The Fundamental Incompatibility of Scalable Hamiltonian Monte Carlo and Naive Data Subsampling

Michael Betancourt BETANALPHA@GMAIL.COM
Department of Statistics, University of Warwick, Coventry, UK CV4 7AL

Abstract

Leveraging the coherent exploration of Hamiltonian flow, Hamiltonian Monte Carlo produces computationally efficient Monte Carlo estimators, even with respect to complex and high-dimensional target distributions. When confronted with data-intensive applications, however, the algorithm may be too expensive to implement, leaving us to consider the utility of approximations such as data subsampling. In this paper I demonstrate how data subsampling fundamentally compromises the scalability of Hamiltonian Monte Carlo.

With the preponderance of applications featuring enormous data sets, methods of inference requiring only subsamples of data are becoming more and more appealing. Subsampled Markov Chain Monte Carlo algorithms, (Neiswanger et al., 2013; Welling & Teh, 2011), are particularly desired for their potential applicability to most statistical models. Unfortunately, careful analysis of these algorithms reveals unavoidable biases unless the data are *tall*, or highly redundant (Bardenet et al., 2014; Teh et al., 2014; Vollmer et al., 2015). Because redundancy can be defined only relative to a given model, the utility of these subsampled algorithms is then a consequence of not only the desired accuracy and also the particular model and data under consideration, severely restricting practicality.

Recently (Chen et al., 2014) considered subsampling within Hamiltonian Monte Carlo (Duane et al., 1987; Neal, 2011; Betancourt et al., 2014b) and demonstrated that the biases induced by naive subsampling lead to unacceptably large biases. Ultimately the authors rectified this bias by sacrificing the coherent exploration of Hamiltonian flow for a diffusive correction, fundamentally compromising the scalability of the algorithm with respect to the complexity of the target distribution. An algorithm scalable with respect to both the size of the data and the complexity of the target distribution would have to maintain the coherent exploration of Hamiltonian flow while subsampling and, unfortunately, these objectives are mutually exclusive in general.

In this paper I review the elements of Hamiltonian Monte Carlo critical to its robust and scalable performance in practice and demonstrate how different subsampling strategies all compromise those properties and consequently induce poor performance.

1. Hamiltonian Monte Carlo in Theory

Hamiltonian Monte Carlo utilizes deterministic, measure-preserving maps to generate efficient Markov transitions (Betancourt et al., 2014b). Formally, we begin by complementing a target distribution,

$$\pi \propto \exp[-V(q)] \, \mathrm{d}^n q,$$

with a conditional distribution over auxiliary *momenta* parameters,

$$\pi_q \propto \exp[-T(p, q)] \, \mathrm{d}^n p.$$

Together these define a joint distribution,

$$\begin{aligned} \varpi_H &\propto \exp[-\left(T(q,p) + V(q)\right)] \, \mathrm{d}^n q \, \mathrm{d}^n p \\ &\propto \exp[-H(q,p)] \, \mathrm{d}^n q \, \mathrm{d}^n p, \end{aligned}$$

and a *Hamiltonian system* corresponding to the *Hamiltonian*, $H(q,p)$. We refer to $T(q,p)$ and $V(q)$ as the *kinetic energy* and *potential energy*, respectively.

The Hamiltonian immediately defines a *Hamiltonian vector field*,

$$\vec{H} = \frac{\partial H}{\partial p} \frac{\partial}{\partial q} - \frac{\partial H}{\partial q} \frac{\partial}{\partial p},$$

and an application of the exponential map yields a *Hamiltonian flow* on the joint space, $\phi^H_\tau = e^{\tau \vec{H}}$ (Lee, 2013), which exactly preserves the joint distribution under a pullback,

$$\left(\phi^H_t \right)_* \pi_H = \pi_H.$$

Proceedings of the 32^{nd} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37. Copyright 2015 by the author(s).

Consequently, we can compose a Markov chain by sampling the auxiliary momenta,

$$q \to (q, p),\, p \sim \pi_q,$$

applying the Hamiltonian flow,

$$(q, p) \to \phi^H_t(q, p)$$

and then projecting back down to the target space,

$$(q, p) \to q.$$

By construction, the trajectories generated by the Hamiltonian flow explore the level sets of the Hamiltonian function. Because these level sets can also span large volumes of the joint space, sufficiently-long trajectories can yield transitions far away from the initial state of the Markov chain, drastically reducing autocorrelations and producing computationally efficient Monte Carlo estimators.

When the kinetic energy does not depend on position we say that the Hamiltonian is *separable*, $H(q, p) = T(p) + V(q)$, and the Hamiltonian vector field decouples into a kinetic vector field, $\vec{T}$ and potential vector field, $\vec{V}$,

$$\begin{aligned} \vec{H} &= \frac{\partial H}{\partial p}\frac{\partial}{\partial q} - \frac{\partial H}{\partial q}\frac{\partial}{\partial p} \\ &= \frac{\partial T}{\partial p}\frac{\partial}{\partial q} - \frac{\partial V}{\partial q}\frac{\partial}{\partial p} \\ &\equiv \quad \vec{T} \quad + \quad \vec{V} \quad . \end{aligned}$$

In this paper I consider only separable Hamiltonians, although the conclusions also carry over to the non-seperable Hamiltonians, for example those arising in Riemannian Hamiltonian Monte Carlo (Girolami & Calderhead, 2011).

2. Hamiltonian Monte Carlo in Practice

The biggest challenge of implementing Hamiltonian Monte Carlo is that the exact Hamiltonian flow is rarely calculable in practice and we must instead resort to approximate integration. *Symplectic integrators*, which yield numerical trajectories that closely track the true trajectories, are of particular importance to any high-performance implementation.

An especially transparent strategy for constructing symplectic integrators is to split the Hamiltonian into terms with soluble flows which can then be composed together (Leimkuhler & Reich, 2004; Hairer et al., 2006). For example, consider the symmetric *Strang* splitting,

$$\phi^V_{\frac{\epsilon}{2}} \circ \phi^T_{\epsilon} \circ \phi^V_{\frac{\epsilon}{2}} = e^{\frac{\epsilon}{2}\vec{V}} \circ e^{\epsilon\vec{T}} \circ e^{\frac{\epsilon}{2}\vec{V}},$$

where ϵ is a small interval of time known as the *step size*. Appealing to the Baker-Campbell-Hausdorff formula, this symmetric composition yields

$$\begin{aligned} &\phi^V_{\frac{\epsilon}{2}} \circ \phi^T_{\epsilon} \circ \phi^V_{\frac{\epsilon}{2}} \\ &\quad = e^{\frac{\epsilon}{2}\vec{V}} \circ e^{\epsilon\vec{T}} \circ e^{\frac{\epsilon}{2}\vec{V}} \\ &\quad = e^{\frac{\epsilon}{2}\vec{V}} \circ \exp\left(\epsilon\vec{T} + \frac{\epsilon}{2}\vec{V} + \frac{\epsilon^2}{4}\left[\vec{T}, \vec{V}\right]\right) + \mathcal{O}\big(\epsilon^3\big) \\ &\quad = \exp\left(\frac{\epsilon}{2}\vec{V} + \epsilon\vec{T} + \frac{\epsilon}{2}\vec{V} + \frac{\epsilon^2}{4}\left[\vec{T}, \vec{V}\right]\right. \\ &\qquad\qquad \left. + \frac{1}{2}\left[\frac{\epsilon}{2}\vec{V}, \epsilon\vec{T} + \frac{\epsilon}{2}\vec{V} + \frac{\epsilon^2}{4}\left[\vec{T}, \vec{V}\right]\right]\right) \\ &\qquad + \mathcal{O}\big(\epsilon^3\big) \\ &\quad = \exp\left(\epsilon\vec{H} + \frac{\epsilon^2}{4}\left[\vec{T}, \vec{V}\right] + \frac{\epsilon^2}{4}\left[\vec{V}, \vec{T}\right] + \frac{\epsilon^2}{8}\left[\vec{V}, \vec{V}\right]\right) \\ &\qquad + \mathcal{O}\big(\epsilon^3\big) \\ &\quad = e^{\epsilon\vec{H}} + \mathcal{O}\big(\epsilon^3\big)\,. \end{aligned}$$

Composing this symmetric composition with itself $L = \tau/\epsilon$ times results in a symplectic integrator accurate to second-order in the step size for any finite integration time, τ,

$$\begin{aligned} \phi^{\widetilde{H}}_{\epsilon,\tau} &\equiv \left(\phi^V_{\frac{\epsilon}{2}} \circ \phi^T_{\epsilon} \circ \phi^V_{\frac{\epsilon}{2}}\right)^L \\ &= \left(e^{\epsilon\vec{H}} + \mathcal{O}\big(\epsilon^3\big)\right)^L \\ &= e^{(L\epsilon)\vec{H}} + (L\epsilon)\,\mathcal{O}\big(\epsilon^2\big) \\ &= e^{\tau\vec{H}} + \tau\mathcal{O}\big(\epsilon^2\big) \\ &= e^{\tau\vec{H}} + \mathcal{O}\big(\epsilon^2\big)\,. \end{aligned}$$

Remarkably, the resulting numerical trajectories are confined to the level sets of a *modified Hamiltonian* given by an $\mathcal{O}\big(\epsilon^2\big)$ perturbation of the exact Hamiltonian (Hairer et al., 2006; Betancourt et al., 2014a).

Although such symplectic integrators are highly accurate, they still introduce an error into the trajectories that can bias the Markov chain and any resulting Monte Carlo estimators. In practice this error is typically compensated with the application of a Metropolis correction, accepting a point along the numerical trajectory only with probability

$$a(p, q) = \min\left(1, \exp\Big(H(q, p) - H \circ \phi^{\widetilde{H}}_{\epsilon,\tau}(q, p)\Big)\right).$$

A critical reason for the scalable performance of such an implementation of Hamiltonian Monte Carlo is that the error in a symplectic integrator scales with the step size, ϵ. Consequently a small bias or a large acceptance probability can be maintained by reducing the step size, regardless of the complexity or dimension of the target distribution (Betancourt et al., 2014a). If the symplectic integrator is compromised, however, then this scalability and generality is lost.

3. Hamiltonian Monte Carlo With Subsampling

A common criticism of Hamiltonian Monte Carlo is that in data-intensive applications the evaluation of potential vector field,

$$\vec{V} = -\frac{\partial V}{\partial q} \frac{\partial}{\partial p},$$

and hence the simulation of numerical trajectories, can become infeasible given the expense of the gradient calculations. This expense has fueled a variety of modifications of the algorithm aimed at reducing the cost of the potential energy, often by any means necessary.

An increasingly popular strategy targets Bayesian applications where the data are independently and identically distributed. In this case the posterior can be manipulated into a product of contributions from each subset of data,

$$\pi(\theta | \mathbf{y}) \propto \pi(\theta) \prod_{j=1}^{J} \pi(y_j | \theta) \, ,$$

and the potential energy likewise decomposes into a sum,

$$\begin{aligned} V(q) &= \sum_{j=1}^{J} V_j(q) \\ &= - \sum_{j=1}^{J} \left(\frac{1}{J} \log \pi(\theta) + \log \pi(y_j | \theta) \right), \end{aligned}$$

where each V_j depends on only a single subset. This decomposition suggests algorithms which consider not the entirety of the data and the full potential energy, V, but rather only a few subsets at a time.

Using only part of the data to generate a trajectory, however, compromises the structure-preserving properties of the symplectic integrator and hence the scalability of its accuracy. Consequently the performance of any such subsampling method depends critically on the details of the implementation and the structure of the data itself. Here I consider the performance of two immediate implementations, one based on subsampling the data in between Hamiltonian trajectories and one based on subsampling the data within a single trajectory. As expected, the performance of both methods leaves much to be desired.

3.1. Subsampling Data In Between Trajectories

Given any subset of the data, we can approximate the potential energy as $V \approx J \, V_j$ and then generate trajectories corresponding to the flow of the approximate Hamiltonian, $H_j = T + J \, V_j$. In order to avoid parsing the entirety of the data, the Metropolis correction at the end of each trajectory can be neglected and the corresponding samples left biased. Unlike the numerical trajectories from the full Hamiltonian, these subsampled trajectories are biased away from the exact trajectories regardless of the chosen step size. In particular, the bias of each step,

$$\begin{aligned} e^{\frac{\epsilon}{2} I \vec{V}_j} \circ e^{\epsilon \vec{T}} \circ e^{\frac{\epsilon}{2} I \vec{V}_j} &= e^{\epsilon \vec{H}_j} + \mathcal{O}\big(\epsilon^3\big) \\ &= e^{\epsilon \vec{H} - \epsilon \overrightarrow{\Delta V}_j} + \mathcal{O}\big(\epsilon^3\big) \, , \end{aligned}$$

where

$$\overrightarrow{\Delta V_j} = - \left(\frac{\partial V}{\partial q} - J \frac{\partial V_j}{\partial q} \right) \frac{\partial}{\partial p}, \qquad (1)$$

persists over the entire trajectory,

$$\left(e^{\frac{\epsilon}{2} J \vec{V}_j} \circ e^{\epsilon \vec{T}} \circ e^{\frac{\epsilon}{2} J \vec{V}_j} \right)^{L} = e^{\tau \left(\vec{H} - \overrightarrow{\Delta V}_j \right)} + \mathcal{O}\big(\epsilon^2\big) \, .$$

As the dimension of the target distribution grows, the subsampled gradient, $J \, \partial V_j / \partial q$, drifts away from the true gradient, $\partial V / \partial q$, unless the data become increasingly redundant. Consequently the resulting trajectory introduces an irreducible bias into the algorithm, similar in nature to the asymptotic bias seen in subsampled Langevin Monte Carlo (Teh et al., 2014; Vollmer et al., 2015), which then induces either a vanishing Metropolis acceptance probability or highly-biased expectations if the Metropolis correction is neglected (Figure 1).

Unfortunately, the only way to decrease the dependency on redundant data is to increase the size of each subsample, which immediately undermines any computational benefits.

Consider, for example, a simple application where we target a one-dimensional posterior distribution,

$$\pi(\mu | \mathbf{y}) \propto \pi(\mathbf{y} | \mu) \, \pi(\mu) \, , \qquad (2)$$

with the likelihood

$$\pi(\mathbf{y} | \mu) = \prod_{n=1}^{N} \mathcal{N}\big(y_n | \mu, \sigma^2\big)$$

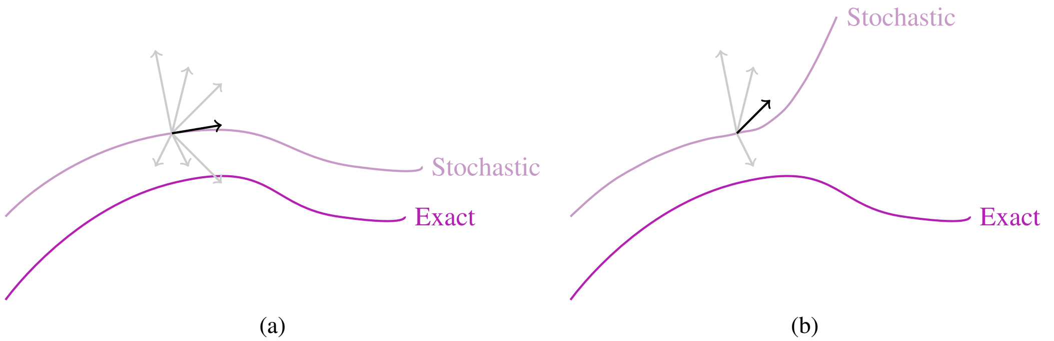
and prior

$$\pi(\mu) = \mathcal{N}\big(\mu | m, s^2\big) \, .$$

Separating the data into $J = N / B$ batches of size B and decomposing the prior into J individual terms then gives

$$\begin{aligned} V_j = \text{const} &+ \frac{B}{N} \frac{\sigma^2 + N s^2}{\sigma^2 s^2} \\ &\times \left(\mu - \frac{ \left(\frac{1}{B} \sum_{n=(j-1)B+1}^{jB} x_n \right) N s^2 + m \sigma^2 }{ \sigma^2 + N s^2 } \right)^2 . \end{aligned}$$

Here I take $\sigma = 2$, $m = 0$, $s = 1$, and generate $N = 500$ data points assuming $\mu = 1$.



(a) (b)

Figure 1. The bias induced by subsampling data in Hamiltonian Monte Carlo depends on how precisely the gradients of the subsampled potential energies integrate to the gradient of the true potential energy. (a) When the subsampled gradient is close to the true gradient, the stochastic trajectory will follow the true trajectory and the bias will be small. (b) Conversely, if the subsampled gradient is not close to the true potential energy then the stochastic trajectory will drift away from the true trajectory and induce a bias. Subsampling between trajectories requires that each subsampled gradient approximate the true gradient, while subsampling within a single trajectory requires only that the average of the subsampled gradients approximates the true gradient. As the dimension of the target distribution grows, however, an accurate approximation in either case becomes increasingly more difficult unless the data become correspondingly more redundant relative to the complexity of the target distribution.

When the full data are used, numerical trajectories generated by the second-order symplectic integrator constructed above closely follow the true trajectories (Figure 2a). Approximating the potential with a subsample of the data introduces the aforementioned bias, which shifts the stochastic trajectory away from the exact trajectory despite negligible error from the symplectic integrator itself (Figure 2b). Only when the size of each subsample approaches the full data set, and the computational benefit of subsampling fades, does the stochastic trajectory provide a reasonable approximation to the exact trajectory (Figure 2c)

As noted above, geometric considerations suggest that this bias should grow with the dimensionality of the target distribution. To see this, consider running subsampled Hamiltonian Monte Carlo on the multivariate generalization of (2),

$$\prod_{d=1}^{D} \pi(\mu_d | \mathbf{y}_d) \, , \tag{3}$$

where the true μ_d are sampled from $\mu_d \sim \mathcal{N}(0,1)$ and trajectories are generated using a subsampled integrator with step size, ϵ, a random integration time $\tau \sim U(0, 2\pi)$, and no Metropolis correction. As a surrogate for the accuracy of the resulting samples I will use the average Metropolis acceptance probability of each new state using the full data.

When the full data are used in this model, the step size of the symplectic integrator can be tuned to maintain constant accuracy as the dimensionality of the target distribution, D, increases. The bias induced by subsampling between trajectories, however, is invariant to the step size of the integrator and rapidly increases with the dimension of the target distribution. Here the data were partitioned into $J = 25$ batches of $B = 20$ data, the subsample used for each trajectory is randomly selected from the first five batches, and the step size of the subsampled trajectory is reduced by $N/(J \cdot B) = 5$ to equalize the computational cost with full data trajectories (Figure 3).

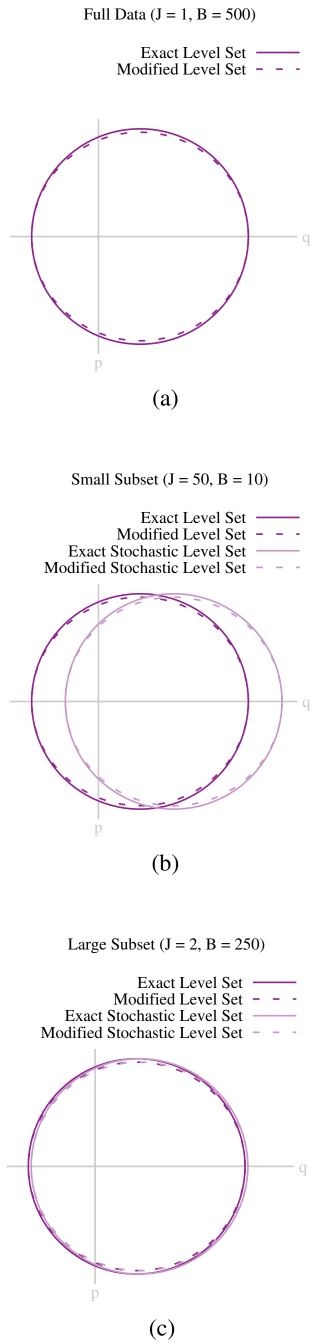
3.2. Subsampling Data within a Single Trajectory

Given that using a single subsample for an entire trajectory introduces an irreducible bias, we might next consider subsampling at each step within a single trajectory, hoping that the bias from each subsample cancels in expectation. Ignoring any Metropolis correction, this is exactly the *naive stochastic gradient Hamiltonian Monte Carlo* of (Chen et al., 2014).

To understand the accuracy of this strategy consider building up such a stochastic trajectory one step at a time. Given the first two randomly-selected subsamples, V_i and then V_j, the first two steps of the resulting integrator are given by

$$\begin{aligned} \phi^{H_j}_{\epsilon} \circ \phi^{H_i}_{\epsilon} &= e^{\epsilon \vec{H} - \epsilon \overrightarrow{\Delta V}_j} \circ e^{\epsilon \vec{H} - \epsilon \overrightarrow{\Delta V}_i} + \mathcal{O}\big(\epsilon^3\big) \\ &= \exp\Big(2 \epsilon \vec{H} - \epsilon \left(\overrightarrow{\Delta V}_i + \overrightarrow{\Delta V}_j \right) \\ &\qquad + \frac{\epsilon^2}{2} \left[\vec{H} - \overrightarrow{\Delta V}_j , \vec{H} - \overrightarrow{\Delta V}_i \right] \Big) \\ &\quad + \mathcal{O}\big(\epsilon^3\big) \end{aligned}$$

$$\begin{aligned} \phi^{H_j}_{\epsilon} \circ \phi^{H_i}_{\epsilon} &= \exp\Big(2 \epsilon \vec{H} - \epsilon \left(\overrightarrow{\Delta V}_i + \overrightarrow{\Delta V}_j \right) \\ &\qquad + \frac{\epsilon^2}{2} \left(- \left[\vec{H}, \vec{V}_{\backslash i} \right] - \left[\vec{V}_{\backslash j}, \vec{H} \right] \right) \Big) \\ &\quad + \mathcal{O}\big(\epsilon^3\big) \, , \end{aligned}$$


Full Data (J = 1, B = 500)
Exact Level Set
Modified Level Set
q
p
Small Subset (J = 50, B = 10)
Exact Level Set
Modified Level Set
Exact Stochastic Level Set
Modified Stochastic Level Set
q
p
Large Subset (J = 2, B = 250)
Exact Level Set
Modified Level Set
Exact Stochastic Level Set
Modified Stochastic Level Set
q
p

(a)

(b)

(c)

Figure 2. Even for the simple posterior (2), subsampling data in between trajectories introduces significant pathologies. (a) When the full data are used, numerical trajectories (dashed line) closely track the exact trajectories (solid line). Subsampling of the data introduces a bias in both the exact trajectories and corresponding numerical trajectories. (b) If the size of each subsample is small then this bias is large. (c) Only when the size of the subsamples approaches the size of the full data, and any computational benefits from subsampling wane, do the stochastic trajectories provide a reasonable emulation of the true trajectories.

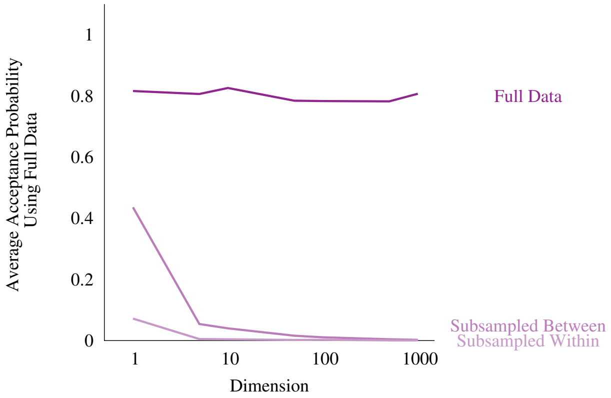


Figure 3. When the full data are used, high accuracy of Hamiltonian Monte Carlo samples, here represented by the average Metropolis acceptance probability using the full data, can be maintained even as the dimensional of the target distribution grows. The biases induced when the data are subsampled, however, cannot be controlled and quickly devastate the accuracy of the algorithm. Here the step size of the subsampled algorithms has been decreased relative to the full data algorithm in order to equalize the computational cost – even in this simple example, a proper implementation of Hamiltonian Monte Carlo can achieve a given accuracy much more efficiently than subsampling.

where we have used the fact that the vector fields $\{\overrightarrow{\Delta V}_j\}$ commute with each other. Similarly, the first three steps are given by

$$\begin{aligned}
&\phi^{H_k}_{\epsilon} \circ \phi^{H_j}_{\epsilon} \circ \phi^{H_i}_{\epsilon} \\
&\quad = \exp\Big(3\epsilon\vec{H} - \epsilon\left(\overrightarrow{\Delta V}_i + \overrightarrow{\Delta V}_j + \overrightarrow{\Delta V}_k\right) \\
&\qquad\quad + \frac{\epsilon^2}{2}\left(-\left[\vec{H}, \overrightarrow{\Delta V}_i\right] - \left[\overrightarrow{\Delta V}_j, \vec{H}\right] \right) \\
&\qquad\quad + \frac{\epsilon^2}{2}\left(\left[\vec{H} - \overrightarrow{\Delta V}_k, 2\vec{H} - \overrightarrow{\Delta V}_i - \overrightarrow{\Delta V}_j\right] \right) \Big) \\
&\qquad + \mathcal{O}\left(\epsilon^3\right) \\
&\quad = \exp\Big(3\epsilon\vec{H} - \epsilon\left(\overrightarrow{\Delta V}_i + \overrightarrow{\Delta V}_j + \overrightarrow{\Delta V}_k\right) \\
&\qquad\quad - \epsilon^2\left(\left[\vec{H}, \overrightarrow{\Delta V}_i\right] - \left[\vec{H}, \overrightarrow{\Delta V}_k\right] \right) \Big) + \mathcal{O}\left(\epsilon^3\right),
\end{aligned}$$

and, letting j_l denote the subsample chosen at the l-th step, the composition over an entire trajectory becomes

$$\begin{aligned}
&\circ_{l=1}^{L} \phi_{\epsilon}^{H_{j_l}} \\
&\quad = \exp\Bigg((L\epsilon)\, \vec{H} - (L\epsilon)\, \frac{1}{L} \sum_{l=1}^{L} \overrightarrow{\Delta V}_{j_l} \\
&\qquad\qquad + (L\epsilon)\, \epsilon \left(\left[\vec{H}, \overrightarrow{\Delta V}_{j_1} \right] - \left[\vec{H}, \overrightarrow{\Delta V}_{j_l} \right] \right) \Bigg) \\
&\qquad + (L\epsilon)\, \mathcal{O}\big(\epsilon^2\big) \\
&\quad = \exp\Bigg(\tau \vec{H} - \tau \frac{1}{L} \sum_{l=1}^{L} \overrightarrow{\Delta V}_{j_l} \\
&\qquad\qquad + \tau\epsilon \left(\left[\vec{H}, \overrightarrow{\Delta V}_{j_1} \right] - \left[\vec{H}, \overrightarrow{\Delta V}_{j_L} \right] \right) \Bigg) + \mathcal{O}\big(\epsilon^2\big) \\
&\quad = \exp\Big(\tau \vec{H} + \tau B_1 + \tau B_2 \Big) + \mathcal{O}\big(\epsilon^2\big) \, ,
\end{aligned}$$

where

$$B_1 = - \frac{1}{L} \sum_{l=1}^{L} \overrightarrow{\Delta V}_{j_l}$$

and

$$B_2 = \epsilon \left(\left[\vec{H}, \overrightarrow{\Delta V}_{j_1} \right] - \left[\vec{H}, \overrightarrow{\Delta V}_{j_L} \right] \right) .$$

Once again, subsampling the data introduces bias into the numerical trajectories.

Although the second source of bias, B_2, is immediately rectified by appending the stochastic trajectory with an update from the initial subsample such that $j_L = j_1$, the first source of bias, B_1, is not so easily remedied. Expanding,

$$\begin{aligned}
\frac{1}{L} \sum_{l=1}^{L} \overrightarrow{\Delta V}_{j_l} &= \frac{1}{L} \sum_{l=1}^{L} \left(\vec{V} - J \, \vec{V}_{j_l} \right) \\
&= \vec{V} - \frac{J}{L} \sum_{n=1}^{L} \vec{V}_{j_l} \\
&= - \left(\frac{\partial V}{\partial q} - \frac{J}{L} \sum_{l=1}^{L} \frac{\partial V_j}{\partial q} \right) \frac{\partial}{\partial p},
\end{aligned}$$

we see that B_1 vanishes only when the average gradient of the selected subsamples yields the gradient of the full potential. Averaging over subsamples may reduce the bias compared to using a single subsample over the entire trajectory (1), but the bias still scales poorly with the dimensionality of the target distribution (Figure 1).

In order to ensure that the bias vanishes identically and independent of the redundancy of the data, we have to use each subsample the same number of times within a single trajectory. In particular, both biases vanish if we use each subsample twice in a symmetric composition of the form

$$\left(\circ_{l=1}^{L} \phi_{\epsilon}^{H_l} \right) \circ \left(\circ_{l=1}^{L} \phi_{\epsilon}^{H_{L+1-l}} \right) .$$

Because this composition requires using all of the subsamples it does not provide any computational savings and it seems rather at odd with the original stochastic subsampling motivation.

Indeed, this symmetric composition is not stochastic at all and actually corresponds to a rather elaborate symplectic integrator where the potential energy from each subsample generates its own flow, equivalent to the integrator in Split Hamiltonian Monte Carlo (Shahbaba et al., 2014) with the larger step size $J\epsilon$. Removing intermediate steps from this symmetric, stochastic trajectory (Figure 4a) reveals the level set of the corresponding modified Hamiltonian (Figure 4b). Because this symmetric composition integrates the full Hamiltonian system, the error is once again controllable and vanishes as the step size is decreased (Figure 4c).

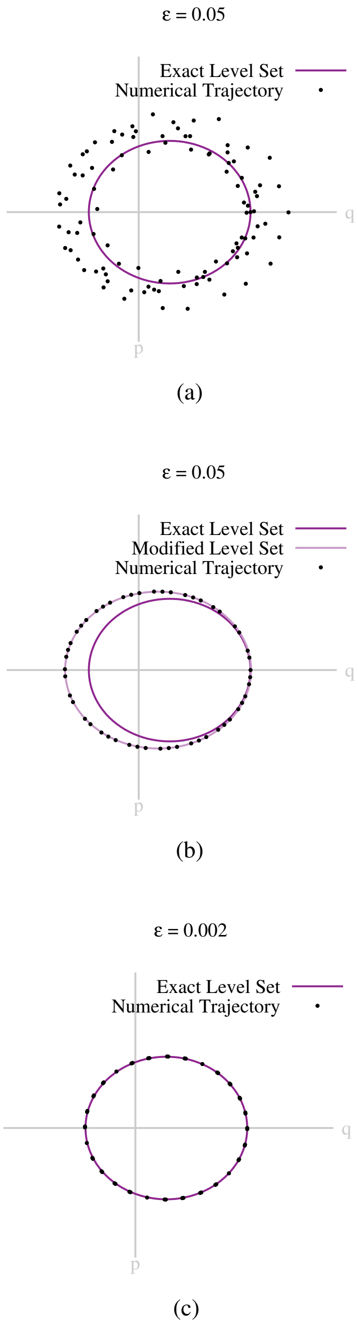
Limiting the number of subsamples, however, leaves the irreducible bias in the trajectories that cannot be controlled by the tuning the step size (Figures 3, 5). Once more we are left dependent on the redundancy of the data for any hope of improved performance with subsampling.

4. Conclusion

The efficacy of Markov Chain Monte Carlo for complex, high-dimensional target distributions depends on the ability of the sampler to explore the intricate and often meandering neighborhoods on which the probability is distributed. Symplectic integrators admit a structure-preserving implementation of Hamiltonian Monte Carlo that is amazingly robust to this complexity and capable of efficiently exploring the most complex target distributions. Subsampled data, however, does not in general have enough information to enable such efficient exploration. This lack of information manifests as an irreducible bias that devastates the scalable performance of Hamiltonian Monte Carlo.

Consequently, without having access to the full data there is no immediate way of engineering a well-behaved implementation of Hamiltonian Monte Carlo applicable to most statistical models. As with so many other subsampling algorithms, the adequacy of a subsampled Hamiltonian Monte Carlo implementation is at the mercy of the redundancy of the data relative to the complexity of the target model, and not in the control of the user.

Unfortunately many of the problems at the frontiers of applied statistics are in the *wide data* regime, where data are sparse relative to model complexity. Here subsampling methods have little hope of success; we must focus our efforts not on modifying Hamiltonian Monte Carlo but rather on improving its implementation with, for example, better memory management and efficiently parallelized gradient calculations.

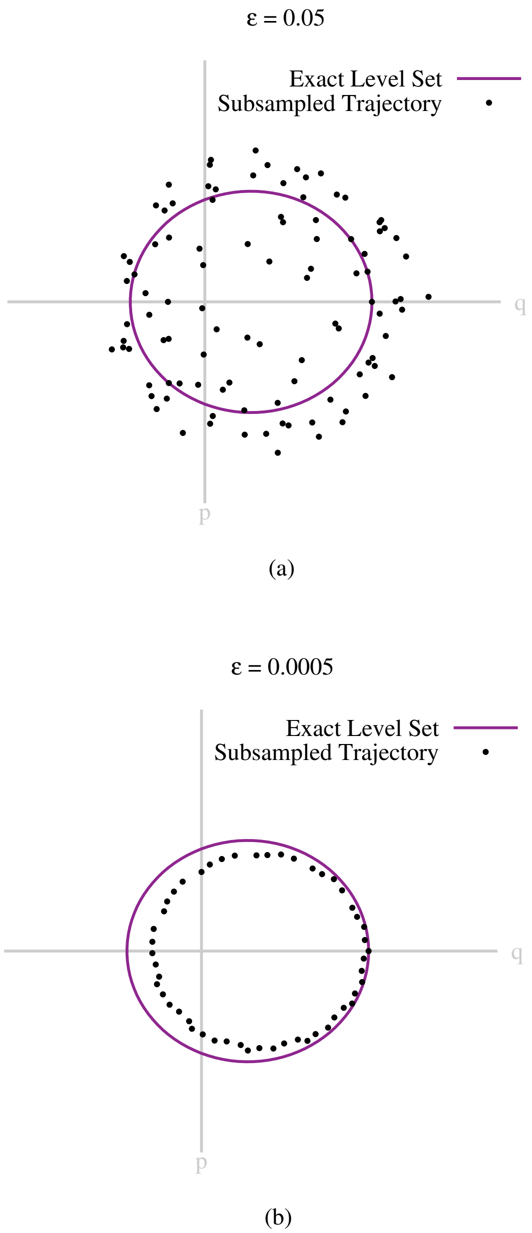


(a)

(b)

(c)

Figure 4. The symmetric composition of flows from each subsamples of the data eliminates all bias in the stochastic trajectory because it implicitly reconstructs a symplectic integrator. Refining (a) all intermediate steps in a stochastic trajectory (b) to only those occurring after a symmetric sweep of the subsamples reveals the level set of the modified Hamiltonian corresponding to the implicit symplectic integrator. Because of the vanishing bias, (c) the error in the stochastic trajectory can be controlled by taking the step size to zero.



(a)

(b)

Figure 5. (a) Utilizing only a few subsamples within a trajectory yields numerical trajectories biased away from the exact trajectories. (b) Unlike the error introduced by a full symplectic integrator, this bias is irreducible and cannot be controlled by tuning the step size. The performance of such an algorithm is limited by the size of the bias which itself depends on the redundancy of the data relative to the target model.

Acknowledgements

It is my pleasure to thank Simon Byrne, Mark Girolami, Matt Hoffman, Matt Johnson, and Sam Livingstone for thoughtful discussion and helpful comments, as well as three anonymous reviewers for constructive assessments. I am supported under EPSRC grant EP/J016934/1.

References

Bardenet, Rémi, Doucet, Arnaud, and Holmes, Chris. An adaptive subsampling approach for MCMC inference in large datasets. In *Proceedings of The 31st International Conference on Machine Learning*, pp. 405–413, 2014.

Betancourt, Michael, Byrne, Simon, and Girolami, Mark. Optimizing the integrator step size for hamiltonian monte carlo. *ArXiv e-prints*, 1410.5110, 11 2014a.

Betancourt, Michael, Byrne, Simon, Livingstone, Samuel, and Girolami, Mark. The geometric foundations of Hamiltonian Monte Carlo. *ArXiv e-prints*, 1410.5110, 10 2014b.

Chen, Tianqi, Fox, Emily B, and Guestrin, Carlos. Stochastic gradient Hamiltonian Monte Carlo. *Proceedings of The 31st International Conference on Machine Learning*, pp. 1683–1691, 2014.

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

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

Hairer, E., Lubich, C., and Wanner, G. *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*. Springer, New York, 2006.

Lee, John M. *Introduction to Smooth Manifolds*. Springer, 2013.

Leimkuhler, B. and Reich, S. *Simulating Hamiltonian Dynamics*. Cambridge University Press, New York, 2004.

Neal, R.M. MCMC using Hamiltonian dynamics. In Brooks, Steve, Gelman, Andrew, Jones, Galin L., and Meng, Xiao-Li (eds.), *Handbook of Markov Chain Monte Carlo*. CRC Press, New York, 2011.

Neiswanger, Willie, Wang, Chong, and Xing, Eric. Asymptotically exact, embarrassingly parallel MCMC. *arXiv e-prints*, 1311.4780, 2013.

Shahbaba, Babak, Lan, Shiwei, Johnson, Wesley O, and Neal, Radford M. Split Hamiltonian Monte Carlo. *Statistics and Computing*, 24(3):339–349, 2014.

Teh, Yee Whye, Thiéry, Alexandre, and Vollmer, Sebastian. Consistency and fluctuations for stochastic gradient Langevin dynamics. *ArXiv e-prints*, 1409.0578, 09 2014.

Vollmer, Sebastian J., Zygalakis, Konstantinos C., , and Teh, Yee Whye. (non-)asymptotic properties of stochastic gradient Langevin dynamics. *ArXiv e-prints*, 1501.00438, 01 2015.

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