Ricochet Monte Carlo: Dual Purpose MCMC Sampling & Nonlinear Programming

Anonymous Authors[1]

Abstract

We introduce a novel dual-purpose algorithm for MCMC sampling and constrained nonlinear optimization. The method simulates a particle bouncing across a surface whose shape is determined by the target distribution or objective function, using a notion of entropic dissipation that resembles simulated annealing without any acceptance step. Similarly, the simple closed-form parabolic trajectories means avoiding costly and potentially unstable numerical integrations to compute trajectories. This enables a more efficient simulation, reducing total evaluations of the target. Our method easily extends to handle nonlinear constraints. We demonstrate the effectiveness of our method on several distributions and nonlinear test functions.

1. Introduction

1.1. Related work

1.2. Outline of work

2. Ricochet Monte Carlo

The intuition behind Ricochet Monte Carlo (RMC) is to treat our target distribution or objective as a sort of hilly terrain across which a point particle freely bounces along. The points where the particle collides with this surface are candidate samples to either be accepted and rejected. The particle will continue to bounce, until it settles in some "valley", at which time it is picked up and tossed in a random direction to repeat the process.

Let $F(\boldsymbol{\theta})$ be some probability distribution to sample from, defined as

$$F(\boldsymbol{\theta}) \coloneqq \frac{E(\boldsymbol{\theta})}{Z}$$

where Z is some intractable normalizing constant and $E(\boldsymbol{\theta})$ is the associated joint distribution, which is assumed tractable and known. We are interested in generating samples $\boldsymbol{\theta}_i \sim F$ that live in a continuous d-dimensional vector space, where i means the ith sample. We perform the standard transformation as done in HMC and define our *surface* as

$$S(\boldsymbol{\theta}) \coloneqq -\log E(\boldsymbol{\theta}).$$

If instead $F(\boldsymbol{\theta})$ is a nonlinear objective function to minimize, then we simply set $S(\boldsymbol{\theta}) \coloneqq F(\boldsymbol{\theta})$.

2.1. Auxillary variables

For each value of $\boldsymbol{\theta}$, we associate an auxillary *height* scalar h. We define a $(d+1)$-length *position* vector $\mathbf{q} \coloneqq [\boldsymbol{\theta}; h]$. We also create an auxillary $(d+1)$-length *position* vector $\mathbf{p}$. We define functions

$$\boldsymbol{\theta}(\mathbf{q}) \coloneqq \boldsymbol{\theta}$$
$$h(\mathbf{q}) \coloneqq h$$

to respectively refer to the specific sample and height components of any particular $\mathbf{q}$. Similarly, we overload $\boldsymbol{\theta}(\mathbf{p})$ to be the vector of the first d entries of $\mathbf{p}$.

We maintain an invariant where the particle is always "above" the surface S, i.e.

$$C_S(\mathbf{q}) \coloneqq h(\mathbf{q}) - S(\boldsymbol{\theta}(\mathbf{q})) > 0. \tag{1}$$

Equation (1) defines the *surface collision* constraint C_S. This terminology will be used to distinguish it from any later nonlinear constraints that will be handled in the optimization formulation. All together, $\mathbf{q}$, $\mathbf{p}$, and S fully describe the state of our system.

2.2. Simulation with Hamiltonian dynamics

To simulate the particle's trajectory across the surface, we start with its associated Hamiltonian equations. Given scalars $m > 0$ and $g > 0$, we define

$$H(\mathbf{q}, \mathbf{p}) \coloneqq U(\mathbf{q}) + K(\mathbf{p}) \tag{2}$$
$$U(\mathbf{q}) \coloneqq m \cdot g \cdot h(\mathbf{q}) \tag{3}$$
$$K(\mathbf{p}) \coloneqq \frac{\|\mathbf{p}\|^2}{2m}. \tag{4}$$

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

Algorithm 1 Find earliest t' with imminent collision on C_S

```
procedure TemporalSearch($C_S$, $\mathbf{q}_0$, $\mathbf{p}_0$, $g$, $m$, $\Delta_0$)
    $(t_{\text{start}}, t_{\text{end}}) \leftarrow (0, \Delta_0)$
    repeat                          ▷ Double timestep in forward scan...
        $t_{\text{start}} \leftarrow t_{\text{end}}$
        $t_{\text{end}} \leftarrow 2 \cdot t_{\text{end}}$
    until $C_S(\mathbf{q}(t_{\text{end}})) \leq 0$          ▷ ...until a collision.
    repeat
        $t_{\text{middle}} \leftarrow \frac{1}{2}(t_{\text{start}} + t_{\text{end}})$
        ▷ If temporal midpoint is a collision...                ◁
        if $C_S(\mathbf{q}(t_{\text{middle}})) \leq 0$ then
            ▷ ...move boundary endtime backward...              ◁
            $t_{\text{end}} \leftarrow t_{\text{middle}}$
        else
            ▷ ...else move boundary starttime forward.         ◁
            $t_{\text{start}} \leftarrow t_{\text{middle}}$
    until $t_{\text{end}} - t_{\text{start}} \leq$ some small tolerance
    ▷ Since $t_{start}$ still satisfies $C_S$, return it.                ◁
    return $t_{\text{start}}$ as $t'$
```

We respectively call m and g the *mass* and *gravity*, and both are hyperparameters of RMC. For convenience of notation, let $\mathbf{g} := [\mathbf{0}_d; g]$ where $\mathbf{0}_d$ is a d-length zero vector. The dynamics of the particle are governed by the following system of differential equations:

$$\frac{d\mathbf{q}}{dt} = \frac{\partial H}{\partial \mathbf{p}} = \frac{dK}{d\mathbf{p}} = \frac{\mathbf{p}}{m} \tag{5}$$

$$\frac{d\mathbf{p}}{dt} = -\frac{\partial H}{\partial \mathbf{q}} = -\frac{dU}{d\mathbf{q}} = -m\mathbf{g}. \tag{6}$$

In order to simulate the particle trajectory, HMC employs Euler's "leapfrog" integration method, which requires an acceptance step to handle cases when the Hamiltonian H diverges and thus isn't conserved. In RMC, because Equations (5) and (6) don't involve what could be an arbitrarily complex $S(\boldsymbol{\theta})$, we can easily solve for the time-parametrized closed forms:

$$\mathbf{q}(t) := \mathbf{q}_0 + \frac{\mathbf{p}_0}{m}t - \frac{\mathbf{g}}{2}t^2 \tag{7}$$

$$\mathbf{p}(t) := \mathbf{p}_0 - m\mathbf{g}t. \tag{8}$$

We use Equations (7) and (8) to find the next candidate sample $\boldsymbol{\theta}_i$ by computing the earliest t' when the particle collides with the S. This search is done using a two-step process that increases t' in doubling increments until a collision, and then does a binary search within a time-bounded window to find the near-exact t' representing the imminent point of collision while still satisfying $C_S(\mathbf{q}(t')) > 0$. Algorithm 1 outlines the procedure in fuller detail.

2.3. Sampling at collisions

Once t' is had, we compute the position $\mathbf{q}_{t'} := \mathbf{q}(t')$ and momentum $\mathbf{p}_{t'} := \mathbf{p}(t')$. The acceptance probability of the candidate sample $\boldsymbol{\theta}(\mathbf{q}_{t'})$ is defined in terms of *lateral momentum*: ignoring the height component, if the angle of the reflection is particularly sharp, we are more likely to reject it. The reasoning is that these "hard" bounces represent regions of low probability in $F(\boldsymbol{\theta})$. If we let

$$\mathbf{n} := \frac{\nabla C_S(\mathbf{q}_{t'})}{\|\nabla C_S(\mathbf{q}_{t'})\|}$$

be the unit normal at $\mathbf{q}_{t'}$, we can define the momentum after reflecting off the surface as

$$\overleftarrow{\mathbf{p}}_{t'} := \mathbf{p}_{t'} - 2 \cdot \langle \mathbf{p}_{t'}, \mathbf{n} \rangle \cdot \mathbf{n}. \tag{9}$$

Our acceptance probability is thus

$$\Pr(\text{ACCEPT} \mid t') := \frac{1}{2}\left(1 + \frac{\langle \boldsymbol{\theta}(\mathbf{p}_{t'}), \boldsymbol{\theta}(\overleftarrow{\mathbf{p}}_{t'}) \rangle}{\|\boldsymbol{\theta}(\mathbf{p}_{t'})\| \|\boldsymbol{\theta}(\overleftarrow{\mathbf{p}}_{t'})\|}\right). \tag{10}$$

If accepted, then we set $\boldsymbol{\theta}_i \leftarrow \boldsymbol{\theta}(\mathbf{q}_{t'})$. Whether the sample is accepted or rejected, we still continue on with the simulation by setting $(\mathbf{q}_0, \mathbf{p}_0) \leftarrow (\mathbf{q}_{t'}, \mathbf{p}_{t'})$ in Equations (7) and (8).

2.4. Entropic dissipation

Given a scalar $\epsilon \in (0, 1]$, we modify Equation (9) slightly:

$$\overleftarrow{\mathbf{p}}_{t'} := \epsilon \cdot (\mathbf{p}_{t'} - 2 \cdot \langle \mathbf{p}_{t'}, \mathbf{n} \rangle \cdot \mathbf{n}). \tag{11}$$

The hyperparameter ϵ is known in the physics literature as the *coefficient of restitution*, a ratio of the velocities before and after an inelastic collision. If $\epsilon = 1$, it is an elastic collision and we have Equation (9) again. Repeated applications of Equation (11) will gradually remove energy from the system until what remains is accounted for in the potential energy $U(\mathbf{q})$.

Lemma 2.1. *Let t be the aggregate time elapsed across all bounces, with $\mathbf{q}(t)$ and $\mathbf{p}(t)$ being the position and momentum at this moment. As $t \to \infty$, $\boldsymbol{\theta}(\mathbf{q}(t))$ will either be a local or global minimum of $S(\boldsymbol{\theta})$.*

Proof. There is a point $\mathbf{q}^*$ such that $S(\boldsymbol{\theta}(\mathbf{q}^*))$ is minimized. It must be the case then that $C_S(\mathbf{q}^*) = 0$ (else we would shift the point downwards). Also, by Equation (1), $C_S(\mathbf{q}(t)) > 0$ always. Let $\Delta U(\mathbf{q}(t)) := U(\mathbf{q}(t)) - U(\mathbf{q}^*)$ be a strictly positive quantity denoting the potential differential. We also observe per Equations (4) and (11) that $K(\overleftarrow{\mathbf{p}}(t)) = \epsilon^2 K(\mathbf{p}(t))$. This implies

$$\lim_{t\to\infty} K(\mathbf{p}(t)) = 0$$

and thus

$$\begin{aligned}\lim_{t\to\infty} H(\mathbf{q}(t), \mathbf{p}(t)) &= U(\mathbf{q}(t)) \\ &= U(\mathbf{q}^*) + \Delta U(\mathbf{q}(t)).\end{aligned}$$

If $K(\mathbf{p}(t)) = 0$ in the limit then the particle has ceased motion, and is resting on S somewhere. If $\Delta U(\mathbf{q}(t)) = 0$, then $\mathbf{q}(t) = \mathbf{q}^*$ and we have settled at the global minimum. Otherwise, we have found a local minimum. $\square$

The result of Lemma 2.1 gives us a means of collecting values $\boldsymbol{\theta}$ that identify potential solutions of $F(\boldsymbol{\theta})$ from which we can select from at the end of the simulation. This is done by running the simulation of bounces until the kinetic energy of the particle drops below some small threshold $\eta > 0$. If that criterion is met, we collect a sample as part of a separate collection of solutions distinct from our MCMC samples. These solutions are therefore not subject to the acceptance step according to Equation (10).

We must refresh momentum variable at the particle has "puttered out". We do this by raising the particle to some random height above S and sampling the momentum randomly as well. For some $\mathbf{q}$ and $\mathbf{p}$ after a collision in which $K(\mathbf{p}) < \eta$, we refresh according to the following:

$$\begin{aligned}\Delta h &\sim \textsc{Rayleigh}(m)\\ \mathbf{q} &\leftarrow [\boldsymbol{\theta}(\mathbf{q}); S(\boldsymbol{\theta}(\mathbf{q})) + \Delta h]\\ \mathbf{p} &\sim \mathcal{N}(0, m\mathbf{I}_{d+1})\end{aligned}$$

where

$$\textsc{Rayleigh}(x \mid \sigma) \coloneqq \frac{x}{\sigma^2} \exp\left(\frac{-x^2}{2\sigma^2}\right)$$

is the Rayleigh distribution with scalar parameter $\sigma > 0$ and support $x \geq 0$. This refresh scheme is visualized as raising the particle to some random height above where it settled and tossing it in a random direction. The raised height and strength of the toss is proportional to the mass m.

Between refreshes, we yield some undefined number of samples, different each time and hard to predict.

2.5. Handling boundary constraints

2.6. Initialization

2.7. Full algorithm

3. Experiments

3.1. Distribution sampling

3.1.1. Single gaussian

3.1.2. Mixture of varied gaussians

3.1.3. Lopsided distributions

3.1.4. Graphical model with boundary constraints

3.2. Nonlinear programming

4. Theoretical analysis & discussion

4.1. Potential extensions

5. Conclusion

References

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

A. You *can* have an appendix here.

You can have as much text here as you want. The main body must be at most 8 pages long. For the final version, one more page can be added. If you want, you can use an appendix like this one, even using the one-column format.