Binary Bouncy Particle Sampler

Ari Pakman
Department of Statistics
Center for Theoretical Neuroscience
Grossman Center for the Statistics of Mind
Columbia University
ari@stat.columbia.edu

Abstract

The Bouncy Particle Sampler is a novel rejection-free non-reversible sampler for differentiable probability distributions over continuous variables. We generalize the algorithm to piecewise differentiable distributions and apply it to generic binary distributions using a piecewise differentiable augmentation. We illustrate the new algorithm in a binary Markov Random Field example, and compare it to binary Hamiltonian Monte Carlo. Our results suggest that binary BPS samplers are better for easy to mix distributions.

1 Introduction

The Bouncy Particle Sampler (BPS) algorithm is a novel generic sampler proposed in [1] and explored in [2, 3]. Given a distribution $p(\mathbf{y})$ with $\mathbf{y} \in \mathbb{R}^d$, the algorithm introduces a random velocity vector $\mathbf{v}$ distributed uniformly on the unit-sphere $\mathbb{S}^d$ and defines a piecewise deterministic Markov process [4] over $(\mathbf{y}, \mathbf{v})$. For reviews and further developments see e.g. [5, 6, 7, 8]. In this contribution we extend the basic BPS algorithm to piecewise differentiable distributions and apply it to generic binary discrete distributions using the augmentation method of [9].

1.1 Discrete Infinitesimal Time Steps

This section is a quick introduction to the BPS sampler for the reader unfamiliar with it, following closely the presentation in [6]. We begin in discrete time and then take the continuous-time limit. Let us introduce first the potential $U(\mathbf{y})$ as

$$p(\mathbf{y}) \propto e^{-U(\mathbf{y})} \qquad \mathbf{y} \in \mathbb{R}^d \tag{1.1}$$

Denoting time by t, consider a discrete Markov process that acts on $(\mathbf{y}, \mathbf{v})$ as

$$(\mathbf{y}, \mathbf{v})_{t+\Delta t} = \begin{cases} (\mathbf{y}+\mathbf{v}\Delta t, \mathbf{v}) & \text{with prob. } 1 - \Delta t[\mathbf{v}\cdot\nabla U(\mathbf{y})]_+ \\ (\mathbf{y}+\mathbf{v}\Delta t, \mathbf{v}_r) & \text{with prob. } \Delta t[\mathbf{v}\cdot\nabla U(\mathbf{y})]_+ \end{cases} \tag{1.2}$$

where

$$[x]_+ = \max(x, 0)\,, \tag{1.3}$$

$$\mathbf{v}_r = \mathbf{v} - 2\frac{(\mathbf{v}\cdot\nabla U(\mathbf{y}))\nabla U(\mathbf{y})}{\|\nabla U(\mathbf{y})\|^2}\,. \tag{1.4}$$

Note that $\mathbf{v}_r$ is a reflection of $\mathbf{v}$ with respect to the plane perpendicular to the gradient ∇U, satisfying $\mathbf{v}_r \cdot \nabla U = -\mathbf{v}\cdot\nabla U$ and $(\mathbf{v}_r)_r = \mathbf{v}$. In other words, the particle $\mathbf{y}$ moves along a straight line in the direction of $\mathbf{v}$ and this direction is reflected as (1.4) with probability $\Delta t[\mathbf{v}\cdot\nabla U(\mathbf{y})]_+$. This

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

probability is non-zero only if the particle is moving in a direction of lower target probability $p(\mathbf{y})$, or equivalently higher potential $U(\mathbf{y})$.

Remarkably, in the limit $\Delta t \to 0$, the algorithm leaves the joint factorized distribution $p(\mathbf{y})p(\mathbf{v})$ invariant. To see this, note that there are just two ways to reach $\mathbf{y}$ with velocity $\mathbf{v}$ at time $t + \Delta t$. The first one is by being at $\mathbf{y} - \mathbf{v}\Delta t$ at time t and moving a distance $\mathbf{v}\Delta t$ without bouncing. This occurs with probability $1 - \Delta t[\mathbf{v} \cdot \nabla U]_+$. The second possibility is that at time t the particle was at $\mathbf{y} - \mathbf{v}_r\Delta t$ with velocity $\mathbf{v}_r$, moved $\mathbf{v}_r\Delta t$ and bounced. This event occurs with probability $\Delta t[\mathbf{v}_r \cdot \nabla U]_+ = \Delta t[-\mathbf{v} \cdot \nabla U]_+$. Thus we have

$$\begin{aligned} p_{t+\Delta t}(\mathbf{y}, \mathbf{v}) &= (1 - \Delta t[\mathbf{v} \cdot \nabla U]_+)p_t(\mathbf{y} - \mathbf{v}\Delta t)p_t(\mathbf{v}) + \Delta t[-\mathbf{v} \cdot \nabla U]_+ p_t(\mathbf{y} - \mathbf{v}_r\Delta t)p_t(\mathbf{v}_r) \\ &= p_t(\mathbf{v})\left[p_t(\mathbf{y}) - \Delta t\mathbf{v} \cdot \nabla p_t(\mathbf{y}) - \Delta t(\mathbf{v} \cdot \nabla U)p_t(\mathbf{y})\right] + O(\Delta t^2) \end{aligned} \tag{1.5}$$

where we have used $p_t(\mathbf{v}) = p_t(\mathbf{v}_r)$ and

$$[\mathbf{v} \cdot \nabla U]_+ - [-\mathbf{v} \cdot \nabla U]_+ = \mathbf{v} \cdot \nabla U \tag{1.6}$$

Inserting now (1.1), the second and third terms in (1.5) cancel and we get

$$p_{t+\Delta t}(\mathbf{y}, \mathbf{v}) = p_t(\mathbf{y})p_t(\mathbf{v}) + O(\Delta t^2) \tag{1.7}$$

which implies that the distribution is stationary, $\frac{dp_t(\mathbf{y},\mathbf{v})}{dt} = 0$.

1.2 Continuous Time Limit for Integrable Distributions

Applying the transition (1.2) repeatedly and taking $\Delta t \to 0$, the random reflection point becomes an event in an inhomogeneous Poisson process with intensity $[\mathbf{v} \cdot \nabla U(\mathbf{y})]_+$. The resulting sampling procedure generates a piecewise linear Markov process [4]. To ensure ergodicity occasional resamplings are required in general [3], but not in the cases we will consider here.

The major challenge when applying the BPS algorithm is the sampling of Poisson events with intensity $[\mathbf{v} \cdot \nabla U(\mathbf{y})]_+$. In this work we consider distributions simple enough that this can be done with the inverse CDF method. In such cases, we initialize $(\mathbf{y}_0, \mathbf{v})$ and then iterate as many times as desired the following steps:

1. Sample a uniform number $u \in [0, 1]$
2. Move $\mathbf{y}$ in a straight line,

$$\mathbf{y}_k = \mathbf{y}_{k-1} + \mathbf{v}t\,, \tag{1.8}$$

   where the time t satisfies

$$u = e^{-\int_0^t dt'[\mathbf{v} \cdot \nabla U(\mathbf{y}_{k-1} + \mathbf{v}t')]_+}\,. \tag{1.9}$$

3. Reflect the velocity as $\mathbf{v} \to \mathbf{v}_r$, defined in (1.4).

2 Piecewise Continuous Distributions

The algorithm described above can be easily extended to piecewise continuous distributions. Without loss of generality, assume $U(\mathbf{y})$ is discontinuous across $y_1 = 0$ and denote by $0^\pm$ the vector $\mathbf{y}$ in both sides of $y_1 = 0$, and by $t^\pm$ the time previous and posterior to the arrival to $y_1 = 0$.

The probability distribution is preserved if a particle that reaches 0^- with $v_1 > 0$ at t^-, crosses to 0^+ with probability

$$q_{-+} = \min\left(1, e^{-U(0^+)+U(0^-)}\right), \tag{2.1}$$

and, in case of rejection, inverts the sign of v_1. Similarly, a particle reaching 0^+ with $v_1 < 0$ at t^-, should cross with probability $q_{+-} = \min\left(1, e^{-U(0^-)+U(0^+)}\right)$. Note that this is basically a Metropolis acceptance condition, with the additional rule of inverting the velocity upon rejection.

To see that this transition rule preserves $p(\mathbf{y})p(\mathbf{v})$, note that a particle at 0^- with $v_1 < 0$ at t^+ can only be the result of either i) a particle arrived at 0^- with $v_1 > 0$, tried to cross with probability q_{-+} and

was rejected, or ii) a particle arrived at 0^+ with $v_1 < 0$, and crossed successfully to 0^- with probability q_{+-} (obtained by inverting the signs in (2.1)). Considering these two possibilities, we get

$$\begin{aligned} p_{t^+}(0^-)p(v_1 < 0) &= (1 - q_{-+})p_{t^-}(0^-)p(v_1 > 0) + q_{+-}p_{t^-}(0^+)p(v_1 < 0) && (2.2)\\ &= p_{t^-}(0^-)p(v_1 < 0) && (2.3)\end{aligned}$$

since $p(v_1 > 0) = p(v_1 < 0)$ and $q_{-+}p_{t^-}(0^-) = q_{+-}p_{t^-}(0^+)$, and thus the probability is preserved. Note that this last equation is the detailed balance condition, although the BPS sampler at continuous points does not satisfy detailed balance.

The BPS algorithm, using the inverse CDF method, can be generalized to include such discontinuities in $U(\mathbf{y})$. For this we define a piecewise continuous CDF

$$w_{\mathbf{v}}(t) = 1 - e^{-\int_0^{t_1^-} dt' [\mathbf{v}\cdot\nabla U(y_0+\mathbf{v}t')]_+} q_{-+}^1 e^{-\int_{t_1^+}^{t_2^-} dt' [\mathbf{v}\cdot\nabla U(y_0+\mathbf{v}t')]_+} \dots q_{-+}^n e^{-\int_{t_n^+}^{t} dt' [\mathbf{v}\cdot\nabla U(y_0+\mathbf{v}t')]_+}, \quad (2.4)$$

with discontinuities at those times t_i where the particle encounters a positive gap at $U(\mathbf{y})$. The algorithm now initializes $(\mathbf{y}_0, \mathbf{v})$ and then iterates over the following steps:

1. Sample a uniform number $u \in [0, 1]$
2. Find

$$t = \sup\{t' \mid w_{\mathbf{v}}(t') \le u\} \quad (2.5)$$

   and move $\mathbf{y}$ in a straight line,

$$\mathbf{y}_{k+1} = \mathbf{y}_k + \mathbf{v}t\,. \quad (2.6)$$

3. If $\mathbf{y}_{k+1}$ is at a differentiable point, reflect the velocity as in (1.4). Otherwise, $t = t_i^-$ for some i, reflect $\mathbf{v}$ with respect to the discontinuity plane.

3 Binary Distributions

We consider now a distribution $p(\mathbf{s})$ over binary variables $\mathbf{s} \in \{\pm 1\}^d$. Such a distribution can be mapped into a piecewise differentiable distribution using the method of [9], which we summarize here. The idea is to augment the distribution $p(\mathbf{s})$ with continuous variables $\mathbf{y} \in \mathbb{R}^d$ distributed as

$$p_G(\mathbf{y}|\mathbf{s}) = \begin{cases} (2/\pi)^{d/2} e^{-\frac{\mathbf{y}\cdot\mathbf{y}}{2}} & \text{for } \operatorname{sign}(y_i) = s_i, \qquad i = 1, \dots, d \\ 0 & \text{otherwise}\,, \end{cases} \quad (3.1)$$

or

$$p_E(\mathbf{y}|\mathbf{s}) = \begin{cases} e^{-|\mathbf{y}|} & \text{for } \operatorname{sign}(y_i) = s_i, \qquad i = 1, \dots, d \\ 0 & \text{otherwise}\,, \end{cases} \quad (3.2)$$

where $|\mathbf{y}| = \sum_{i=1}^d |y_i|$. Considering first p_G, the joint distribution is now

$$p_G(\mathbf{s}, \mathbf{y}) = p(\mathbf{s})p_G(\mathbf{y}|\mathbf{s}) \quad (3.3)$$

We can easily marginalize over $\mathbf{s}$ and obtain

$$\begin{aligned} p_G(\mathbf{y}) &= \sum_{\mathbf{s}'} p(\mathbf{s}')p_G(\mathbf{y}|\mathbf{s}') && (3.4)\\ &\propto e^{-\frac{\mathbf{y}\cdot\mathbf{y}}{2}} p(\mathbf{s}_\mathbf{y}) && (3.5)\end{aligned}$$

where

$$\mathbf{s}_\mathbf{y} = \operatorname{sign}(\mathbf{y})\,. \quad (3.6)$$

Note that $p_G(\mathbf{y})$ is piecewise continuous, and defines a potential energy

$$\begin{aligned} U(\mathbf{y}) &= -\log p_G(\mathbf{y}) && (3.7)\\ &= \frac{\mathbf{y}\cdot\mathbf{y}}{2} - \log p(\mathbf{s}_\mathbf{y}) + const. && (3.8)\end{aligned}$$

Using $p_E(\mathbf{y})$ we obtain similarly

$$\begin{aligned} U(\mathbf{y}) &= -\log p_E(\mathbf{y}) && (3.9)\\ &= |\mathbf{y}| - \log p(\mathbf{s}_\mathbf{y}) + const. && (3.10)\end{aligned}$$

In order to sample from the original distribution $p(\mathbf{s})$, we sample from either $p_G(\mathbf{y})$ or $p_E(\mathbf{y})$ using the method of Section 2, and read out the values of $\mathbf{s}$ from (3.6).

Other distributions where this method could be applied are mixed binary and (truncated) Gaussian variables (such as the spike-and-slab regression [9]) or Bayesian Lasso models [10].

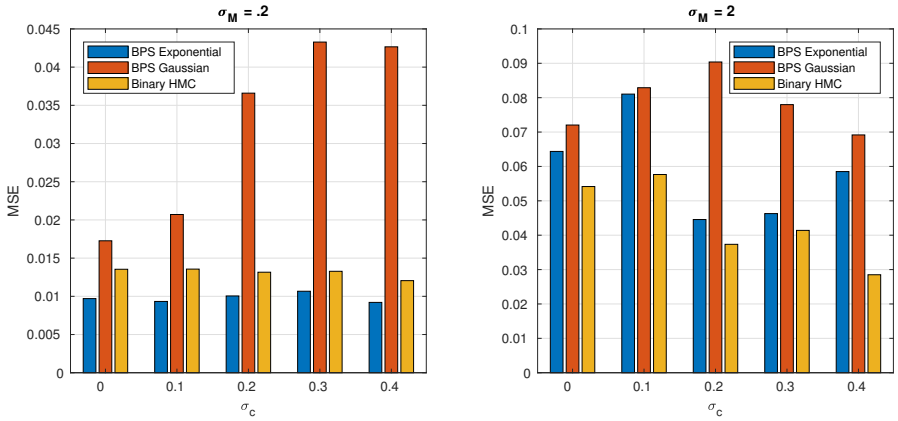


Figure 1: MSEs of $E[s_i]$ and $E[s_i, s_j]$ for $d = 10$ and different values of the standard deviations $\sigma_{\mathbf{M}}$ and $\sigma_{\mathbf{r}}$ of the coefficients in (4.1). The bars show the median of 30 runs, with the same CPU time for all samplers. The HMC travel time was $T = 6.5\pi$, but the results are similar for other Ts. In this low d regime, BPS with *exponential* augmentation dominates for easy to mix, low $\sigma_{\mathbf{M}}$ cases.

4 Example: Binary Markov Random Field

We consider distributions of the form

$$\log p(\mathbf{s}) = -\mathbf{s}^T\mathbf{r} - \frac{1}{2}\mathbf{s}^T\mathbf{M}\mathbf{s} \qquad \mathbf{s} \in \{\pm 1\}^d \tag{4.1}$$

The coefficients of $\mathbf{M}$ and $\mathbf{r}$ were sampled from zero-mean normal distributions with standard deviations $\sigma_{\mathbf{M}}$ and $\sigma_{\mathbf{r}}$. The value of $\sigma_{\mathbf{M}}$ affects the heights of the different modes and thus controls the difficulty of mixing of an MCMC sampler.

We compare the binary BPS sampler, with exponential and Gaussian augmentations, with binary HMC with Gaussian augmentation [9]. All algorithms were implemented in C++ with MATLAB wrappers.[1]

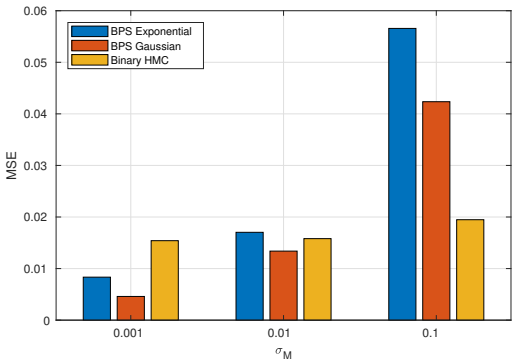
Figure 1 shows the sum of the MSEs of the $E[s_i]$ and $E[s_i, s_j]$ for $d = 10$, in easy ($\sigma_{\mathbf{M}} = .2$) and difficult ($\sigma_{\mathbf{M}} = 2$) to mix regimes, and different values of $\sigma_{\mathbf{r}}$.

For a fair comparison, all the samplers were run for the same CPU time. The results in Figure 1 show that BPS with *exponential* augmentation is the best of the three samplers for easy to mix cases ($\sigma_{\mathbf{M}} = .2$), while HMC is better for the more challenging distributions ($\sigma_{\mathbf{M}} = 2$).

Figure 2 considers the case $d = 100$ and $\mathbf{r} = 0$. In this high dimensional case, the best sampler for low σ_M is BPS with *Gaussian* augmentation, while binary HMC dominates again in the difficult, high $\sigma_{\mathbf{M}}$ regime.

To summarize, our results show that the binary BPS samplers dominate over binary HMC for easy to mix distributions. The preferred augmentation depends on the dimension: exponential for low d, Gaussian for high d. This stands in contrast to binary HMC, where the best results are obtained uniformly with the Gaussian augmentation [9].

Figure 2: MSEs of the $E[s_i]$ for $d = 100$, $\mathbf{r} = 0$, and different values of the standard deviations $\sigma_{\mathbf{M}}$ of the coefficients of $\mathbf{M}$ in (4.1). The bars show the median of 30 runs, and the travel time of each HMC iteration was tuned to $T = .5\pi$. In this high d regime, BPS with *Gaussian* augmentation dominates for easy to mix, low $\sigma_{\mathbf{M}}$ cases.



[1] Code available at https://github.com/aripakman/binary_bps.

References

[1] E.A.J.F. Peters and de With G. Rejection-free Monte Carlo sampling for general potentials. *Physical Review E*, 85(2):026703, 2012.

[2] Pierre Monmarché. Piecewise deterministic simulated annealing. *ALEA*, 13(1):357–398, 2016.

[3] Alexandre Bouchard-Côté, Sebastian J Vollmer, and Arnaud Doucet. The bouncy particle sampler: A non-reversible rejection-free Markov chain Monte Carlo method. *Journal of the American Statistical Association*, 2017.

[4] Mark HA Davis. Piecewise-deterministic Markov processes: A general class of non-diffusion stochastic models. *J. Royal Stat. Soc., Series B (Methodological)*, pages 353–388, 1984.

[5] Paul Fearnhead, Joris Bierkens, Murray Pollock, and Gareth O Roberts. Piecewise Deterministic Markov Processes for Continuous-Time Monte Carlo. *arXiv preprint arXiv:1611.07873*, 2016.

[6] Ari Pakman, Dar Gilboa, David Carlson, and Liam Paninski. Stochastic Bouncy Particle Sampler. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

[7] Joris Bierkens, Alexandre Bouchard-Côté, Arnaud Doucet, Andrew B Duncan, Paul Fearnhead, Gareth Roberts, and Sebastian J Vollmer. Piecewise Deterministic Markov Processes for Scalable Monte Carlo on Restricted Domains. *arXiv preprint arXiv:1701.04244*, 2017.

[8] Paul Vanetti, Alexandre Bouchard-Côté, George Deligiannidis, and Arnaud Doucet. Piecewise Deterministic Markov Chain Monte Carlo. *arXiv preprint arXiv:1707.05296*, 2017.

[9] Ari Pakman and Liam Paninski. Auxiliary-variable exact Hamiltonian Monte Carlo samplers for binary distributions. In *Advances in Neural Information Processing Systems*, 2013.

[10] Ari Pakman and Liam Paninski. Exact Hamiltonian Monte Carlo for Truncated Multivariate Gaussians. *Journal of Computational and Graphical Statistics*, 2014.