Relativistic Monte Carlo

Xiaoyu Lu[*]
Department of Statistics
University of Oxford

Valerio Perrone[*]
Department of Statistics
University of Warwick

Leonard Hasenclever
Department of Statistics
University of Oxford

Yee Whye Teh
Department of Statistics
University of Oxford

Sebastian Vollmer
Department of Statistics
University of Warwick

Abstract

Hamiltonian Monte Carlo (HMC) is a popular Markov chain Monte Carlo (MCMC) algorithm that generates proposals for a Metropolis-Hastings algorithm by simulating the dynamics of a Hamiltonian system. However, HMC is sensitive to large time discretizations and performs poorly if there is a mismatch between the spatial geometry of the target distribution and the scales of the momentum distribution. In particular the mass matrix of HMC is hard to tune well.

In order to alleviate these problems we propose relativistic Hamiltonian Monte Carlo, a version of HMC based on relativistic dynamics that introduces a maximum velocity on particles. We also derive stochastic gradient versions of the algorithm and show that the resulting algorithms bear interesting relationships to gradient clipping, RMSprop, Adagrad and Adam, popular optimisation methods in deep learning. Based on this, we develop relativistic stochastic gradient descent by taking the zero-temperature limit of relativistic stochastic gradient Hamiltonian Monte Carlo. In experiments we show that the relativistic algorithms perform better than classical Newtonian variants and Adam.

1 Introduction

Markov chain Monte Carlo (MCMC) techniques based on continuous-time physical systems allow the efficient simulation of posterior distributions, and are an important mainstay of Bayesian machine learning and statistics. Hamiltonian Monte Carlo (HMC) [1, 2, 3, 4, 5] is based on Newtonian dynamics on a frictionless surface, and has been argued to be more efficient than techniques based on diffusions [6]. On the other hand, stochastic gradient MCMC techniques based on diffusive dynamics [7, 8, 9, 10] have allowed scalable Bayesian learning using mini-batches.

An important consideration when designing such MCMC algorithms is adaptation or tuning to the geometry of the space under consideration [11, 12, 13]. To give a concrete example, consider HMC. Let $f(\theta)$ be a target density that can be written as $f(\theta) \propto e^{-U(\theta)}$ where $U(\theta)$ is interpreted as the potential energy of a particle in location θ. HMC introduces an auxiliary momentum variable p so that the joint distribution is $f(\theta, p) \propto e^{-H(\theta,p)}$ where the Hamiltonian is $H(\theta, p) = U(\theta) + \frac{1}{2m} p^\top p$. The quantity $\frac{1}{2m} p^\top p$, where m is the mass of the particle, represents the kinetic energy. Denoting by $\dot{\theta}$ and $\dot{p}$ the time derivative of θ and p,

[*]These authors contributed equally.

the leapfrog discretisation [2] of Hamilton's equations $\dot{\theta} = \frac{\partial H}{\partial p}$ and $\dot{p} = -\frac{\partial H}{\partial \theta}$ gives

$$\begin{aligned} p_{t+1/2} &\leftarrow p_t - \tfrac{1}{2}\epsilon \nabla U(\theta_t), \\ \theta_{t+1} &\leftarrow \theta_t + \epsilon \tfrac{p_{t+1/2}}{m}, \\ p_{t+1} &\leftarrow p_{t+1/2} - \tfrac{1}{2}\epsilon \nabla U(\theta_{t+1}) \end{aligned}$$

where ϵ is the time discretisation and the velocity is $\frac{p_{t+1/2}}{m}$. If m is too small, the particle travels too fast leading to an accumulation of discretisation error. To compensate, ϵ needs to be set small and the computational cost required increases. On the other hand, if m is too large, the particle travels slowly resulting in slow mixing for the resulting Markov chain. While the mass parameter can be tuned, e.g., to optimise acceptance rate according to theory [12], it only incidentally controls the velocity which ultimately affects the discretisation error and algorithm stability.

In this paper, we are interested in making MCMC algorithms based on physical simulations more robust by directly controlling the velocity of the particle. This is achieved by replacing the Newtonian dynamics in HMC with the relativistic dynamics [14], whereby particles cannot travel faster than the "speed of light". We also develop relativistic variants of stochastic gradient MCMC algorithms and show that they work better and are more robust than the classical Newtonian variants.

The relativistic MCMC algorithms we develop have interesting relationships with a number of optimisation algorithms popular in deep learning. Firstly, the maximum allowable velocity (speed of light) is reminiscent of gradient clipping [15]. Our framework gives Bayesian alternatives to gradient clipping, in the sense that our algorithms demonstrably sample from instead of optimising the target distribution (exactly or approximately). Secondly, the resulting formulas (see (3)), which include normalisations by L_2 norms, bear strong resemblances to (but are distinct from) RMSprop, Adagrad and Adam [16, 17, 18]. Motivated by these connections, we develop a relativistic stochastic gradient descent (SGD) algorithm by taking the zero-temperature limit of relativistic SGHMC, and show in experiments on feedforward networks trained on MNIST that it achieves better performance than Adam.

2 Relativistic Hamiltonian Dynamics

Our starting point is the Hamiltonian that governs the dynamics in special relativity [14],

$$H(\theta, p) = U(\theta) + K(p) \tag{1}$$

$$K(p) = mc^2 \left(\frac{p^\top p}{m^2 c^2} + 1 \right)^{\frac{1}{2}} \tag{2}$$

where the target density is $f(\theta) \propto e^{-U(\theta)}$, for $\theta \in \mathbb{R}^d$ interpreted as the position of the particle, $p \in \mathbb{R}^d$ is a momentum variable, and $K(p)$ is the relativistic kinetic energy. The two tunable hyperparameters are a scalar "rest mass" m and the "speed of light" c which bounds the particle's speed. The joint distribution $f(\theta, p) \propto e^{-H(\theta,p)}$ is separable, with the momentum variable having marginal distribution $\propto e^{-K(p)}$, a multivariate generalisation of the symmetric hyperbolic distribution.

The resulting dynamics is given by Hamilton's equations, which read

$$\begin{aligned} \dot{\theta} &= \frac{\partial H}{\partial p} = M^{-1}(p)p, & M(p) &= m \left(\frac{p^\top p}{m^2 c^2} + 1 \right)^{\frac{1}{2}} \\ \dot{p} &= -\frac{\partial H}{\partial \theta} = -\nabla U(\theta), \end{aligned} \tag{3}$$

where $M(p)$ can be interpreted as the relativistic mass and $M^{-1}(p)p$ as the velocity of the particle (c.f. the velocity under Newtonian dynamics is $m^{-1}p$). Note that the relativistic mass is lower bounded by and increases asymptotically to $\|p\|/c$ as the momentum increases, so that the speed $M^{-1}(p)\|p\|$ is upper bounded by and asymptototes to c. On the other hand, the larger the rest mass m, the smaller the typical "cruising" speed of the particle.

Conversely, as $m \to 0$ the particle will travel at the speed of light at all times, i.e. it behaves like a photon. This gives an intuition for tuning both hyperparameters c and m based on knowledge about the length scale of the target density: we choose c as an upper bound on the speed at which the parameter of interest θ changes at each iteration, while we choose m to control the typical sensible speed at which the parameter changes. We will demonstrate this intuition in the experimental Section 5.

In very high dimensional problems (e.g. those in deep learning, collaborative filtering or probabilistic modelling), the maximum overall speed imposed on the system might need to be very large so that reasonably large changes in each coordinate are possible at each step of the algorithm. This means that each coordinate could in principle achieve a much higher speed than desirable. An alternative approach is to upper bound the speed at which each coordinate changes by choosing the following relativistic kinetic energy

$$K(p) = \sum_{j=1}^{d} m_j c_j^2 \left(\frac{p_j^2}{m_j^2 c_j^2} + 1 \right)^{\frac{1}{2}}, \tag{4}$$

where j indexes the coordinates of the d-dimensional system, and each coordinate can have its own mass m_j and speed of light c_j. This leads to the same Hamiltonian dynamics (3), but with all variables interpreted as vectors, and all arithmetic operations interpreted as element-wise operations. Experimental results will be based on this independent-momenta variant, which showed consistently better performance. For simplicity, in the theoretical sections we will describe only the version (2).

2.1 Relativistic Hamiltonian Monte Carlo

As a demonstration of the relativistic Monte Carlo framework, we derive a relativistic variant of the Hamiltonian Monte Carlo (HMC) algorithm [2, 1]. In the following, we will refer to all classical variants as Newtonian, since they follow Newtonian dynamics (e.g. *Newtonian HMC (NHMC)* vs *relativistic HMC (RHMC)*).

Each iteration of HMC involves first sampling the momentum variable, followed by a series of L leapfrog steps, followed by a Metropolis-Hastings accept/reject step. The momentum can be simulated by first simulating the speed $\|p\|$ followed by simulating p uniformly on the sphere with radius $\|p\|$. The speed $\|p\|$ has marginal distribution given by a symmetric hyperbolic distribution, for which specialised random variate generators exist. Alternatively, the density is log-concave, and we used adaptive rejection sampling to simulate it. The leapfrog steps [19] with stepsize ϵ follows (3) directly: set θ_0, p_0 to the current location and momentum and for $t = 1, \ldots, L$,

$$\begin{aligned} p_{t+1/2} &\leftarrow p_t - \tfrac{1}{2}\epsilon \nabla U(\theta_t) \\ \theta_{t+1} &\leftarrow \theta_t + \epsilon M^{-1}(p_{t+1/2}) p_{t+1/2} \\ p_{t+1} &\leftarrow p_{t+1/2} - \tfrac{1}{2}\epsilon \nabla U(\theta_{t+1}) \end{aligned}$$

The leapfrog steps leave the Hamiltonian H approximately invariant and are volume-preserving [20], so that the MH acceptance probability is simply $\min(1, \exp(-H(\theta_L, p_L) + H(\theta_0, p_0)))$.

Observe that the momentum p is unbounded and may become very large in the presence of large gradients in the potential energy. However, the size of the θ update is bounded by ϵc and therefore the stability of the proposed sampler can be controlled. This behaviour is essential for good algorithmic performance on complex models such as neural networks, where the scales of gradients can vary significantly across different parameters and may not be indicative of the optimal scales of parameter changes. This is consistent with past experiences optimising neural networks, where it is important to adapt the learning rates individually for each parameter so that typical parameter changes stay in a sensible range [15, 16, 17, 18, 21]. Such adaptation techniques have also been explored for stochastic gradient MCMC techniques [22, 23], but we will argue in Sections 3.2 and 5 that they introduce another form of instability that is not present in the relativistic approach.

3 Relativistic Stochastic Gradient Markov Chain Monte Carlo

In recent years stochastic gradient MCMC (SGMCMC) algorithms have been very well explored as methods to scale up Bayesian learning by using mini-batches of data [7, 10, 9, 8, 24]. In this section we develop relativistic variants of SGHMC [10] and SGNHT [9, 24]. These algorithms include momenta, which serve as reservoirs of previous gradient computations, thus can integrate and smooth out gradient signals from previous mini-batches of data. As noted earlier, because the momentum can be large, particularly as the stochastic gradients can have large variance, the resulting updates to θ can be overly large, and small values of the step size are required for stability, leading potentially to slower convergence. This motivates our development of relativistic variants.

We make use of the framework of [8] for deriving SGMCMC algorithms. Let z be a collection of variables with target distribution $f(z) \propto e^{-H(z)}$. Consider an SDE in the form

$$\begin{aligned} dz &= -[D(z) + Q(z)]\nabla H(z)dt + \Gamma(z)dt + \sqrt{2D(z)}dW \\ \Gamma_i(z) &= \sum_{j=1}^{d} \tfrac{\partial [D_{ij}(z)+Q_{ij}(z)]}{\partial z_j} \end{aligned} \tag{5}$$

where $D(z)$ is a symmetric positive-definite diffusion matrix, $Q(z)$ is a skew-symmetric matrix describing an energy-conserving dynamics, $\Gamma(z)$ is a correction factor, and W is the d-dimensional Wiener process (Brownian motion). [8] showed that under mild conditions the SDE converges to the desired stationary distribution $f(z)$. Hence in the following we simply have to choose the appropriate z, D and Q. Once the correction factor Γ is computed, the SDE discretised, and a stochastic estimate $\nabla\tilde{U}(z)$ for $\nabla U(z)$ substituted, we obtain a correct relativistic SGMCMC algorithm. The stochastic gradient has asymptotically negligible variance compared to the noise injected by W.

3.1 Relativistic Stochastic Gradient Hamiltonian Monte Carlo

Suppose our noisy gradient estimate $\nabla\tilde{U}(\theta)$ of $\nabla U(\theta)$ is based on a minibatch of data. Then, appealing to the central limit theorem, we can assume that $\nabla\tilde{U}(\theta) \approx \nabla U(\theta) + \mathcal{N}(0, B(\theta))$. Let $z = (\theta, p)$ and $H(z)$ be the relativistic Hamiltonian in (2). Choosing

$$D(z) = \begin{pmatrix} 0 & 0 \\ 0 & D \end{pmatrix}, \quad Q(z) = \begin{pmatrix} 0 & -I \\ I & 0 \end{pmatrix}, \tag{6}$$

and thus $\Gamma(z) = \mathbf{0}$, where D is a fixed symmetric diffusion matrix results in the following relativistic SGHMC dynamics:

$$\begin{pmatrix} d\theta \\ dp \end{pmatrix} = \begin{pmatrix} M^{-1}(p)p \\ -\nabla U(\theta) - DM^{-1}(p)p \end{pmatrix} dt + \begin{pmatrix} 0 & 0 \\ 0 & \sqrt{2D} \end{pmatrix} dW_t$$

Using a simple Euler-Maruyama discretisation, the relativistic SGHMC algorithm is,

$$\begin{aligned} p_{t+1} &\leftarrow p_t - \epsilon_t \nabla\tilde{U}(\theta_t) - \epsilon_t DM^{-1}(p_t)p_t + \mathcal{N}(0, \epsilon_t(2D - \epsilon_t\hat{B}_t)) \\ \theta_{t+1} &\leftarrow \theta_t + \epsilon_t M^{-1}(p_{t+1})p_{t+1} \end{aligned} \tag{7}$$

where $\hat{B}$ is an estimate of the noise coming from the stochastic gradient $B(\theta)$. The term $DM^{-1}(p)p$ can be interpreted as friction, which prevents the kinetic energy to build up and corrects for the noise coming from the stochastic gradient.

It is useful to compare RSGHMC with preconditioned SGLD [22, 23], which attempts to adapt the SGLD algorithm to the geometry of the space, using adaptations similar to RMSProp, Adagrad or Adam. The relevant term above is the update $M^{-1}(p_{t+1})p_{t+1}$ to θ_{t+1}:

$$M^{-1}(p_{t+1})p_{t+1} = \frac{p_{t+1}}{\sqrt{\frac{p_{t+1}^\top p_{t+1}}{c^2} + m^2}} \tag{8}$$

Note the surprising similarity to RMSProp, Adagrad and Adam, with the main difference being that the relativistic mass adaptation uses the current momentum instead of being separately estimated using the square of the gradient. This has the advantage that the relativistic SGHMC enforces a maximum speed of change. In contrast, as we also observe in Section 5, preconditioned SGLD has the following failure mode: when the gradient is small, the adaptation scales up the gradient so that the gradient update has a reasonable size. However it also scales up the injected noise, which can end up being significantly larger than the gradient update, and making the algorithm unstable.

3.2 Relativistic Stochastic Gradient Descent (with Momentum)

Motivated by the relationship to RMSprop, Adagrad and Adam, we develop a relativistic stochastic gradient descent (RSGD) algorithm with momentum by taking the zero-temperature limit of the RSGHMC dynamics. This idea connects to Santa [25], a recently developed algorithm where an annealing scheme on the system temperature makes it possible to obtain a stochastic optimization algorithm starting from a Bayesian one.

From thermodynamics [26], the canonical (Gibbs Boltzmann) density is proportional to $e^{-\beta U(z)}$, where β is the inverse temperature. Previously we have been using $\beta = 1$, which corresponds to the posterior distribution. For general β,

$$\begin{pmatrix} d\theta \\ dp \end{pmatrix} = \begin{pmatrix} \beta M^{-1}(p)p \\ \beta\left(-\nabla U(\theta) - DM^{-1}(p)p\right) \end{pmatrix} dt + \begin{pmatrix} 0 & 0 \\ 0 & \sqrt{2D} \end{pmatrix} dW$$

By taking $\beta \to \infty$ the target distribution becomes more peaked around the MAP estimator. Simulated annealing [27, 28, 25], which increases $\beta \to \infty$ over time, forces the sampler to converge to a MAP estimator. We can derive RSGD by rescaling time as well, guaranteeing a non-degenerate limit process. Letting $\hat{\theta}(t) = \theta(\beta t)$, $\hat{p}(t) = p(\beta t)$, so that

$$\begin{pmatrix} d\hat{\theta} \\ d\hat{p} \end{pmatrix} = \begin{pmatrix} M^{-1}(\hat{p})\hat{p} \\ -\nabla U(\hat{\theta}) - DM^{-1}(\hat{p})\hat{p} \end{pmatrix} dt + \begin{pmatrix} 0 & 0 \\ 0 & \sqrt{\frac{2D}{\beta}} \end{pmatrix} dW$$

and letting $\beta \to \infty$, we obtain the following ODE:

$$\begin{pmatrix} d\theta \\ dp \end{pmatrix} = \begin{pmatrix} M^{-1}(p)p \\ -\nabla U(\theta) - DM^{-1}(p)p \end{pmatrix} dt$$

Discretising the above then gives RSGD. Notice that if the above converges, i.e. $\dot{\theta} = \dot{p} = 0$, it does so at a critical point of U. Similar to other adaptation schemes, RSGD adaptively rescales the learning rates for different parameters, which enables effective learning especially in high dimensional settings. Moreover, the update in each iteration is upper bounded by the speed of light. Our algorithm differs from others through the use of a momentum, and adapting based on the momentum instead of the average of squared gradients.

4 A Stochastic Gradient Nosé-Hoover Thermostat for Relativistic Hamiltonian Monte Carlo

Borrowing a second concept from physics, SGHMC can be improved by introducing a dynamic variable ξ that adaptively increases or decreases the momenta. The new variable ξ can be thought of as a *thermostat* in a statistical physics setting and its dynamics expressed as

$$d\xi = \frac{1}{d}\left(p^T p - d\right) dt. \tag{9}$$

The idea is that the system adaptively changes the friction for the momentum, 'heating' or 'cooling down' the system. The dynamics of this new variable, known as Nosé-Hoover [29] thermostat due to its links to statistical physics, has been shown to be able to remove the additional bias due to the stochastic gradient provided that the noise is isotropic Gaussian and spatially constant ([9],[20]). In general, the noise is neither Gaussian, spatially constant or isotropic. Nevertheless, there is numerical evidence that the thermostat increases stability

and mixing. Heuristically, the dynamics for ξ can be motivated by the fact that in equilibrium we have

$$\mathbb{E}\left[\frac{\partial^2 K}{\partial p_i^2}\right] = \int \frac{\partial^2 K}{\partial p_i^2} e^{-K(p)} dp = -\int \frac{\partial K}{\partial p_i}\left(-\frac{\partial K}{\partial p_i} e^{-K(p)}\right) dp = \mathbb{E}\left[\left(\frac{\partial K}{\partial p_i}\right)^2\right].$$

Since $\frac{\partial K}{\partial p_i} = p_i$ and $\frac{\partial^2 K}{\partial p_i^2} = 1$ this implies

$$\mathbb{E}\left[\frac{d\xi}{dt}\right] = \mathbb{E}\left[\frac{1}{d}\sum_i \left(\left(\frac{\partial K}{\partial p_i}\right)^2 - \frac{\partial^2 K}{\partial p_i^2}\right)\right] = 0$$

The additional dynamics pushes the system towards $\frac{d\xi}{dt} = 0$, suggesting that the distribution will be moved closer to the equilibrium. This gives a recipe for a stochastic gradient Nosé-Hoover thermostat with a general kinetic energy $K(p)$.

We first augment the Hamiltonian with ξ:

$$H(q, p, \xi) = U(q) + K(p) + \frac{d}{2}(\xi - D)^2.$$

We are now in the position to derive the SDE preserving the probability density $\propto \exp(-H)$ by adopting the framework of [8] and defining:

$$\begin{aligned}
H(\theta, p, \xi) &= U(\theta) + K(p) + \frac{d}{2}(\xi - D)^2 \\
D(\theta, p, \xi) &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & D \cdot I & 0 \\ 0 & 0 & 0 \end{pmatrix} \\
Q(\theta, p, \xi) &= \begin{pmatrix} 0 & -I & 0 \\ I & 0 & \nabla K(p)/d \\ 0 & -\nabla K(p)^T/d & 0 \end{pmatrix}.
\end{aligned}$$

From (5) it follows that $\Gamma = (0\ 0\ -\Delta K(p)/d)^T$ and the dynamics becomes

$$\begin{pmatrix} d\theta \\ dp \\ d\xi \end{pmatrix} = \begin{pmatrix} \nabla K(p) \\ -\nabla \tilde{U} dt - \xi \nabla K(p) \\ \frac{1}{d}\left(\|\nabla K(p)\|^2 - \Delta K(p)\right) \end{pmatrix} dt + \begin{pmatrix} 0 & 0 & 0 \\ 0 & \sqrt{2D} & 0 \\ 0 & 0 & 0 \end{pmatrix} dW_t$$

where Δ is the Laplace operator defined as $\Delta K(p) = \sum_i \frac{\partial^2 K(p)}{\partial p_i^2}$. For the relativistic kinetic energy $K(p)$, we have that $\nabla_p K(p) = M^{-1}(p)p$ with $M(p) := m\left(\frac{p^T p}{m^2 c^2} + 1\right)^{\frac{1}{2}}$ a scalar and that $\Delta K(p) = tr\left(\frac{d}{dp}\left(\frac{1}{d} M^{-1}(p)p\right)\right)$. The Stochastic Gradient Nosé-Hoover Thermostat for relativistic HMC follows:

$$\begin{pmatrix} d\theta \\ dp \\ d\xi \end{pmatrix} = \begin{pmatrix} M^{-1}(p)p dt \\ -\nabla \tilde{U} - \xi M^{-1}(p)p \\ \frac{p^T p}{d}\left(M^{-2}(p) + c^{-2} M^{-3}(p)\right) - M^{-1}(p) \end{pmatrix} dt + \begin{pmatrix} 0 & 0 & 0 \\ 0 & \sqrt{2D} & 0 \\ 0 & 0 & 0 \end{pmatrix} dW_t.$$

5 Experiments

5.1 Small examples

All the experimental results in this section are based on the independent-momenta versions (4) as they give superior results. We first explore the performances of the algorithms on a set of small examples including a two dimensional banana function (Banana) [30] with density $p(\mathbf{x}) \propto \exp\{-0.5(0.01x_1^2 + (x_2 + 0.1x_1^2 - 10)^2)\}$, and Gaussian mixture models (GMM1, GMM2, GMM3) obtained by combining the three following Gaussian random variables

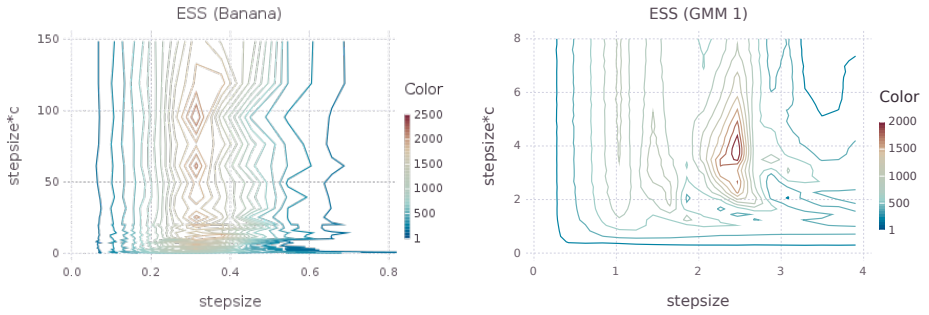

Figure 1: ESS contour plots of $\epsilon \times c$ versus ϵ for Banana and GMM1 models.

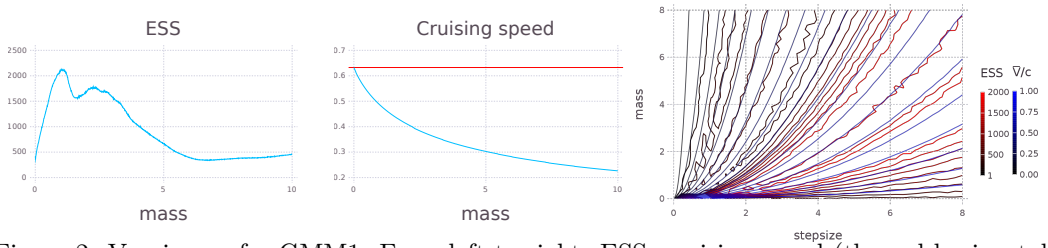

Figure 2: Varying m for GMM1. From left to right: ESS, cruising speed (the red horizontal line is c), and ESS and relative cruising speed $\bar{v}/c$ contour plots versus m and ϵ.

with equal mixing proportions: $\mathcal{N}(-5, 1/\sigma^2)$, $\mathcal{N}(0, \sigma^2)$, $\mathcal{N}(5, 1/\sigma^2)$, where $\sigma^2 = 1, 0.5, 0.3$. When $\sigma^2 = 1$ the three Gaussians have the same variance, while a lower σ^2 means larger discrepancies between their variances and thus a wider range of length scales and log density gradients. The density plots of the examples can be found in the top row of Figure 3.

We start with an exploration of the behaviour of RHMC as the tuning parameters m, c and ϵ are varied. First we considered the effective sample sizes (ESS) of the algorithm on the Banana and GMM1 datasets. We varied both ϵ and $\epsilon \times c$ over a grid, and computed the average ESS, over 20 chains, each of length 10^4 for Banana, and over 100 chains of length 10^5 for GMM1. The ESS contour plots can be found in Figure 1, which suggests that ϵc and ϵ can be independently tuned. While ϵ controls the time discretisation of the continuous-time dynamics, ϵc controls the maximum change in the parameters at each leapfrog step. Next we varied the mass parameter m for GMM1, showing plots in Figure 2. As expected the ESS is optimised at an intermediate value of m, and the average "cruising speed" $\bar{v}$ decreases with m. In order to understand how to tune m, on the third panel we overlaid two contour plots: one for ESS and the other for $\bar{v}$. We see that the cruising speed $\bar{v}$ correlates much better with the ESS than m does, which suggests that m should be tuned via $\bar{v}$, e.g. by the user specifying a desired value for $\bar{v}$ and m being adapted to achieve the speed (noting that m and $\bar{v}$ have a monotonic relationship, which makes for easy adaptation).

We next compare the performances of NHMC and RHMC for a wide range of stepsizes, via the ESS (higher better), the *mean absolute error (MAE)* between the true probabilities and the histograms of the sample frequencies (lower better), and the log Stein discrepancy [31] (lower better). The Stein discrepancy is a more accurate measure of sample quality, the reason being that it can be used to bound the Wasserstein distance, thus accounting for bias and insufficient exploration of the target. The results can be found in rows 2-4 of Figure 3. It can be seen that RHMC achieves better performance and is strikingly more robust to the step size ϵ than NHMC. As expected, this behaviour is particularly pronounced when the step size is large. Moreover, when the gradients of the target model span a large range of values (GMM2, GMM3), the improvements yielded by the relativistic variants are more pronounced. These results confirm that, since the speed of particles is bounded by c, RHMC is less sensitive to the presence of large gradients in the target density and more stable with respect to the choice of ϵ, allowing for a more efficient exploration of the target density.

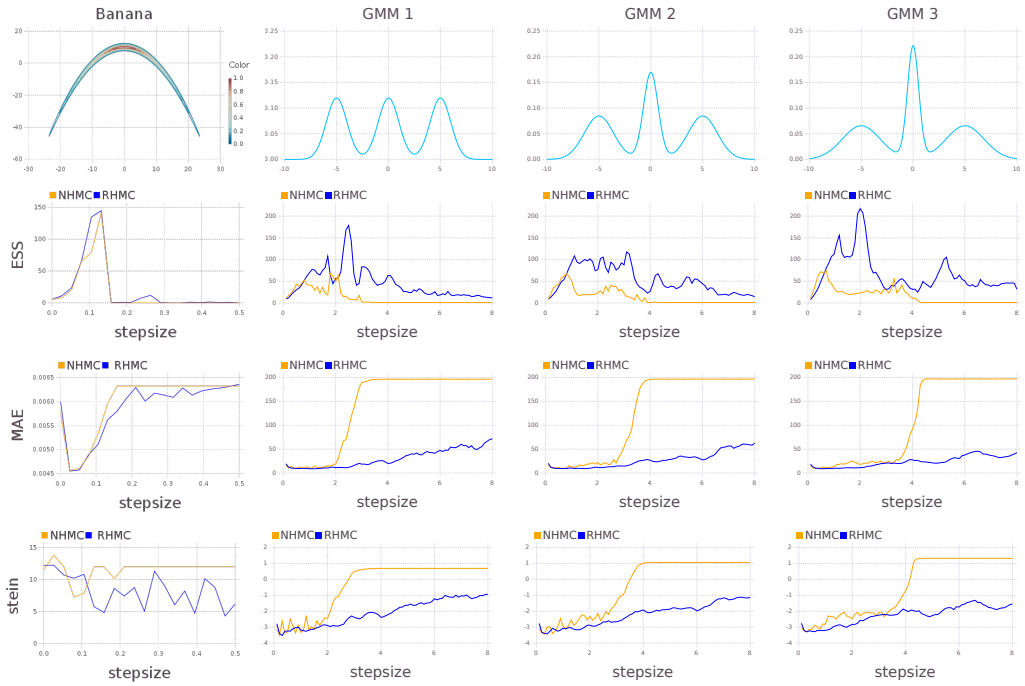

Figure 3: Left to right: Banana, GMM1, GMM2, GMM3 datasets. Top to bottom: density plot, ESS versus step size ϵ, MAE versus ϵ, log stein discrepancy versus ϵ.

Next we compare both the Newtonian and relativistic variants of HMC and SGMCMC algorithms on a simulated 3-dimensional logistic regression example with 500 observations. For the stochastic versions of the algorithms, we use mini-batches of size 100. After a burn-in period of 1000 iterations, we calculated the Stein discrepancy for different ϵ while keeping the product $\epsilon \times c$ fixed. To make a fair comparison, we used 200 samples for NHMC and RHMC and 1000 samples for the SGMCMC algorithms. From Figure 4, we see that the relativistic variants are significantly more robust than the Newtonian variants. The NHT algorithms were able to correct for stochastic gradient noise and performed better than SGHMC algorithms. Particularly, RSGNHT had lower Stein discrepancies than the other algorithms for most values of ϵ.

5.2 Neural Network

Turning to more complex models, we first considered a neural network with 50 hidden units and initialised its weights by the widely used Xavier initialisation. We used the Pima Indians dataset for binary classification (552 observations and 8 covariates) to compare the relativistic and the preconditioning approach. Indeed, these methods represent two different ways to normalise gradients so that the update sizes are reasonable for the local lengthscale of the target distribution. In particular we consider SGLD Adam, namely a preconditioned SGLD algorithm with an additional Adam-style debiasing of the preconditioner. Figure 5 compares the test-set accuracy of SGLD Adam with RSGD and RHMC, showing that the first is significantly outperformed by the relativistic algorithms. Due to Xavier initialization all of the weights are small, which causes small gradients; therefore, the injected noise becomes very large due to the rescaling by the inverse of squared root of the average gradients, which makes SGLD Adam unstable. The histograms reveal that at the first iteration the preconditioning causes the weights to become extremely large and this strongly compromises the performance of SGLD Adam, which takes a long time to recover. The relativistic framework represents therefore a much better approach to perform adaptation of the learning rates specific to each parameter.

We then apply our algorithms to the standard MNIST dataset, which consists of 28×28 handwritten digital images from 10 classes with a training set of size $60,000$ and a test set of size $10,000$. We tested our optimization algorithm on a single layer with 100 hidden units and two multi-layer neural networks with $500 * 300$ and $400 * 400$ hidden units. In

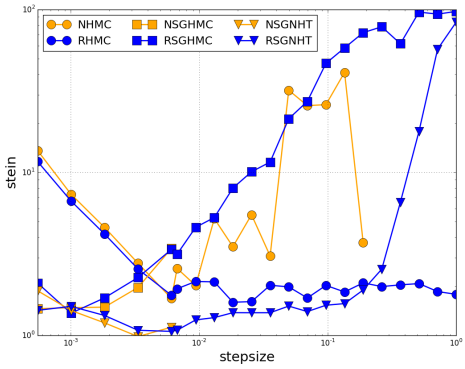

Figure 4: Stein discrepancy versus stepsize ϵ for logistic regression. NSGHMC and NSGNHT were unstable for $\epsilon > 6 \times 10^{-3}$ and thus their Stein discrepancies were not plotted.

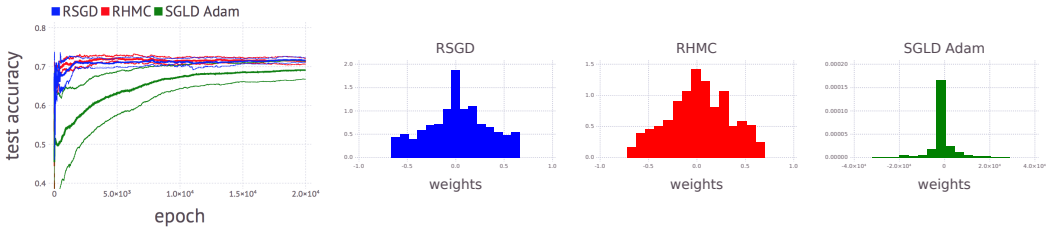

Figure 5: Comparison between RSGD, RHMC and SGLD Adam on the Pima Indians dataset using 50 hidden units. The histograms show the neural network weights at the first iteration.

Figure 6 a comparison with Adam and Santa [25] is displayed (their relation is discussed in more detail in Section 3.2). Note that, to ensure a fair comparison, we consider Santa SGD, namely a version of Santa that does not make use of symmetric splitting and simulated annealing. In other words, we adopt an Euler integration scheme for all algorithms and consider the zero-temperature limit for Santa. It can be observed that our algorithm is competitive with Adam and is able to achieve a lower error rate, particularly with the 100 hidden units architecture. Moreover, RSGD performs significantly better than Santa SGD on all the considered architectures.

6 Conclusion

Our numerical experiments demonstrate that the relativistic algorithms discussed in this paper are much more stable and robust to the choice of parameters and noise in stochastic gradients compared to the Newtonian counterparts. Moreover, we have a good understanding on how to choose the parameters c, m and ϵ. First the discretization parameter ϵ needs to be set, then we choose the maximal step $c \cdot \epsilon$ and in relation to this we choose the "cruising speed" $\frac{\bar{V}}{c}$ by picking m.

The connection of our algorithms with popular stochastic optimizers such as Adam and RMSProp is novel and gives an interesting perspective to understand them. Moreover, the relativistic stochastic gradient descent showed to be very competitive with state of the art stochastic gradient methods for fitting neural nets. We anticipate a variety of algorithms with different kinetic energies to be developed following our work.

Each of the proposed methodologies has scope for further research. The HMC version of the algorithm could be improved by employing some more advanced HMC methodology such as the NUTS version [4] and using partial moment refreshment instead of Adaptive Rejection Sampling [2]. Additionally, better numerical integration schemes could be employed. Finally,

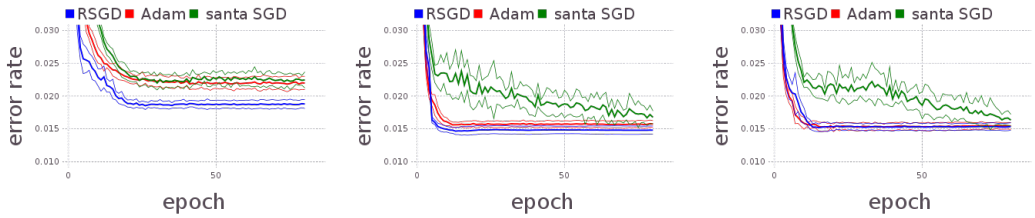

Figure 6: Comparison of the test-set error rate on the MNIST dataset. From left to right: 100 hidden units; $500 * 300$ hidden units; $400 * 400$ hidden units.

we leave an in-depth ergodic theory of the relativistic algorithms as an interesting avenue for future work.

References

[1] S. Duane, A.D. Kennedy, B.J. Pendleton, and D. Roweth. Hybrid Monte Carlo. *Physics letters B*, 195(2):216–222, 1987.

[2] R. M. Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 54:113–162, 2010.

[3] B. Carpenter, D. Lee, A. M. Brubaker, A. Riddell, A. Gelman, B. Goodrich, J. Guo, M. Hoffman, M. Betancourt, and P. Li. Stan: A probabilistic programming language. *Journal of Statistical Software*, 2015.

[4] Matthew D. Hoffman and Andrew Gelman. The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15:1593–1623, 2014.

[5] S. Livingstone, M. Betancourt, S. Byrne, and M. Girolami. On the Geometric Ergodicity of Hamiltonian Monte Carlo. *ArXiv e-print 1601.08057*, 2016.

[6] T. Xifara, C. Sherlock, S. Livingstone, S. Byrne, and M. Girolami. Langevin diffusions and the Metropolis-adjusted Langevin algorithm. *Statistics & Probability Letters*, 91(C):14–19, 2014.

[7] M. Welling and Y.W. Teh. Bayesian Learning via Stochastic Gradient Langevin Dynamics. In *ICML*, pages 681–688. Omnipress, 2011.

[8] Y. Ma, T. Chen, and E. B. Fox. A Complete Recipe for Stochastic Gradient MCMC. In *NIPS*, 2015.

[9] N. Ding, Y. Fang, R. Babbush, C. Chen, R. D. Skeel, and H. Neven. Bayesian Sampling Using Stochastic Gradient Thermostats. In *NIPS*, pages 3203–3211, 2014.

[10] T. Chen, E. Fox, and C. Guestrin. Stochastic Gradient Hamiltonian Monte Carlo. In *ICML*, pages 1683–1691, 2014.

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

[12] A. Beskos, N. Pillai, G. O. Roberts, J. M. Sanz-Serna, and A. M. Stuart. Optimal tuning of hybrid Monte Carlo algorithm. *Bernoulli*, 19:1501–1534, 2013.

[13] S. Patterson and Y. W. Teh. Stochastic gradient Riemannian Langevin dynamics on the probability simplex. In *Advances in Neural Information Processing Systems*, 2013.

[14] A. Einstein. On the Electrodynamics of Moving Bodies. *Annalen der Physik*, 17, 1905.

[15] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *ICML*, 2013.

[16] T. Tieleman and G. Hinton. Lecture 6.5-RMSProp: Divide the gradient by a running average of its recent magnitude, 2012. COURSERA: Neural Networks for Machine Learning.

[17] J. Duchi, E. Hazan, and Y. Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *JMLR*, 12:2121–2159, July 2011.

[18] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[19] J. M. Sanz-Serna and M.P. Calvo. *Numerical Hamiltonian problems*. Applied mathematics and mathematical computation. Chapman & Hall, 1994.

[20] B. Leimkuhler and X. Shang. Adaptive Thermostats for Noisy Gradient Systems. *SIAM Journal on Scientific Computing*, 38(2):A712–A736, 2016.

[21] Umut Şimşekli, Roland Badeau, A Taylan Cemgil, and Gaël Richard. Stochastic Quasi-Newton Langevin Monte Carlo. *ICML*, 2016.

[22] Chunyuan Li, Changyou Chen, David Carlson, and Lawrence Carin. Preconditioned Stochastic Gradient Langevin Dynamics for Deep Neural Networks. In *AAAI Conference on Artificial Intelligence*, 2016.

[23] L. Hasenclever, S. Webb, T. Lienart, Y. Whye Teh, S. Vollmer, B. Lakshminarayanan, and C. Blundell. Distributed Bayesian Learning with Stochastic Natural-gradient Expectation Propagation and the Posterior Server. *ArXiv e-print 1512.09327*, December 2015.

[24] X. Shang, Z. Zhu, B. Leimkuhler, and A. J. Storkey. Covariance-Controlled Adaptive Langevin Thermostat for Large-Scale Bayesian Sampling. In *NIPS*, pages 37–45, 2015.

[25] Changyou Chen, David Carlson, Zhe Gan, Chunyuan Li, and Lawrence Carin. Bridging the gap between stochastic gradient MCMC and stochastic optimization. *AISTATS*, 2016.

[26] Normand M Laurendeau. *Statistical Thermodynamics: Fundamentals and Applications*. Cambridge University Press, 2005.

[27] S. Geman and C. Hwang. Diffusions for global optimization. *SIAM Journal on Control and Optimization*, 24(5):1031–1043, 1986.

[28] C. Andrieu and A. Doucet. Simulated annealing for maximum a posteriori parameter estimation of hidden Markov models. *IEEE Transactions on Information Theory*, 46(3):994–1004, 2000.

[29] B. Leimkuhler and S. Reich. A Metropolis adjusted Nosé-Hoover thermostat. *ESAIM: Mathematical Modelling and Numerical Analysis*, 43:743–755, 7 2009.

[30] H. Haario, E. Saksman, and J. Tamminen. Adaptive Proposal Distribution for Random Walk Metropolis Algorithm. *Computational Statistics*, 14(3):375–396, 1999.

[31] J. Gorham and L. W. Mackey. Measuring Sample Quality with Stein's Method. In *NIPS*, pages 226–234, 2015.