesian learning via stochastic gradient Langevin dynamics". In: *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*. 2011, pp. 681–688.

[4] Sam Patterson and Yee Whye Teh. "Stochastic Gradient Riemannian Langevin Dynamics on the Probability Simplex". In: *Advances in Neural Information Processing Systems 26*. 2013, pp. 3102–3110.

[5] Radford M Neal. "An improved acceptance procedure for the hybrid Monte Carlo algorithm". In: *Journal of Computational Physics* 111.1 (1994), pp. 194–203.

[6] E. Kloeden P.E. & Platen. *Numerical Solution of Stochastic Differential Equations*. Springer, 1992.

[7] Sungjin Ahn, Anoop Korattikara, and Max Welling. "Bayesian posterior sampling via stochastic gradient Fisher scoring". In: *arXiv preprint arXiv:1206.6380* (2012).

[8] Matthew Hoffman, Francis R. Bach, and David M. Blei. "Online Learning for Latent Dirichlet Allocation". In: *Advances in Neural Information Processing Systems 23*. Ed. by J. D. Lafferty et al. Curran Associates, Inc., 2010, pp. 856–864. URL: http://papers.nips.cc/paper/3902-online-learning-for-latent-dirichlet-allocation.pdf.

[9] Sam Patterson and Yee Whye Teh. https://www.stats.ox.ac.uk/~teh/sgrld.html.

[10] https://github.com/blei-lab/onlineldavb.