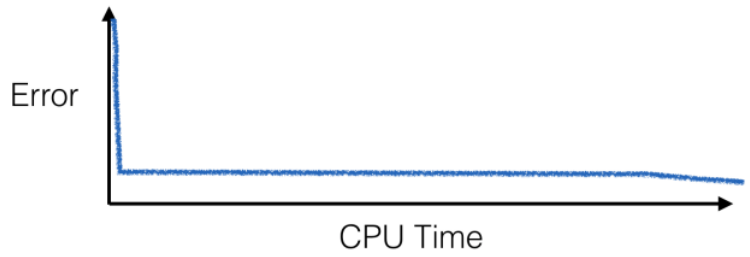
Ongoing: Hybrid of MCMC and Variational Inference with a preconditioned twist

1 Interpolation between (SG)MCMC and variational inference

1.1 Motivation

We apply an interpolation method to combine two major methods for approximate inferene, which are Markov chain Monte Carlo (MCMC) and variational inference (VI.) Our main reference is the paper by Domke [Dom17], which provides a hybrid algorithm which interpolates between Langevin dynamics and reparametrization-based stochastic gradient variational inference (SGVI.) The former is a specific MCMC algorithm (Grenander & Miller [GM94]; Robert & Casella [RC04, Sec. 7.8.5]) which injects noise into the random walk such that the parameters will converge to the full posterior distribution; while the latter is a scalable variational inference algorithm (Hoffman et al. [HBWP13]) which scales to massive data using stochastic optimization.

The motivation for a combination of these two methods is the fact that VI is only an approximate algorithm while MCMC can be very slow. In fact, MCMC may require many orders of magnitude more time to achieve the same accuracy performance as VI. Informally, the interpolation of the performance of these two methods would look like this [1]

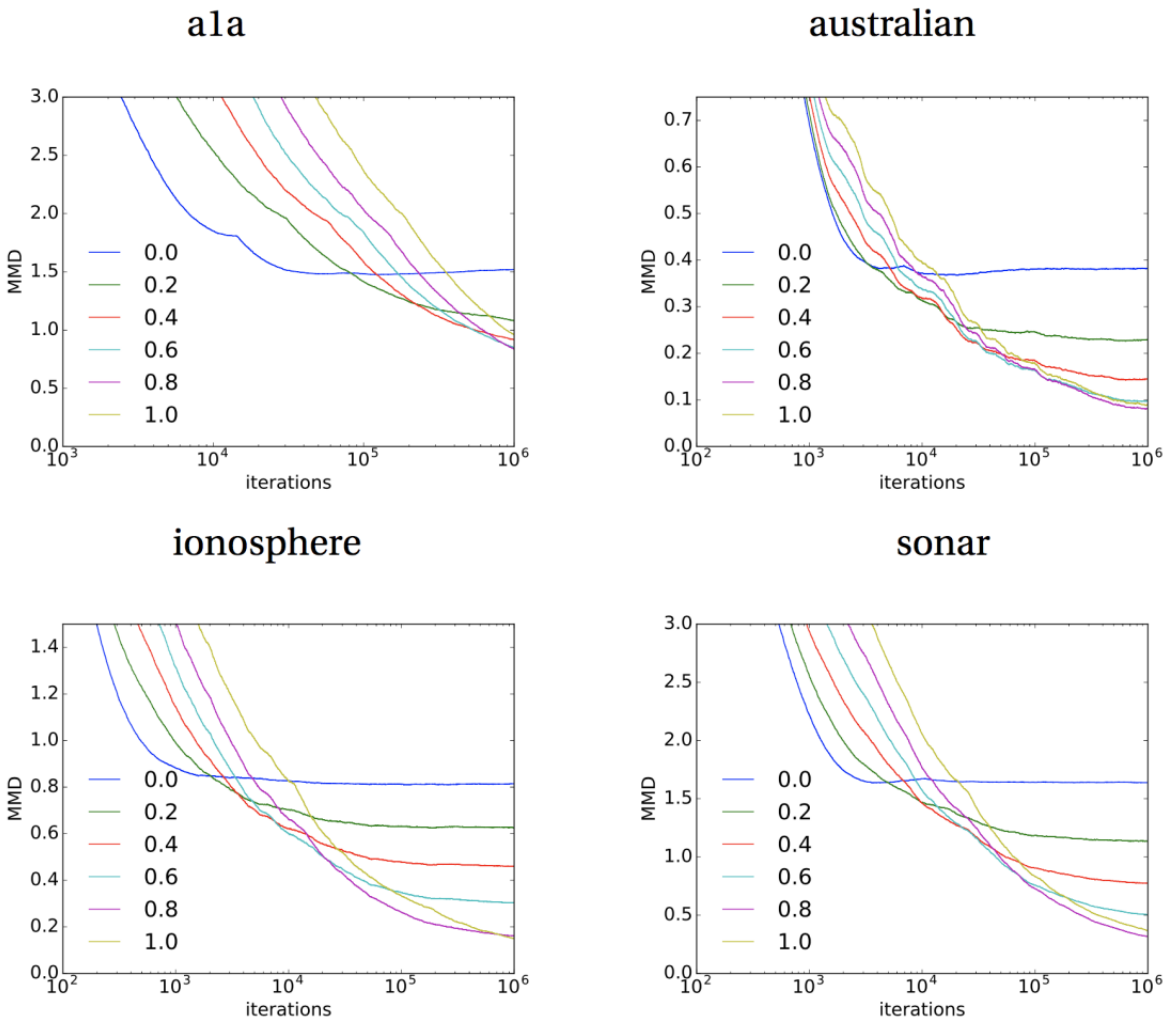


The advantage of such an algorithm would be a reasonable performace on the training error for a period in the training process. The traning error would be reduced faster than the given MCMC algorithm, while also converge lower than the given VI algorithm. In other words, the algorithm would combine the benefits of both MCMC and VI, for a period in the training process. A user with concern on the memory usage would benefit from this algorithm since they would be able to achieve acceptable performance from running fewer epochs than they would with an MCMC algorithm.

As a reference for such achievement we include here the experiments of Domke which achieved the better performation than MCMC on the early epochs (here MMD [GBR+07] is a measure of the distance between the means of two samples in the feature space)[2]

[1] Image courtesy of https://justindomke.wordpress.com/2017/11/16/a-divergence-bound-for-hybrids-of-mcmc-and-variational-inference-and/

[2] See footnote 1



1.2 Domke's interpolation algorithm

We outline the work of Domke [Dom17]. In the Bayesian setting, z denotes the unknown parameters and $p(z)$ a posterior over z. Langevin dynamics iterates

$$z \leftarrow z + \frac{\epsilon}{2}\nabla_z \log p(z) + \sqrt{\epsilon}\eta \tag{1}$$

where η is sampled from a Gaussian distribution and ϵ is the step-size. This random walk is an instance of the MCMC algorithm and will approximate the posterior distribution. Note that the Metropolis-Hastings rejection step is unused here since the acceptance probability approches 1 when $\epsilon \to 0$.

Alternatively, one can perform approximation by VI. Starting with a fixed variational family $q(z|w)$ and a target distribution $p(z)$ over z, the paper seeks a distribution $q(w)$ over the parameter w (based on $q(w)$, the paper produces the random walk (3) below) such that

$$q(z) = \int_w q(w)q(z|w) \approx p(z).$$

As in VI, one can minimize the KL-divergence $KL(q(Z)||p(Z)) = \int_z q(z) \log \frac{q(z)}{p(z)}$, by stochastic gradient ascent, which iterates

$$w \leftarrow w + \frac{\epsilon}{2} \nabla_w KL(q(Z|w) \| p(Z)). \tag{2}$$

Instead, the paper derives a bound $D_\beta = (1-\beta)D_0 + \beta D_1$ on $KL(q(Z)||p(Z))$, where

$$D_0 = KL(q(Z|W)||p(Z)),$$

$$D_1 = KL(q(W,Z)||p(W,Z)).$$

(Technically, the bound D_1 comes from adjoining a distribution $p(w|z)$ to $p(z)$, and hence D_β depends on the choice of $p(w|z)$.)

The paper then derives an explicit formula for the distribution $q^*(w)$ which minimizes D_β (theorem 3 in [Dom17]), upon which one can apply Langevin dynamics to derive the hybrid algorithm

$$w \leftarrow w + \frac{\epsilon}{2} \nabla_w \Big(KL(q(Z|w) \| p(Z)) - \beta H(w) + \beta \log r_\beta(w) \Big) + \sqrt{\epsilon \beta} \eta, \tag{3}$$

where $r_\beta(w)$ can be thought of as a prior to the distribution of w and $H(w)$ is the entropy of w. Note that (3) becomes the SGVI equation (2) when $\beta \to 0$ and becomes the Langevin equation (1) when $\beta \to 1$ ($q(w)$ is chosen by the paper such that high density is given to values of w where $q(Z|w)$ is highly concentrated, hence sampling from $q(w)$ becomes equivalent to sampling from $p(z)$ when $\beta \to 1$.)

For later use, equation (3) can be written as

$$w \leftarrow w + \frac{\epsilon}{2} \nabla_w \Big(\mathbb{E}_{q_w} (\log p(Z) + (\beta - 1) \log q_w(Z)) + \beta \log r_\beta(w) \Big) + \sqrt{\epsilon \beta} \eta. \tag{4}$$

2 Hybrid of preconditioned stochastic gradient Langevin dynamics and variational inference

2.1 Preconditioned stochastic gradient Langevin dynamics

Two MCMC algorithms with strong performace in the context of neural networks was given in Li et al [LCCC16] and Chen et al. [CCG+16]. One of the main ingredients in these papers was the use of a preconditioner, which is a measure of the curvature of the parameter space which is similar to the Fisher information metric [LMV+17]. The idea was to use the gradient of the parameter space to adjust the learning rate appropriately. Hereafter, this MCMC-based optimization method will be called pSGLD, which stands for *preconditioned stochastic Langevin dynamics*. Note that, compared with [LCCC16], the paper [CCG+16] went further in obtaining a smooth transition from pSGLD to stochastic optimization of the parameters.

In order to state our main hybrid algorithm in section (2.2), we quote below the Santa (*Stochastic AnNealing Thermostats with Adaptive momentum*) algorithm in Chen et al [CCG+16]., to which we base our work.

Algoritm 1: Santa algorithm [CCG+16], which uses the idea of pSGLD.

Input: η_t (learning rate), σ, λ, *burnin*, $\beta = \{\beta_1, \beta_2, \dots\} \to \infty$, $\{\zeta_t \in \mathbb{R}^p\} \sim N(0, I)$.
Initialize $\theta_0, u_0 = \sqrt{\eta} \times N(0, I), \alpha_0 = \sqrt{\eta} C, v_0 = 0$;
for t=1,2,... **do**
    Evaluate $\tilde{f}_t \triangleq \nabla_\theta \tilde{U}(\theta_{t-1})$ on the t^{th} mini-batch:
    $v_t = \sigma v_{t-1} + \frac{1-\sigma}{m^2} \tilde{f}_t \odot \tilde{f}_t$;
    $g_t = 1 \oslash \sqrt{\lambda + \sqrt{v_t}}$;
    if $t <$ *burnin* **then**

8 /* exploration
9 $\alpha_t = \alpha_{t-1} + (u_{t-1} \odot u_{t-1} - \eta/\beta_t)$;
10 $u_t = \frac{\eta}{\beta_t}(1 - g_{t-1} \oslash g_t) \oslash u_{t-1} + \sqrt{\frac{2\eta}{\beta_t} g_{t-1}} \odot \zeta_t$;
11 **else**
12 /* refinement
13 $\alpha_t = \alpha_{t-1}$;
14 $u_t = 0$;
15 **end**
16 $u_t = u_t + (1 - \alpha_t) \oslash u_{t-1} - \eta g_t \odot \tilde{f}_t$;
17 $\theta_t = \theta_{t-1} + g_t \odot u_t$;
18 **end**

where $\tilde{U}(\theta)$ is the negative log-posterior

$$\tilde{U}(\theta) \triangleq -\log p(\theta) - \sum_{n=1}^{N} \log p(x_n|\theta)$$

where $p(\theta)$ is the prior for θ and the sum in the second term is log-likelihood.

The interested reader can find in section 4 of [CCG+16] the stochastic differential equations which give rise to the Santa algorithm above. However, the main motivation of Santa is the structural similarities between stochastic gradient MCMC algorithms in Bayesian learning and stochastic optimization methods. This algorithm uses the preconditioned stochastic gradient MCMC algorithms for optimization. A major benefit is that the Bayesian learning is able to fully explore the parameter space. Hence it may theoretically be able to settle in better local optima for non-convex objective functions.

2.2 Main algorithm: Hybrid of pSGLD and variational inference

Instead of sampling θ directly as in the Santa algorithm above, we update a *variational parameter* $w = (\mu, \nu)$ by Santa and then sample $\theta \sim N(\mu, 10^{\nu})$. We denote this distribution as $q_w(\theta)$.

The update needs to modify the gradient of the log-posterior $\tilde{U}(\theta)$ by formula (4) in Domke. The reader should interpret the θ in this section as the z in fomulas (3) and (4) above by Domke. On the other hand, the term $\nabla_w(\mathbb{E}_{q_w}(\log p(Z)))$ in (4) should be interpreted as $\nabla_\theta \tilde{U}(\theta)$. The distribution $q_w(\theta)$ comes from $q_w(Z)$ in formulas (3) and (4).

The pseudo-code is as follows.

Algoritm 2: Hybrid between Santa and VI
Input: η_t (learning rate), σ, λ, *burnin*, $\beta = \{\beta_1, \beta_2, \dots\} \to \infty$, $\{\zeta_t \in \mathbb{R}^p\} \sim N(0, I)$.
Initialize $w_0 = (\mu_0, \nu_0), u_0 = \sqrt{\eta} \times N(0, I), \alpha_0 = \sqrt{\eta} C, v_0 = 0$;
for t=1,2,... **do**
Evaluate $\tilde{f}_t \triangleq \nabla_w \Big(\tilde{U}(w_{t-1}) + (\beta - 1)\mathbb{E}_{q_w} \log q_w(\theta) + \beta r_\beta(w) \Big)$ on the t^{th} mini-batch:
$v_t = \sigma v_{t-1} + \frac{1-\sigma}{m^2} \tilde{f}_t \odot \tilde{f}_t$;
$g_t = 1 \oslash \sqrt{\lambda + \sqrt{v_t}}$;
if $t <$ *burnin* **then**
/* exploration
$\alpha_t = \alpha_{t-1} + (u_{t-1} \odot u_{t-1} - \eta/\beta_t)$;
$u_t = \frac{\eta}{\beta_t}(1 - g_{t-1} \oslash g_t) \oslash u_{t-1} + \sqrt{\frac{2\eta}{\beta_t} g_{t-1}} \odot \zeta_t$;
else
/* refinement
$\alpha_t = \alpha_{t-1}$;
$u_t = 0$;

15 **end**
16 $u_t = u_t + (1-\alpha_t) \oslash u_{t-1} - \eta g_t \odot \tilde{f}_t$;
17 $w_t = w_{t-1} + g_t \odot u_t$;
18 Sample $\theta_t \sim q_{w_t}(\theta)$
19 **end**

Notice in line 4 the modified version of $\tilde{f}_t$. It is now in the form of the gradient term in formula (4) of Domke.

The theoretical foundation of our algorithm is given by theorem 3 of Domke [Dom17], which guarantees that such an update (4) will minimize a certain upper bound on the KL divergence between the true posterior of θ and $\int_w q_w(\theta)q(w)$, as mentioned in subsection (1.2).

Similarly to subsection (1.2), our algorithm 2 will collapse back to the Santa algorithm when $\beta \to 1$ and to SGVI when $\beta \to 0$.

2.3 Experiements

2.3.1 Practical implementation of Algorithm 2

The direct implementation of Algorithm 2 above is highly numerically unstable. To remedy this, we implement a hybrid between pSGLD [LCCC16] and VI as follows.

Algoritm 3: Hybrid between pSGLD and VI
1 **Input:** ϵ_t, σ, λ.
2 **Initialize** $w_0 = (\mu_0, \nu_0), u_0 = \sqrt{\eta} \times N(0, I), \alpha_0 = \sqrt{\eta}C, v_0 = 0$;
3 **for** t=1,2,... **do**
4 Evaluate $\tilde{f}_t \triangleq \nabla_w \Big(\tilde{U}(w_{t-1}) + (\beta - 1)\mathbb{E}_{q_w} \log q_w(\theta) + \beta r_\beta(w) \Big)$ on the t^{th} mini-batch:
5 $v_t = \sigma v_{t-1} + \frac{1-\sigma}{m^2} \tilde{f}_t \odot \tilde{f}_t$;
6 $g_t = 1 \oslash \sqrt{\lambda + \sqrt{v_t}}$;
7 $w_t = w_{t-1} + \frac{\epsilon_t}{2} g_t \tilde{f}_t + N(0, \epsilon_t g_t)$
8 Sample $\theta_t \sim q_{w_t}(\theta)$
9 **end**

where the stepsizes ϵ_t satisfy the following assumption, which is necessary for the asymptotic convergence to the true posterior.

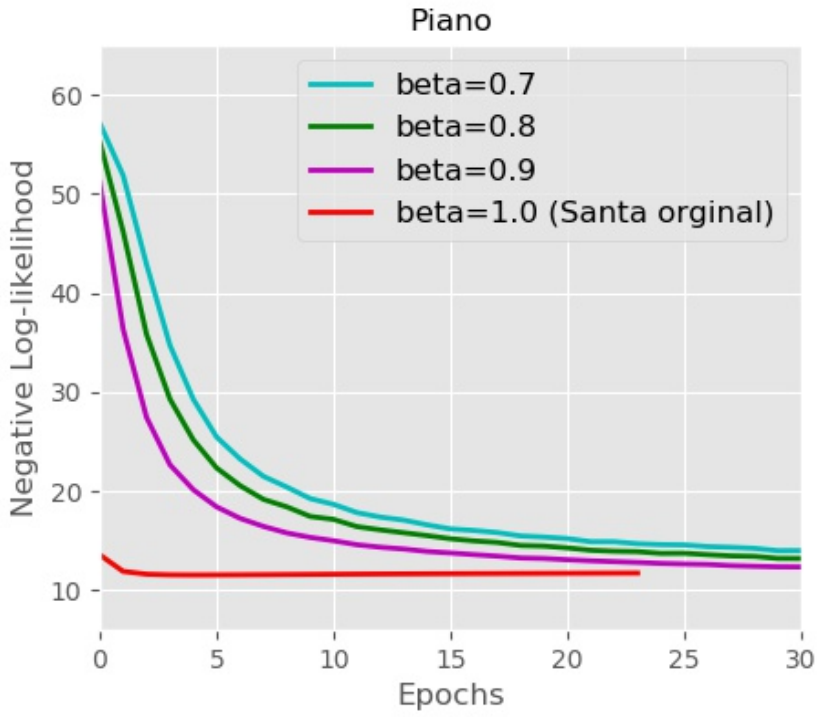
Assumption on stepsizes: ϵ_t are decreasing with

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

2.3.2 Experiments

We apply our algorithm to modify the experiment of Santa on the Recurrent Neural Network for sequence modeling, as presented in section 5.4 of Chen et al. [CCG+16]. We consider the task of sequence modeling on the Piano dataset [BLBV12]. To make prediction of the next musical note in the sequence, the piece of music is discretized into configurations which vary according to time steps. There are 88 possible pitches and hence every time step is one-hot encoded into an 88-dimensional binary vector. The sequence of input vectors is then plugged into an RNN-based architecture (Chen et al. [CCG+16] uses the *gated recurrent unit* - GRU). Note that the number of possible configurations can be *too high* for such a formulation (if there are N possible notes there will be 2^N configurations.) Hence, instead of taking the softmax classification as in image classification and language modeling, one needs to use a sigmoid cross-entropy loss function for prediction (namely, to compute the probability of whether each note is in the configuration or not).

When updating the network parameter θ according to our Algorithm 3, we adopt the hyper-parameters in Chen et al. [CCG+16]. The number of epochs is 40. The number of hidden units is 200. The learning rate is chosen to be 0.0001. The annealing rate is set to 0.5 and the norm of the gradient is truncated to always be less than 5.



The graphs of the negative log likelihood for $\beta \in \{0.9, 0.8, 0.7\}$ all lie entirely above the graph of the Santa algorithm ($\beta = 1.$) We seek to fine tune our hyperparmeters further in oder to achieve better performance for $\beta \in \{0.9, 0.8, 0.7\}$ and to beat Santa for the earlier training epochs.

3 Literature Review

We have done extensive review on the literature of the subjects of Langevin MCMC and SGD. On the Langevin MCMC side, 1994 Grenander [GM94] stated Langevin Monte Carlo, and Besag [BG93] discussed the case of Bayesian inference. Then 1996 [RT+96] Roberts and Tweedie proved the exponential convergence of Langevin MCMC to target distribution. On the SGD side, SGD was first proposed by Robbins, Monroe and Sutton in 1951 [RM51]. After the development of Langevin MCMC in 1990s', in 2011 Welling and Teh [WT11] combined Langevin MCMC and SGD into stochastic Langevin MCMC.

References

[BG93] Julian Besag and Peter J Green. Spatial statistics and bayesian computation. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 25–37, 1993.

[BLBV12] Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. *arXiv preprint arXiv:1206.6392*, 2012.

[CCG+16] Changyou Chen, David Carlson, Zhe Gan, Chunyuan Li, and Lawrence Carin. Bridging the gap between stochastic gradient mcmc and stochastic optimization. In *Artificial Intelligence and Statistics*, pages 1051–1060, 2016.

[Dom17] Justin Domke. A divergence bound for hybrids of mcmc and variational inference and an application to langevin dynamics and sgvi. *Proceedings of the 34th International Conference on Machine Learning*, 2017.

[GBR+07] Arthur Gretton, Karsten M Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex J Smola. A kernel method for the two-sample-problem. In *Advances in neural information processing systems*, pages 513–520, 2007.

[GM94] Ulf Grenander and Michael I Miller. Representations of knowledge in complex systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 549–603, 1994.

[HBWP13] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

[LCCC16] Chunyuan Li, Changyou Chen, David E Carlson, and Lawrence Carin. Preconditioned stochastic gradient langevin dynamics for deep neural networks. In *AAAI*, volume 2, page 4, 2016.

[LMV+17] Alexander Ly, Maarten Marsman, Josine Verhagen, Raoul PPP Grasman, and Eric-Jan Wagenmakers. A tutorial on fisher information. *Journal of Mathematical Psychology*, 80:40–55, 2017.

[RC04] Christian Robert and George Casella. *Monte carlo statistical methods*. Springer-Verlag, 2004.

[RM51] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

[RT+96] Gareth O Roberts, Richard L Tweedie, et al. Exponential convergence of langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341–363, 1996.

[WT11] 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)*, pages 681–688, 2011.