Article Review: The No-U-Turn Sampler

Adonis Jamal
ENS Paris-Saclay
adonis.jamal@ens-paris-saclay.fr

Jean-Vincent Martini
ENS Paris-Saclay
jean-vincent.martini@ens-paris-saclay.fr

Abstract

This paper provides a detailed study of the paper "The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo" by Matthew D. Hoffman and Andrew Gelman Hoffman et al. [2014]. The paper introduces the No-U-Turn Sampler, an extension of Hamiltonian Monte Carlo that adaptively determines the path length during sampling, eliminating the need for manual tuning of this parameter. We further extend this analysis by providing a critical evaluation of the strengths and weaknesses of NUTS and discussing TODO . The full code related to this report is available at https://github.com/adonis107/MVA-BML.

a remplir

1 Introduction and Context

This report provides a comprehensive review of the article "The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo" written by Matthew D. Hoffman and Andrew Gelman Hoffman et al. [2014]. The paper addresses a central practical limitation of Hamiltonian Monte Carlo (HMC) methods: the need to manually tune the path length parameter, which can significantly affect the efficiency of sampling. By proposing the No-U-Turn Sampler (NUTS), the authors introduce an extension to HMC that adaptively determines the path length during sampling while retaining, and in some cases improving, the efficiency of well-tuned HMC.

The context of this work is rooted in the broader field of Bayesian computation. For complex models, exact posterior inference is often intractable, necessitating the use of approximate methods. While deterministic approximations can be computationally attractive Wainwright and Jordan [2008], Markov Chain Monte Carlo (MCMC) methods remain appealing because they are asymptotically unbiased and broadly applicable Neal [1993]. However, traditional algorithms such as random-walk Metropolis Metropolis et al. [1953] and Gibbs sampling Geman and Geman [1984] suffer from inefficient exploration in high-dimensional or highly correlated parameter spaces. HMC addresses these issues by introducing auxiliary momentum variables and simulating Hamiltonian dynamics to propose new states, allowing for larger and more informed jumps in the parameter space Neal et al. [2011]. Despite its advantages, HMC requires the user to specify two sensitive parameters: the step size and the number of leapfrog steps (or equivalently, the integration time). While adaptive methods have been developed to mitigate this issue Andrieu and Thoms [2008], Nesterov [2009], it remains a significant barrier that limits the accessibility and widespread adoption of HMC in general-purpose software Hoffman et al. [2014].

The NUTS algorithm eliminates the need to specify the number of leapfrog steps by dynamically building a set of candidate states along the Hamiltonian trajectory and stopping automatically when the simulated path begins to double back on itself, hence the name "No-U-Turn". In addition, the authors introduce a primal–dual averaging scheme to adapt the step size during a warm-up phase.

Together, these innovations yield a nearly tuning-free algorithm that preserves the theoretical strengths of HMC while dramatically improving usability.

In this report, we first examine the theoretical, computational, and empirical methods developed in the paper, detailing how NUTS modifies and extends standard HMC. We then critically assess the strengths and limitations of the approach, both from a theoretical and practical perspective. Finally, we TODO.

a remplir

2 The No-U-Turn Sampler: Methodology and Analysis

2.1 Problem Definition and Limitations of HMC

HMC Neal et al. [2011], Duane et al. [1987] introduces auxiliary momentum variables $r \in \mathbb{R}^D$ paired with the parameters of interest $\theta \in \mathbb{R}^D$, defining the joint distribution $p(\theta, r) \propto \exp\left\{\mathcal{L}(\theta) - \frac{1}{2} r \cdot r\right\}$, where $\mathcal{L}(\theta)$ is the log-joint density up to a normalising constant. Each iteration resamples $r \sim \mathcal{N}(0, I)$ and simulates Hamiltonian dynamics via the leapfrog integrator for L steps of size ϵ, producing a proposal accepted or rejected via a Metropolis step. Because the leapfrog integrator is volume-preserving, proposals can be simultaneously distant and highly probable, enabling efficient exploration of high-dimensional targets.

However, as mentioned earlier, HMC's performance depends critically on both ϵ and L. Too large an ϵ yields high rejection rates and too small wastes computation. If L is too small the chain degenerates into a random walk, while if L is too large the trajectory loops back on itself, wasting gradient evaluations. Selecting a good L typically requires several costly preliminary runs and considerable expertise, limiting HMC's adoption in general-purpose inference software Hoffman et al. [2014].

2.2 The No-U-Turn Criterion and Trajectory Construction

NUTS eliminates the need to specify L by stopping trajectory simulation automatically when further simulation would no longer move the proposal away from the starting point. The key quantity is the time derivative of half the squared distance between current position $\tilde{\theta}$ and initial position θ. When this derivative becomes negative, the trajectory is doubling back, making further simulation inefficient. A naive stopping rule based on this criterion would violate time-reversibility, so NUTS instead builds trajectories both forwards and backwards in fictitious time via a recursive doubling scheme.

At each doubling step j, a direction $v_j \sim \text{Uniform}(\{-1, +1\})$ is drawn and 2^j leapfrog steps of size $v_j \epsilon$ are taken, implicitly constructing a balanced binary tree whose leaves are visited position-momentum states. To ensure detailed balance, the model is augmented with a slice variable $u \sim \text{Uniform}([0, \exp\{\mathcal{L}(\theta) - \frac{1}{2} r \cdot r\}])$, and the candidate set $\mathcal{C} \subseteq \mathcal{B}$ is restricted to visited states lying within the slice. Doubling halts either upon numerical instability or when a U-turn is detected in any balanced subtree, i.e., when the leftmost and rightmost leaves (θ^-, r^-), (θ^+, r^+) satisfy

$$(\theta^+ - \theta^-) \cdot r^- < 0 \quad \text{or} \quad (\theta^+ - \theta^-) \cdot r^+ < 0.$$

States added during a halting iteration are excluded from $\mathcal{C}$ to preserve detailed balance. The efficient implementation further reduces memory from $O(2^j)$ to $O(j)$ by propagating a single representative state through the recursion and applying a Metropolis-Hastings accept/reject step after each doubling rather than sampling from $\mathcal{C}$ explicitly at the end.

2.3 Adaptive Step Size Selection via Dual Averaging

To also automate the tuning of ϵ, the authors adapt the primal-dual averaging algorithm of Nesterov [2009]. The step size is only updated during a warm-up phase to drive the average acceptance statistic towards a target δ, via:

$$x_{t+1} = \mu - \frac{\sqrt{t}}{\gamma(t + t_0)} \sum_{i=1}^{t} H_i, \qquad \bar{x}_{t+1} = t^{-\kappa} x_{t+1} + (1 - t^{-\kappa}) \bar{x}_t,$$

where $x = \log \epsilon$, H_t is a custom statistic that describes the behavior of an MCMC algorithm at iteration $t \geq 1$ (possibly a function of the acceptance rate of the Metropolis step), $\mu = \log(10\epsilon_0)$

encourages testing larger values of ϵ, and γ, t_0, κ are fixed hyperparameters set to 0.05, 10, and 0.75 throughout. This scheme renders NUTS entirely tuning-free.

2.4 Empirical Evaluation

The authors evaluate NUTS and HMC on four targets: a 250-dimensional multivariate normal, the posterior of a Bayesian logistic regression (25 parameters) and its hierarchical extension (302 parameters) both fitted to the German credit data set, and a 3001-dimensional stochastic volatility model fitted to S&P 500 returns. Performance is measured as effective sample size (ESS) per gradient evaluation, with the minimum ESS across dimensions reported. HMC is tested across a grid of simulation lengths and target acceptance rates, while NUTS is tested across values of δ only, for a total of over 3800 experiments.

The dual averaging scheme reliably matches the realised acceptance statistic to its target δ and $\bar{\epsilon}$ converges within a few hundred warm-up iterations on most models. NUTS trajectory lengths concentrate on integer powers of two, confirming that the U-turn criterion is typically triggered only after a complete doubling. In terms of ESS per gradient, NUTS matches or exceeds the best tuned HMC across all four distributions, outperforming it by roughly a factor of two on the multivariate normal and 1.5 on the stochastic volatility model, without requiring any tuning of the trajectory length.

3 Critical Evaluation

4 Our Implementation and Extensions

4.1 The Transition Kernel and Proof of Detailed Balance

To reduce memory complexity from $\mathcal{O}(2^j)$ to $\mathcal{O}(j)$, Efficient NUTS introduces a custom transition kernel $T(w'|w, \mathcal{C})$. Instead of uniformly sampling the full candidate set $\mathcal{C}$, the algorithm partitions $\mathcal{C}$ into $\mathcal{C}^{old}$ (previously visited states) and $\mathcal{C}^{new}$ (states added during the final doubling). For a current state $w \in \mathcal{C}^{old}$, it proposes a jump to $w' \in \mathcal{C}^{new}$ uniformly and accepts it with probability $\min\left(1, \frac{|\mathcal{C}^{new}|}{|\mathcal{C}^{old}|}\right)$. This allows incremental sampling from subtrees without storing all states, while favoring exploration of distant regions.

The authors state that this kernel satisfies detailed balance with respect to the uniform distribution $p(w|\mathcal{C}) = \frac{1}{|\mathcal{C}|}$, but omit the exact proof. We provide a brief derivation here.

For $w \in \mathcal{C}^{old}$ and $w' \in \mathcal{C}^{new}$, the detailed balance condition requires:

$$p(w|\mathcal{C})T(w'|w, \mathcal{C}) = p(w'|\mathcal{C})T(w|w', \mathcal{C}) \tag{1}$$

Substituting the respective proposal and acceptance probabilities yields:

$$\frac{1}{|\mathcal{C}|}\frac{1}{|\mathcal{C}^{new}|}\min\left(1, \frac{|\mathcal{C}^{new}|}{|\mathcal{C}^{old}|}\right) = \frac{1}{|\mathcal{C}|}\frac{1}{|\mathcal{C}^{old}|}\min\left(1, \frac{|\mathcal{C}^{old}|}{|\mathcal{C}^{new}|}\right)$$

Multiplying both sides by $|\mathcal{C}||\mathcal{C}^{old}||\mathcal{C}^{new}|$ simplifies the expression to:

$$|\mathcal{C}^{old}|\min\left(1, \frac{|\mathcal{C}^{new}|}{|\mathcal{C}^{old}|}\right) = |\mathcal{C}^{new}|\min\left(1, \frac{|\mathcal{C}^{old}|}{|\mathcal{C}^{new}|}\right)$$

Distributing the factors into the minimum functions gives:

$$\min(|\mathcal{C}^{old}|, |\mathcal{C}^{new}|) = \min(|\mathcal{C}^{new}|, |\mathcal{C}^{old}|)$$

Since both sides are trivially equal, detailed balance holds, proving that the memory-efficient transition kernel preserves the target uniform distribution over $\mathcal{C}$.

a voir si on garde

5 Conclusion

References

Matthew D Hoffman, Andrew Gelman, et al. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623, 2014.

Martin J Wainwright and Michael I Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305, 2008.

Radford M Neal. Probabilistic inference using markov chain monte carlo methods. 1993.

Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741, 1984.

Radford M Neal et al. Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2, 2011.

Christophe Andrieu and Johannes Thoms. A tutorial on adaptive mcmc. *Statistics and computing*, 18(4): 343–373, 2008.

Yurii Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical programming*, 120(1): 221–259, 2009.

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