Non-Smooth Stochastic Optimization for MCMC

Venkata Krishna Pillutla VPILLUTL@ANDREW.CMU.EDU
School of Computer Science, Carnegie Mellon University

Satwik Kottur SKOTTUR@ANDREW.CMU.EDU
Electrical and Computer Engineering Department, Carnegie Mellon University

Abstract

How do we sample efficiently from the Bayesian Lasso in a high dimensional problem with a large dataset? Hybrid Monte Carlo (HMC) has grown in popularity because it enables more efficient exploration of the state space in high-dimensional problems. Also, Stochastic Gradient-HMC has been proposed to enable application of HMC to large datasets. However, these methods apply to sampling from smooth energy functions only. We propose two ways of dealing with this: (1) SPG-HMC: Stochastic Proximal Gradient-HMC, to enable sampling from non-smooth energy functions without losing the benefits of stochasticity, and (2) Smoothing-SG-HMC. Further, we analyze its properties theoretically and empirically.

1. Introduction

Markov Chain Monte Carlo (MCMC) methods are popular to sample complex distributions, particularly for Bayesian posterior sampling. A major advantage of this method is to asymptotically guarantee samples from the true posterior distribution. However, with the advent of Big Data and ever increasing scale, datasets with billions of points have become commonplace, and drawing even a single sample can become a costly affair.

There is another class of MCMC methods that add noise to optimization rules: MALA (Robert & Casella, 2005) and Hybrid Monte Carlo (HMC) (Neal, 2010). These methods simulate physical systems with Langevin or Hamiltonian Dynamics respectively to generate samples. MH correction after each simulated step ensures that samples are from the target distribution. Important theoretical properties of the methods, known from Statistical Physics literature guarantee effective MCMC sampling.

For the 10-708 class project. Special thanks to Willie Neiswanger for all the inputs and guidance, and to Manzil Zaheer for the discussions.

Recent work (Welling & Teh, 2011b), (Chen et al., 2014) has put together these classic MCMC techniques with Stochastic Optimization techniques. These methods, being stochastic, touch only a portion of the dataset each iteration, and are super-fast. Further, such methods are more amenable to parallelization and scale more elegantly with the dimension of the data, where Gibbs sampling methods may not be tractable.

However, HMC cannot handle problems where we have to sample from non-differentiable energy functions such as the l_1 or the l_p. There sparsity promoting norms appear as Laplacian or Generalized Gaussian priors in the Bayesian context. A typical example is the Bayesian Lasso (Park & Casella, 2005). Standard Gibbs sampling approaches express the double exponential prior as a scale mixture of normals. However, even the Gibbs Sampling approach has to touch each datapoint to generate a sample. Non-differentiable energy sampling is also widely used in sparse signal and image recovery, as in (Lotfi Chaari & Batatia, 2015).

Further, there is work on Riemann Manifold Langevin and Hamiltonian dynamics (Girolami et al.), where the energy function is defined on a Riemann Manifold or when the high-dimensional target exhibits strong correlations. The method also takes advantage of the local structure as captured by the Riemannian geometry.

In this paper, we shall deal with non-smooth energy functions, such the Baysian Lasso. We explore two approaches. The proximal gradient algorithm (Eckstein & Bertsekas, 1992) is a natural first choice. For the l_1 norm which is also the most commonly encountered non-smooth term, the proximity operator has a closed from. Not surprisingly, this approach is also widely used for lasso regression, and for other variants such as the elastic net and the group lasso, among others. Further, constrained optimization problems can be posed as unconstrained optimization problems to the proximal operator. Secondly, we look at smooth relaxations of non-smooth functions as a general means of dealing with non-differentiability, that can also be used in conjunction with the proximal apporach for complicated functions.

2. Related Work

2.1. Hamiltonian Monte Carlo

Hamiltonian Monte Carlo (HMC) is a class of sampling algorithms inspired by the *Hamiltonian dynamics*. This type of dynamics was first formulated in (Alder & Wainwright, 1959) for the dynamics of the molecules and was later combined with MCMC to solve lattice field theory simulations (Duane et al., 1987a).

There is an intuitive physical interpretation for the process of sampling through HMC, due to its derivation from Hamiltonian dynamics used to model physical systems. The motion of a particle can be completely characterized by its current position (q) and its current momentum (p). If a particle of mass m, at a position q is moving with a velocity $v = \dot{q} = \frac{\partial q}{\partial t}$, its kinetic energy K can be expressed as $K(v) = \frac{1}{2}mv^2$ or equivalently $K(p) = \frac{1}{2m}p^2$. We will denote the potential energy of the particle with $U(q)$. Observe that the kinetic energy of the particle is a function of momentum alone while the potential energy is that of position. The *Hamiltonian* of the system is the quantity that equals the sum of the potential and kinetic energy, i.e., $H(q, p) = U(q) + K(p)$.

The dynamics of the particle can then be specified by a set of coupled differential equations, Eq. 1 and Eq. 2.

$$\frac{\partial H}{\partial q} = -\frac{\partial p}{\partial t} \tag{1}$$

$$\frac{\partial H}{\partial p} = v = \frac{\partial q}{\partial t} \tag{2}$$

In order to implement Hamiltonian equations, we can make use of any numerical integration. *Leap frog integrator* is preferred (Neal, 2010) as it handles the discretization errors better than simpler methods like Euler integration. The integrator equations (at time t with a step of ϵ) are shown below:

$$p_{t+\epsilon/2} = p_t - \frac{\epsilon}{2}\frac{\partial U(q_t)}{\partial q} \tag{3a}$$

$$q_{t+\epsilon} = q_t + \epsilon\frac{\partial K(p_{t+\epsilon/2})}{\partial p} \tag{3b}$$

$$= q_t + \epsilon\frac{\partial p_{t+\epsilon/2}}{\partial p} \tag{3c}$$

$$p_{t+\epsilon} = p_{t+\epsilon/2} - \frac{\epsilon}{2}\frac{\partial U(q_{t+\epsilon/2})}{\partial q} \tag{3d}$$

The above leap frog integrator is used n times to evolve the system from time t to time $t + n\epsilon$. It is interesting to note at this junction that there exists a special case of Hamiltonian dynamics, known as the *Langevin Dynamics* where the leap frog integrator is run only once, i.e. $n = 1$, to get the updated values of (q, p). Manipulating the above equations, we arrive at :

$$q_{t+\epsilon} = q_t + \epsilon p_t - \frac{\epsilon^2}{2}\frac{\partial U(q_t)}{\partial q} \tag{4}$$

Having understood the physical relevance, HMC now can be used to obtain samples from a distribution. For this, imagine the position variable q to be the desired quantity that we wish to sample (Eg. $\theta \in \mathcal{R}^D$, from a certain posterior) and introduce an auxiliary variable to represent the momentum p. The distribution from which the samples are to be taken need to be expressed as the energy function, for the Hamiltonian. Ignoring the temperature, the posterior distribution can be expressed in terms of canonical distribution, as follows:

$$P(q, p) = \frac{1}{Z}\exp(-U(q))\exp(-K(p)) \tag{5}$$

The potential energy is defined as the negative log posterior likelihood:

$$U(q) = -\log(\pi(q)L(q|\mathcal{D})) \tag{6}$$

where we impose a prior on q through $\pi(q)$ and $L(q|\mathcal{D})$ is the likelihood given the data. We use a zero-mean Gaussian distribution on the auxiliary variable p to replicate the quadratic nature of 'momentum' variable. Further, a mass matrix $\mathbf{M}$ can also be introduced which also acts as a preconditioning matrix (Welling & Teh, 2011a), (Girolami & Calderhead, 2011). Random samples for the momentum variable p are drawn at each instant and a Metropolis Hastings update is performed based on the probability distribution.

$$\theta_t = \theta_{t-1} + \frac{\epsilon_t^2}{2}\mathbf{M}^{-1}\{\sum_{i=1}^{n}\nabla_\theta \log P(x_i|\theta_{i-1}) + \nabla_\theta \log P(\theta_{t-1})\} + \epsilon_t\mathcal{M}^{-1}p_t \tag{7}$$

$$p_t \sim \mathcal{N}(0, \mathbf{M}) \tag{8}$$

The Hamiltonian has to satisfy three important properties, which are crucial for the samples to come from the true distribution. They are:

1. **Reversibility**
   Hamiltonian dynamics is reversible i.e. the transition from state (q_t, p_t) to (q_{t+T}, p_{t+T}) is reversible and can be obtained by reversing the time derivatives in the corresponding update equations. The inverse can also be obtained by negating the momentum, p_t, applying the same forward transition and negating the momentum again.

2. **Conservation of Hamiltonian**
   Ideally, the Hamiltonian must remain invariant in order to be sampling from the true distribution. Discretization errors make $H(q,p)$ only approximately invariant. Mathematically,

$$\frac{dH}{dt} = \sum_{i=1}^{d}\left[\frac{dq_i}{dt}\frac{\partial H}{\partial q_i} + \frac{dp_i}{dt}\frac{\partial H}{\partial p_i}\right] = 0 \tag{9}$$

3. **Volume Preservation**
   Hamiltonian dynamics propose a transition of the current state such that the volume in the (q,p) space is preserved. This is an important property, failing which a Jacobian has to be evaluated to compensate for the volume changes the transition causes. A way to state this properties is by equating the the divergence of vector field to be zero (Eq. 10).

$$\begin{aligned}\sum_{i=1}^{d}\left[\frac{\partial}{\partial q_i}\frac{dq_i}{dt} + \frac{\partial}{\partial p_i}\frac{dp_i}{dt}\right] &= \sum_{i=1}^{d}\left[\frac{\partial}{\partial q_i}\frac{\partial H}{\partial p_i} - \frac{\partial}{\partial p_i}\frac{\partial H}{\partial q_i}\right] \\ &= \sum_{i=1}^{d}\left[\frac{\partial^2 H}{\partial q_i \partial p_i} - \frac{\partial H^2}{\partial p_i \partial q_i}\right] \\ &= 0 \end{aligned} \tag{10}$$

2.2. Stochastic Gradient HMC

The potential energy $U(q)$ (from Eq. 6) involves calculating the log-likelihood of the data, given the current value of the parameter θ_t at time t. Therefore, update equation for p (Eq. 2) has the gradient of $U(q)$. When dealing with large dataset which is not uncommon in Bayesian inference, this often becomes a bottle-neck or perhaps even intractable as gradient is computed at every leap-frog step. In order to mitigate this huge computational burden, the family of stochastic methods introduced in (Robbins & Monro, 1951) are used. This results in the Stochastic Gradient Hamiltonian Monte Carlo (SG-HMC), where the gradient evaluated over a small batch of data is taken as an estimate of the total gradient of $U(q)$.

However, naively replacing the gradient with the stochastic variant will not yield the correct Hamiltonian dynamics (Chen et al., 2014) as one of the fundamental properties, the conservation of Hamiltonian, gets violated. The authors have added an additional 'friction' term in order to account for this disparity. Also, the step size for the update rule can be made small enough to get rid of the Metropolis Hastings rejection step at the end of leap frog updates at every iteration. The update equations transform to:

$$d\theta = \mathbf{M}^{-1}p\epsilon \tag{11}$$

$$dp = -\nabla U(\theta)\epsilon - B\mathbf{M}^{-1}p\epsilon + \mathcal{N}(0, 2B\epsilon) \tag{12}$$

Here, $B\mathbf{M}^{-1}p\epsilon$ is the friction term used to account for the lose of conservation of Hamiltonian.

2.3. Proximal Gradients and Proximal Gradient HMC

The proximal gradient method is based on the proximal operator (Eckstein & Bertsekas, 1992), $\mathbf{prox} : \mathbb{R}^d \to \mathbb{R}^d$ defined as:

$$\mathbf{prox}_{\lambda f}(x) = \arg\min_z \left[f(z) + \frac{1}{2\lambda}\|x - z\|_2^2\right] \tag{13}$$

When λ is zero, $\mathbf{prox}_{\lambda f}$ returns the same point itself. When λ is infinity, $\mathbf{prox}_{\lambda f}$ returns the global minimizer of the function. It can be viewed as some returning a nearby point that reduces the function value. This compares directly to the gradient, which gives the direction of maximum increase of the function. This leads immediately to the *Proximal Point Algorithm*, which performs the following iterations to minimize f:

$$x^{(t+1)} = \mathbf{prox}_{\lambda f}(x^{(t)})$$

Under the mind assumption that f has a minimizer, this algorithm converges because of the fixed point property of $\mathbf{prox}$, which states that x_* minimizes f iff $\mathbf{prox}_{\lambda f}(x_*) = x_*$ (Parikh & Boyd, 2014).

For convex, differentiable f, the Proximal Point Algorithm is very similar to gradient descent. In fact, both of them are related to gradient flow. Gradient flow is the differential equation $\frac{d}{dt}x(t) = \nabla f(x(t))$, the equilibrium point of which is the minimizer of f. Gradient descent is the explicit Euler discretization of this continuous time process, where as, the proximal point algorithm is the implicit Euler discretization, which is know to have better stability properties but computationally more expensive to solve an implicit equation. See (Parikh & Boyd, 2014) for more details.

Consider the minimization problem of the form:

$$\min_x f(x) + h(x)$$

where f, h are closed convex functions and f is differentiable. h can be an extended value representation to encode constraints. The *Proximal Gradient Algorithm* performs iterations of the form:

$$x^{(t+1)} = \mathbf{prox}_{\lambda^t h}(x^{(t)} - \lambda^t \nabla f(x^{(t)}))$$

One may wonder how solving an optimization problem inside another will make it any easier to solve the original problem. The real power to proximal gradient algorithm lies in that fact that the $\mathbf{prox}$ operator has a simple, perhaps closed form for some functions. It may be seen that

for the l_1 norm, the proximal operation is soft-threshold the input at level λ,

$$(\mathbf{prox}_{\lambda\|\cdot\|_1}(x))_i = \begin{cases} x_i - \lambda & x_i > \lambda \\ x_i + \lambda & x_i < -\lambda \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

This makes the proximal gradient algorithm popular for solving the lasso or its numerous extensions (Parikh & Boyd, 2014). Further, solving the implicit Euler update, as discussed earlier, now comes free of cost.

Constrained optimization problems may be expressed using the indicator function for a set C as $h(x) = I_C(x)$ which is 0 if $x \in C$ and $+\infty$ otherwise. It can also be seen that $\mathbf{prox}$ is now the Euclidean projection:

$$\mathbf{prox}_{I_C}(x) = \arg\min_{z \in C} \|x - z\|^2 \tag{15}$$

PROXIMAL LANGEVIN MONTE CARLO

Authors of (Pereyra, 2013) introduced Proximal Langevin Dynamics: application of the proximal operator to Langevin Dynamics. They talk about two techniques: P-ULA, and Proximal MALA, with and without the MH correction step, respectively. Essentially, they use the relation: $\mathbf{prox}_{\lambda f}(x) \approx x - \lambda \nabla f(x)$, to obtain (to sample from π:

$$\theta^* = (1 - \frac{\epsilon}{2\lambda})\theta + \frac{\epsilon}{2\lambda}\mathbf{prox}_{\lambda \log \pi}(\theta) + \eta \tag{16}$$

where $\eta \sim N(0, \epsilon I)$

2.4. Bayesian Lasso

Introduced in (Park & Casella, 2005), the Bayesian Lasso considers the sparcity inducing l_1 penalty on the parameters as a double exponential prior on the parameters.

$$\pi(\theta_j|\sigma^2) = \frac{\lambda}{2\sigma}\exp(-\lambda|\theta_j|/\sigma) \tag{17}$$

Inference is done using Gibbs sampling by using the following trick:

$$\frac{a}{2}\exp(-az) = \int_0^\infty \frac{1}{\sqrt{2\pi s}}\exp(-z^2/2s)\frac{a^2}{2}\exp(-a^2 s/2)ds \tag{18}$$

Further, Bayesian credible intervals are derived. We aim to do the same, but without visiting the entire dataset for each sample.

3. Method

3.1. Proximal HMC

We propose Stochastic Proximal Gradient HMC for sampling from non-smooth densities without ever touching the whole dataset in each iteration. But first, let us derive HMC using $\mathbf{prox}$. For this, we define the gradient mapping. This quantity has been used implicitly in several papers in the past. See, for instance, (Parikh & Boyd, 2014).

Definition 1. *For a function h and a step size η, define the gradient mapping of f to be:*

$$g_\eta(h)(x) = (x - \mathbf{prox}_{\eta h}(x))/\eta \tag{19}$$

Further, if f is differentiable,

$$g_\eta(f + h)(x) = (x - \mathbf{prox}_{\eta h}(x - \eta\nabla f(x)))/\eta \tag{20}$$

Note, for a differentiable f that $g_\eta(h)(x) \propto \nabla f(\mathbf{prox}_h(x))$, by definition. Further, for the proximal gradient algorithm, η is exactly the same as the step-length for the first step. From (Parikh & Boyd, 2014), under appropriate conditions, $\mathbf{prox}_{\lambda f}(x) \approx x - \lambda\nabla f(x)$. If h is differentiable, $g_\eta(f + h)(x) \approx \nabla f(x) + \nabla h(x)$.

First, for non-smooth Hamiltonian Dynamics, since the differential equations change as:

$$\frac{\partial p}{\partial t} \in -\partial_q H \tag{21}$$

$$\frac{\partial q}{\partial t} = \frac{\partial H}{\partial p} = v \tag{22}$$

We propose to solve 21 with the proximal gradient method. Note that the proximal gradient method is equivalent to taking an explicit Euler step with the smooth part of the energy and an implicit Euler step with respect to the non-smooth term. The resulting leap-frog update equations now look like:

$$p_{t+\epsilon/2} = p_t - \frac{\epsilon}{2}g_{\epsilon/2}(U)(q_t) \tag{23a}$$

$$q_{t+\epsilon} = q_t + \epsilon\frac{\partial K(p_{t+\epsilon/2})}{\partial p} \tag{23b}$$

$$= q_t + \epsilon p_{t+\epsilon/2} \tag{23c}$$

$$p_{t+\epsilon} = p_{t+\epsilon/2} - \frac{\epsilon}{2}g_{\epsilon/2}(U)(q_t) \tag{23d}$$

To make this stochastic, the gradient of the likelihood must be stochastic. But as demonstrated in (Chen et al., 2014), a noise is added implicitly by considering a stochastic gradient. From the central limit theorem, the noise is approximately Gaussian. In order to make sure that the noise doesn't change the estimate by much, a "friction" term is required. We use the same trick to make the stochastic variant of Proximal HMC to work.

Algorithm 1 Stochastic Proximal HMC

input starting position $\theta^{(0)}$ and step size ϵ
  for $t = 1, 2, ...$ **do**
    Resample Momentum
    $p^{(t)} \sim \mathcal{N}(0, M)$
    Leap-frog Steps
    $(\theta_0, p_0) \sim (\theta^{(t)}, r^{(t)})$
    $Z \sim \mathcal{N}(0, 2(C - \hat{B})\epsilon)$
    $p_0 \leftarrow p_0 - \frac{\epsilon}{2}\tilde{g}_{\epsilon/2}(U)(\theta_0) - \epsilon C M^{-1}\theta_0 + Z$
    for $i = 1..m$ **do**
      $\theta_i \leftarrow \theta_{i-1} + \epsilon M^{-1} p_{i-1}$
      $Z \sim \mathcal{N}(0, 2(C - \hat{B})\epsilon/2)$
      $p_i \leftarrow p_{i-1} - \epsilon\tilde{g}_\epsilon(U)(\theta_i) - \frac{\epsilon}{2} C M^{-1}\theta_i + Z$
    end for
    No MH Correction
    $\theta^{(t+1)} \leftarrow \theta_m$
  end for

3.2. Smoothing HMC

We look at a different approach of dealing with non-smooth functions by approximating them with smooth functions. Further, we can make them arbitrarily close to each other, as we shall see. We consider this for the running example of the Bayesian Lasso, and similar procedures can be followed for any other convex non-smooth functions. We remark that evaluation of the estimator for other arbitrary norms may not be as simple.

First, start with the variational form of the l_1 norm.

$$R(x) = \lambda\|x\|_1 = \max_{\|y\|_\infty \le 1} \lambda y^T x \tag{24}$$

Now, we regularize this maximization problem to get a smooth approximation using a parameter $\mu > 0$:

$$\tilde{R}_\mu(x) = \max_{\|y\|_\infty \le 1} \lambda y^T x - \frac{\mu}{2}\|y\|_2^2 \tag{25}$$

Note that if $\mu = 0$, $\tilde{R}_\mu = R$. We shall drop the subscript μ when it is not relevant. This approximation is smooth.

Proposition 1. *$\tilde{R}(x)$ is a smooth function of x and its gradient is*

$$\nabla\tilde{R}(x) = \lambda y^*(x) \tag{26}$$

where

$$y^*(x) = \arg\max_{-1 \le y \le 1}(\lambda y^T x - \frac{\mu}{2}\|y\|_2^2) \tag{27}$$

Proof. Note that the l_∞ constraint is a box constraint $-1 \le y \le 1$. Fix a value of x. Equation 27 gives us the maximizer(s) $y^*(x)$ for this value of x. From first principles, we can see that the set of subgradients, $\partial_x\tilde{R}(x) = \{\lambda y | y \in \arg\max_{-1\le y\le 1}(\lambda y^T x - \frac{\mu}{2}\|y\|_2^2)\}$.

From strong concavity of the maximization problem, $y^*(x)$ is unique for each x, and so, the set of subgradients is a singleton set. In other words, $\tilde{R}$ is differentiable with $\nabla\tilde{R}(x) = \lambda y^*(x)$. □

Note also that computation of y^* is $\mathcal{O}(d)$ and can be solved in closed form for each dimension independently as $(y^*(x))_i = \text{sign}(x_i).\min\{\frac{\lambda|x_i|}{\mu}, 1\}$.

Now we shall see that the approximation can be uniformly made as sharp as desired by controlling μ. The trade-off is between approximation quality and smoothness.

Proposition 2. $\forall x : |\tilde{R}_\mu(x) - R(x)| \le \mu.d$

Proof. WLOG, suppose $\lambda = 1$. For other λ, we can scale μ appropriately. Let $a \wedge b = \min\{a, b\}$. First, notice that $\tilde{R}_\mu(x) = \sum_i[(\frac{x_i^2}{\mu} \wedge |x_i|) - (\frac{x_i^2}{2\mu} \wedge \frac{\mu}{2})]$. So, we have,

$$\begin{aligned}|\tilde{R}_\mu(x) - R(x)| &= |\sum_{i=1}^d[(\frac{x_i^2}{\mu} - |x_i| \wedge 0) - (\frac{x_i^2}{2\mu} \wedge \frac{\mu}{2})]| \\ &\le \sum_{i=1}^d |(\frac{x_i^2}{\mu} - |x_i|) \wedge 0| + |\frac{x_i^2}{2\mu} \wedge \frac{\mu}{2}|\end{aligned}$$

The second term is at most $\mu/2$. The first term is non-zero only when $\frac{x_i^2}{\mu} - |x_i|$ is negative. It is a quadratic in x and the smallest it can be for any x is $-\mu/2$. Hence, even the first term is at most $\mu/2$. And there are d dimensions. Putting these together, we have,

$$|\tilde{R}_\mu(x) - R(x)| \le \mu.d \tag{28}$$

□

For more complicated loss functions, a combination of both of these methods can be used, as in (Chen et al., 2012).

4. Theoretical Analysis

In this section, we show that the without any MH-correction, proximal HMC is ergodic and approximates the true density under certain conditions.

First, we shall look at ergodicity.

Theorem 1. *Let p, the target density be one-dimensional, and let $U = -\log p$, the potential energy be continuously differentiable. For some $d \in [0, 1)$, define:*

$$S_d^+ := \lim_{\theta\to\infty} \theta^{-d}(\mathbf{prox}_{\epsilon U/2}(\theta) - \theta)$$
$$S_d^- := \lim_{\theta\to-\infty} |\theta|^{-d}(\mathbf{prox}_{\epsilon U/2}(\theta) - \theta)$$

If the step-length, ϵ is sufficiently small, and the number of leap step are not too large, proximal HMC is ergodic if:

- *for some $0 \leq d \leq 1$, both S_d^+ and S_d^- exist; and*
- *if $d = 1$, then $(S_d^+ - 1)(1 - S_d^-) < 1$*

Proof. In one iteration of prox-HMC, let the initial value of θ be θ_0, and the next sample be θ_L, after L leap-frop steps passing through intermediate values $\theta_1, ...\theta_{L-1}$. Note that the HMC update is equivalent to $\theta_i = \theta_{i-1} + \epsilon Z - (\epsilon^2/2)\nabla U(\theta_0)$, where Z is a standard normal. By Taylor's Theorem, we have,

$$\begin{aligned}\nabla U(\theta_i) =& \nabla U(\theta_{i-1} + \nabla^2 U(\theta_{i-1})(\theta_i - \theta_{i-1}) + \mathcal{O}(\epsilon^2) \\ \nabla U(\theta_t) =& \nabla U(\theta_0) + \sum_{k=1}^{t} \nabla^2 U(\theta_k)(\theta_k - \theta_{k-1}) + \mathcal{O}(t.\epsilon^2) \\ =& \nabla U(\theta_0) + \mathcal{O}(t\epsilon)\end{aligned}$$

From this,

$$\begin{aligned}\theta_t &= \theta_0 + t\epsilon Z - \frac{t\epsilon^2}{2}\nabla U(\theta_0) - \sum_{i=1}^{t-1}(t-i)\epsilon^2 \nabla U(\theta_0) \\ &= \theta_0 + t\epsilon Z - \frac{t^2\epsilon^2}{2}\nabla U(\theta_0) + \mathcal{O}(t^3\epsilon^3)\end{aligned}$$

Provided $L\epsilon$ is small enough, Theorem 3.1 of (Pereyra, 2013) can directly be applied to get the required result. □

The intuition behind this result as explained in (Pereyra, 2013) is that $\lim_{\theta \to \pm\infty} \frac{d}{d\theta}\tilde{U}(\theta) = S_d^{\pm}\theta^d + o(\theta^d)$ where $\tilde{U}$ is the Moreau envelope (Parikh & Boyd, 2014) of U.

Now we shall look at the convergence to the desired stationary density.

Theorem 2. *If $p \in \mathcal{C}^2$ (the set of function with continuous second derivatives). If L is small, as the step-size $\epsilon \to 0$, prox-HMC converges in mean square to continuous Hamiltonian Dynamics with stationary measure p.*

Proof. Follows from the previous proof and Theorem 3.2 in (Pereyra, 2013). □

A caveat on the usefulness of these results are that assumptions are made on the differentiability of the function, which do not hold for the application area of this algorithm. Another concern is the restriction that L should not be too large. Favorable properties of HMC over LMC such as being able to reach high probability regions far away are largely due to moderate number of leap steps, L. Hence, we ought to take these results with a pinch of salt.

5. Experiments

To validate the proposed method for dealing with non-smooth energy functions, a series of experiments are conducted with both synthetically generated data as well as real world datasets.

5.1. Sampling in a single dimension

First, we setup a non-smooth, single dimension probability distribution from which samples are to be drawn. For this, we consider $P(\theta) \propto \exp\left(-\lambda|\theta| - \theta^2\right)$. The presence of $|\theta|$ makes this distribution non-smooth with a sharp peak at 0 (as seen in Figure 1). Samples are considering by evolving the modified stochastic HMC dynamics as described in the previous sections, using the proximal gradient approach. We compare our approach with the baseline of a standard HMC implementation both with and without the MH correction step[1]. We also plot the results from the Naive Stochastic gradient (without the friction term), once again both with and without MH step. Figure 1 shows the empirical distribution of the samples from these algorithms.

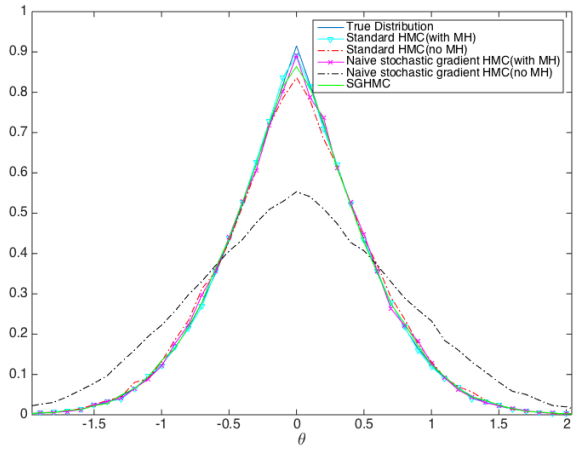

Figure 1. Empirical distribution using various sampling algorithms for a non-smooth energy function $U(\theta) = -\lambda|\theta| - \theta^2$. All these methods use smoothing approach. Here, $\lambda = 1$ is used, smoothing parameter $\mu = 10^{-5}$ is used.

We see that the standard HMC gives close results, even without the MH step. Clearly, as the theory suggests, Naive stochastic gradient without MH correction diverges from the true distribution. Although the Naive Stochastic gradient with MH step seems to approximate the distribution well, it results in a lot of rejections which is often undesirable. SGHMC, without the costly MH step also results in a good approximation for the desired distribution. This validates our proximal gradient approach to deal with non-

[1]Code courtesy Chen et al. (2014)

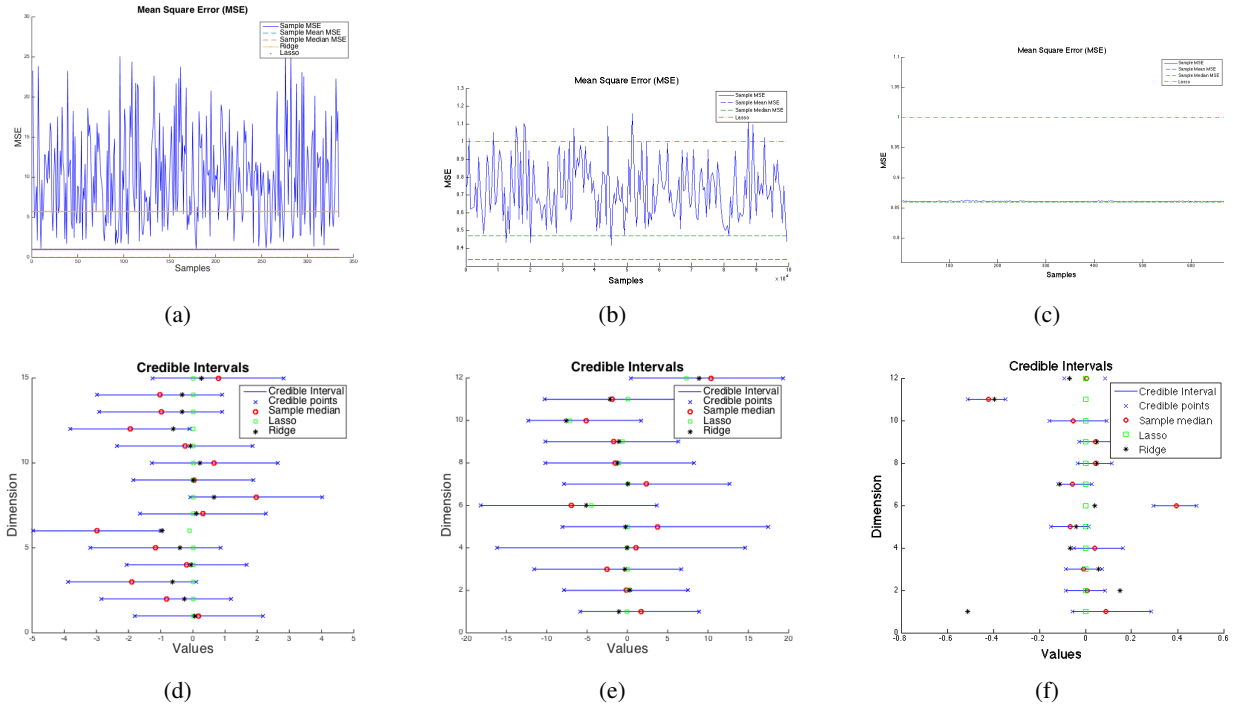

Figure 2. Results for Stochastic HMC with non-smooth energy functions. First row (a, b, c): Mean squared error (MSE) for the regression samples generated. MSE is normalized to MSE of Lasso. Ridge regression results are also plotted in case they are reasonable. The MSE for the sample converges indicating the convergences of the Markov chain. Second row (d, e, f): Bayesian 95% credible intervals for the samples. Most of the intervals contain zero which is desirable. Even the lasso estimates are within the intervals. Column-wise results correspond to synthetic, CPUSmall and Year prediction datasets respectively.

smooth functions for sampling algorithms and SG-HMC, in particular.

5.2. Regression

We also test our system on the Bayesian Lasso setup which enforces sparsity through l_1 regularization that makes the overall energy non-smooth. Specifically, we aim to obtain samples for the regression parameters with the l_1 penalty. The Lasso implementation of MATLAB is taken as the baseline for this experiment. We use the Mean Squared Error (MSE) on the test set as a measure to assess the regression co-efficients. We also consider the ridge regression implementation of MATLAB for comparison.

The following datasets are used:

1. SYNTHETIC DATASET
   We first consider a synthetic dataset $\{X_i, y_i\}_{i=1}^N$ obtained through the following generative process:

$$\text{Given } X_1, X_2, \ldots, X_n \text{ and } \beta$$
$$y_i|\beta \sim \beta^T X_i + \epsilon_i$$
$$\epsilon_i \sim \mathcal{N}(0, \sigma^2)$$

   We generate a total of 10000 such data points of dimension 300, and use 80% of it as training instances and the remaining for testing.

2. CPUSMALL DATASET[2]
   CPUSmall dataset has around 8200 data points of dimension 12. Around 80% of it is used as training while the remaining is split equally into testing and validation data.

3. YEAR PREDICTION MSD[3]
   It has around 460,000 data points of dimension 90. However, the data points are given in two separate files containing the training and test sets.

We compare the two proposed methods – proximal gradients and smoothed l_1– for dealing with non-smooth l_1 penalty. For smoothed l_1, try out $\mu \in \{10^{-6}, 10{-}5, ..., 10^3\}$, and use the best value for each dataset (cross-validation). As both the methods were observed to give very identical results (same till two decimal

[2] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/regression/cpusmall

[3] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/regression/YearPredictionMSD.bz2

places of MSE), we show only the MSE on testing dataset for the samples generated through smoothed l_1.

ESTIMATING λ

Observe that we have a smoothing parameter λ in equation 24. An **empirical Bayes** estimate is obtained using a different validation data (obtained from testing set) by running EM algorithm.

E-step: Samples for β are generated by running the SG HMC, using the previous value of λ.

M-step: The M-step can be computed in closed form, as in (Park & Casella, 2005):

$$\lambda = \frac{d}{\mathbb{E}\left[\frac{\|\beta\|_1}{\sqrt{\sigma^2}}\right]} \tag{29}$$

Here, d is the number of dimensions for the current problem setting. Since we cannot compute the expectation in closed form, we use instead the empirical expectation computed from the samples generated in E-step.

FIGURES

Figure 2(a) shows the MSE on testing dataset. Clearly, the error converges indicating the convergence of the Markov chain. As a point estimate, we consider the dimension-wise median for the samples. We observe that this point estimate performs better than Lasso and ridge regression. To further investigate the sparsity in the samples, we compute 95% credible intervals for all the dimensions. The presence of zero in the credible interval is desirable in the Bayesian setting. Figure 2(d) shows the credible intervals along with the median of samples. Clearly, most of the credible intervals contain the zero. The Lasso estimators are also within these intervals.

On the CPUSmall data set, ridge regression has extremely poor performance and hence we avoid plotting it. As seen from Figure 2(b), the samples perform better than Lasso while the median gives a way better MSE for this particular dataset. Next, we plot the credible intervals as before in Figure 2(e). The intervals are 95% Bayesian confidence intervals and contain both zero and the lasso estimates, which is desirable for sparsity.

Finally, Figure 2(c) shows the MSE plotted for the large dataset of year prediction. Stochastic HMC results in an immense speed up compared to full gradient HMC as only small portion of the dataset is used to estimate gradient at every step. As before, ridge regression results in a high MSE (about 6 times compared to Lasso) and is not shown. The credible intervals for year prediction are plotted in Figure 2(f) indicating the desired property of containing zero in the intervals.

6. Conclusion

In this work, we have extended Stochastic HMC to deal with non-smooth energies by proposing two approaches. The first approach makes use of the idea of proximal gradients. We prove some theoretical properties for this. On the other hand, the second approach deals with non-smooth energies by smoothing it out. The technique has been demonstrated for the Bayesian Lasso, but can be applied much more generally. We show experiments on large synthetic and real datasets, demonstrating the effectiveness of our proposed methods.

Open Questions: The absence of an MH-step must mean that the step length has to be super small. The first straightforward extension would be to include an approximate MH-step based on a subset of the data. The method proposed in (Balan et al., 2013) is slow to the point where it is not practical and alternatives have to be explored. Further, better theoretical results are needed. To distribute this method, asyncrhonous stochastic gradient methods can be used with appropriate modification for the prox operator. We intend to pursue this next as it looks promising.

References

Alder, B. J. and Wainwright, T. E. Studies in molecular dynamics. i. general method. *J. Chem. Phys.*, 31:459, 1959.

Balan, Anoop Korattikara, Chen, Yutian, and Welling, Max. Austerity in MCMC land: Cutting the metropolis-hastings budget. *CoRR*, abs/1304.5299, 2013. URL http://arxiv.org/abs/1304.5299.

Bottou, Lon and Bousquet, Olivier. The tradeoffs of large scale learning. In *IN: ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 20*, pp. 161–168, 2008.

Chen, T., Fox, E.B., and Guestrin, C. Stochastic gradient Hamiltonian Monte Carlo. In *Proc. International Conference on Machine Learning*, June 2014.

Chen, Xi, Lin, Qihang, Kim, Seyoung, Carbonell, Jaime G., and Xing, Eric P. Smoothing proximal gradient method for general structured sparse regression. *Ann. Appl. Stat.*, 6(2):719–752, 06 2012. doi: 10.1214/11-AOAS514. URL http://dx.doi.org/10.1214/11-AOAS514.

Duane, Simon, Kennedy, A. D., Pendleton, Brian J., and Roweth, Duncan. Hybrid monte carlo. *Physics Letters B*, 195(2):216 – 222, 1987a. ISSN 0370-2693. doi: DOI:10.1016/0370-2693(87)91197-X. URL http://www.sciencedirect.com/science/article/B6TVN-46YSWPH-2XF/2/0f89cdc6cf214a2169b03df7414f3df4.

Duane, Simon, Kennedy, A.D., Pendleton, Brian J., and Roweth, Duncan. Hybrid monte carlo. *Physics Letters B*, 195(2):216 – 222, 1987b.

Eckstein, Jonathan and Bertsekas, Dimitri P. On the douglas-rachford splitting method and the proximal point algorithm for maximal monotone operators. *Math. Program.*, 55(3):293–318, June 1992. ISSN 0025-5610.

Girolami, Mark and Calderhead, Ben. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society Series B*, 73(2):123–214, 03 2011. URL http://ideas.repec.org/a/bla/jorssb/v73y2011i2p123-214.html.

Girolami, Mark, Calderhead, Ben, and Chin, Siu A. Riemann manifold langevin and hamiltonian monte carlo methods. *J. of the Royal Statistical Society, Series B (Methodological*.

Li, Shuying, Pearl, Dennis K., and Doss, Hani. Phylogenetic tree construction using markov chain monte carlo, 1999.

Lotfi Chaari, Jean-Yves Tourneret, Caroline Chaux and Batatia, Hadj. A hamiltonian monte carlo method for non-smooth energy sampling. 2015.

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

Parikh, Neal and Boyd, Stephen. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3), 2014.

Park, Trevor and Casella, George. The bayesian lasso. Technical report, 2005.

Patterson, Sam and Teh, Yee Whye. Stochastic gradient riemannian langevin dynamics on the probability simplex. In *Advances in Neural Information Processing Systems 26*. Curran Associates, Inc., 2013.

Pereyra, Marcelo. Proximal markov chain monte carlo algorithms. *arXiv preprint arXiv:1306.0187*, 2013.

Robbins, Herbert and Monro, Sutton. A stochastic approximation method. *Ann. Math. Statist.*, 22(3):400–407, 09 1951. doi: 10.1214/aoms/1177729586.

Robert, Christian P. and Casella, George. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005. ISBN 0387212396.

Tibshirani, Robert. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

Welling, Max and Teh, Yee Whye. Bayesian learning via stochastic gradient langevin dynamics. In Getoor, Lise and Scheffer, Tobias (eds.), *ICML*, pp. 681–688. Omnipress, 2011a. URL http://dblp.uni-trier.de/db/conf/icml/icml2011.html#WellingT11.

Welling, Max and Teh, Yee Whye. Bayesian learning via stochastic gradient langevin dynamics. In Getoor, Lise and Scheffer, Tobias (eds.), *ICML*, pp. 681–688. Omnipress, 2011b.