Efficient Parallel Discrete Sampling via Euler Method and Picard Iteration

Raj Pabari rajpabari@stanford.edu

December 11, 2023

Contents

Abstract

We simplify the framework for parallel sampling from arbitrary discrete distributions on the hypercube $\{\pm 1\}^n$ via continuous walks originally proposed by Anari et al. [3]. We propose to use Euler discretization to approximately solve the stochastic differential equation that arises in their stochastic localization analysis of the sampling problem. We then show that Picard iteration can be used to parallelize this efficiently while introducing little error.

Under a slightly weaker set of assumptions than Anari et al. [3], we show that our proposed method is in RNC for semi-log-concave distributions on $\{\pm 1\}^n$, that is, in time $\text{poly}(\log n)$ using $\text{poly}(n)$ processors. Their work did not fully resolve the reduction from approximate parallel counting to approximate parallel sampling in the case of Eulerian tours on digraphs because it guaranteed only Quasi-RNC parallel sampling (that is, with $n^{O(\log n)}$ processors), while with our weaker set of assumptions we give an RNC parallel sampling algorithm for this distribution. Thus, this work resolves the last remaining open counting to sampling reduction raised by [2].

1 Introduction

The work of Anari et al. [3] proposes an RNC algorithm for parallel sampling from arbitrary distributions μ on the hypercube $\{\pm 1\}^n$ using convolution with Gaussians. Discrete distributions on the hypercube, which are equivalent to subsets of $\{1, \ldots, n\}$, have applications to many important problems in parallel sampling, such as arborescences, determinantal point processes, and planar perfect matchings. While these problems are foundational in counting and sampling, the standard reductions from approximate counting to approximate sampling are inherently sequential. Efficient parallel algorithms were largely unknown until recent works of Anari et al. [3, 2].

1.1 Related Work

The key insight of [3] is to use techniques from stochastic localization to parallelize the reduction. Stochastic localization (see [8]) is an analysis technique that allows us to consider samples from an arbitrary disribution μ as the law of a measure-valued stochastic process $\{\mu_t\}_{t=0}^{\infty}$, where as $t \to \infty$, $\mu_t \to \delta_{x_*}$ for some random $x_* \sim \mu$.

In one of the seminal algorithmic applications of stochastic localization, [1] uses it to sample from the Sherrington-Kirpatrick model. In their algorithmic implementation, they use an Euler discretization of the stochastic differential equation (SDE), which in the case of the Sherrington-Kirpatrick model causes a large approximation error. In their work, they are unable to guarantee closeness in total variation distance, rather they obtain a much weaker notion of $o(n)$-closeness in Wasserstein distance.

Given the looseness caused by Euler discretization in [1], Anari et al. [3] avoid using Euler discretization. Instead of directly solving the SDE, they instead opt to use the randomized midpoint method of Shen and Lee [9] in their algorithm to produce approximate samples from the convolution of the exponentially tilted distribution with a Gaussian of small variance. Our main insight is proving that Euler discretization does not cause the anticipated blowup in error, and instead can be used to solve the SDE for parallel sampling on $\{\pm 1\}^n$.

Similar parallel sampling methods have been recently used in practice for sampling from continuous distributions in diffusion models in machine learning [10, 12]. Diffusion models are a class of generative machine learning models that produce outputs from a latent distribution by convolving samples with Gaussian noise in the forward process and producing samples from pure noise dictated by a stochastic (or ordinary, see [4]) differential equation in the reverse process [11]. The similarity between the sampling methods comes from the fact that, up to a change of variables and constant factors, the SDE arising in the Ornstein-Uhlenbeck process in diffusion models is equivalent to that arising in stochastic localization [8]. In fact, our results largely come from a recent theoretical analysis of diffusion models from Chen et al. [5]. We argue that sampling from discrete distributions on the hypercube is essentially a special case of the more general diffusion sampling problem analyzed by Chen et al. [5].

1.2 Organization of Paper

We will begin by introducing our definitions and assumptions, consistent with Anari et al. [3]. Then, we will present our algorithm and prove that it is in RNC, drawing on the work of Chen et al. [5].

2 Mathematical Setup

In their work, Anari et al. [3] suppose there exists an oracle for computing the logarithmic Laplace transform of μ, and observe that given this, the density and low order derivatives of $\exp(\langle\omega_i, \cdot\rangle)\mu(\cdot) * \mathcal{N}(0, \delta I)$ can be efficiently computed. From there they use the randomized midpoint method of Shen and Lee [9] to implement the approximate sampling substep of their algorithm. Accumulating these approximate samples from the convolved distributions over T timesteps yields the final sample.

The sampling method of Anari et al. [3] is inspired by stochastic localization but refrains from explicitly solving the stochastic differential equation that arises in their analysis. In our work, we do exactly this using Euler discretization for the approximate solution and Picard iteration to parallelize. Notably, we are able to guarantee RNC sampling algorithm under slightly weaker assumptions than Anari et al. [3], and thus resolve the problem of sampling random Eulerian tours in digraphs. Notably, this resolves the last remaining open question raised by Anari et al. [2] regarding the parallel sampling of distributions that admit parallel counting.

2.1 Notation, Definitions and Assumptions

For a distribution ν supported on $\mathbb{R}^n$, we'll denote $\text{mean}(\nu) := \mathbb{E}_{x\sim\nu}[x]$. Following the lead of Anari et al. [3], we will assume that we have access to an oracle for computing logarithmic Laplace transformations of distributions on the hypercube. In the remainder of this section, we will more carefully define this assumption.

Definition 1. *An exponential tilt or external field of μ supported on $\{\pm 1\}$, is an operator τ_ω such that*

$$\tau_\omega\mu(x) \propto \exp(\langle\omega, x\rangle)\mu(x)$$

In order to make this a probability distribution, we would need to normalize and divide by $\sum_x \exp(\langle\omega, x\rangle)\mu(x)$. Based on the work of [6], we introduce the following definitions –

Definition 2. *The logarithmic Laplace transform of a distribution μ on $\{\pm 1\}^n$ is*

$$\mathcal{L}_\mu(\omega) := \log\left(\sum_x \exp(\langle\omega, x\rangle)\mu(x)\right)$$

From the definiton, the following is immediate. A more detailed exposition can be found in [6].

Proposition 1. *We have $\nabla\mathcal{L}_\mu(\omega) = \text{mean}(\tau_\omega\mu)$ and $\nabla^2\mathcal{L}_\mu(\omega) = \text{cov}(\tau_\omega\mu)$.*

Definition 3. *A distribution μ on $\{\pm 1\}^n$ is β-semi-log-concave if, for all $\omega \in \mathbb{R}^n$,*

$$\nabla^2\mathcal{L}_\mu(\omega) \preceq \beta I,$$

When β is omitted, we will take this to mean that μ is $O(1)$-semi-log-concave. As an immediate consequence of the defintions, we have the following important proposition –

Proposition 2. *A distribution μ is β-semi-log-concave if and only if $\nabla \mathcal{L}_\mu$ is β-Lipschitz, eg.*

$$\|mean(\tau_\omega \mu) - mean(\tau_{\omega'} \mu)\|_2 \leq \beta \|\omega - \omega'\|_2$$

Assumptions. Throughout our analysis, on top of the assumption of having an oracle for computing the logarithmic Laplace transform L_μ, we also assume that our distribution μ on $\{\pm 1\}^n$ is semi-log-concave. See [3] for a discussion of why this is a fairly weak assumption that captures many common distributions.

3 Efficient Parallel Sampling via Euler Method and Picard Iteration

Algorithm 1: Approximate parallel sampling from μ on $\{\pm 1\}^n$ via stochastic localization

Input: Step size $\delta \in \mathbb{R}^+$, number of discretization points $N \in \mathbb{N}$, number of picard iterations $K \in \mathbb{N}$
$(h_0, h_\delta, \ldots, h_{N\delta}) \leftarrow 0$
$(g_0, g_\delta, \ldots, g_{N\delta}) \sim \mathcal{N}(0, \delta I)$
for *picard iteration* $1, \ldots, K$ **do**
  for $p = 1, \ldots, N$ **do**
    $h'_{p\delta} \leftarrow h_0 + \sum_{j=0}^{p-1} \delta \cdot \text{mean}(\tau_{h_{j\delta}} \mu) + g_{j\delta}$
  end
  $(h_0, h_\delta, \ldots, h_{N\delta}) \leftarrow (h'_0, h'_\delta \ldots, h'_{N\delta})$
end
return $\text{sign}(h_{N\delta}) \in \{\pm 1\}^n$

Our main result is Algorithm 1, an algorithmic application of stochastic localization for parallel sampling from arbitrary distributions μ on the hypercube.

3.1 Stochastic Localization

Stochastic localization is a measure-valued stochastic process $\{\mu_t\}_{t=0}^\infty$ dictated by a stochastic differential equation (SDE). For a distribution μ defined on a subset of $\mathbb{R}^n$ and a matrix process C_t, we have for all x the SDE

$$d\mu_t(x) = \langle x - \text{mean}(\mu_t), C_t dB_t \rangle \mu_t(x) \tag{1}$$

for B_t a Brownian motion. The process satisfies the property that as $t \to \infty, \mu_t \to \delta_{x_*}$ for a random x_* distributed according to μ. Additionally, it is a martingale. Notably, the μ_t need not be supported on the same space as μ. See more details in [8]. We have that for all t,

$$\mu_t(x) \propto \mu(x) \cdot \exp\left(-\frac{t\|x\|_2^2}{2} + \langle h_t, x \rangle\right), \quad \text{where } dh_t = \text{mean}(\mu_t)dt + dB_t \quad (h_0 = 0) \tag{2}$$

This relation is a result of standard stochastic calculus techniques, a detailed detivation can be found in [8, 7]. With this definition of h_t, the μ_t represent the posterior distribution of μ given the value of x_t. Furthermore, because the $x \in \{\pm 1\}^n$, their norm is constant. Thus, we can simplify the relation to

$$\mu_t(x) \propto \mu(x) \cdot \exp\left(\langle h_t, x \rangle\right) = \tau_{h_t}\mu \implies \text{mean}(\mu_t) = \text{mean}(\tau_{h_t}\mu)$$

We've thus reduced the problem of sampling from μ to the problem of solving the SDE in Equation 2 for h_t.

To solve this SDE, we apply standard techniques, using Euler discretization to find an approximate solution and Picard iteration to parallelize the solution. The key result in our proof comes from the theory of diffusion models, in which this specific SDE is well-studied. In fact, the isotropic Gaussian stochastic localization process (as outlined above) is equivalent to that used in diffusion models, up to a change of variables and constants [8].

3.2 Discretization

Euler discretization is a method of approximating solutions to SDEs up to arbitrary precision. For SDE $dX_t = f(t)dt + dB_t$, the simplest version of the Euler discretization is as follows for some step size Δ, $b_t \overset{\text{iid}}{\sim} \mathcal{N}(0, I)$ –

$$X_{t+\Delta} \approx X_t + \nabla f(t)\Delta + b_{t+\Delta}\sqrt{\Delta}$$

As $\Delta \to 0$, this discretization faithfully approximates the true solution to the SDE. Our Euler discretization of 2 then follows immediately for interval size δ. Letting $b_t \overset{\text{iid}}{\sim} \mathcal{N}(0, I)$ and $g_t \overset{\text{iid}}{\sim} \mathcal{N}(0, \delta I)$, we have

$$h_{t+\delta} = h_t + \delta \cdot \text{mean}(\mu_t) + \sqrt{\delta} b_t = h_t + \delta \cdot \text{mean}(\tau_{h_t}\mu) + g_t \implies h_{\ell\delta} = h_0 + \sum_{j=0}^{\ell-1} \delta \cdot \text{mean}(\tau_{h_j}\mu) + g_{j\delta} \tag{3}$$

Equation 3 should be familiar from Algorithm 1. From the field of diffusion models, Chen et al. [5] characterizes a more general case of the error introduced by Euler discretization. In their paper, Chen et al. tackle the problem of sampling from an unknown distribution q, which they call the score function. Because the distribution is entirely unknown, they use a neural network to estimate the score function up to some precision $\varepsilon_{\text{score}}$, which incurs some error. This term of the error does not appear in our context.

Furthermore, in their denoising/reverse process, [5] initializes from pure Gaussian noise, because they do not have access to the law of the noising/forward process after termination. However, in the context of stochastic localization, the SDE in equation 2 is not derived as a result of a time reversal of a forward process, so we do not incur an initialization error. See that initializing at $h_0 = 0$ gives $\mu_0 \propto \tau_0\mu = \mu$, which is exactly as desired. Thus, we need not consider the KL divergence term.

Given this analysis, we can directly adapt the bound of [5] into a bound on Euler discretization of our SDE. Fix a step size δ, let $N := T/\delta$ be the total number of timesteps, and let $\text{dist}(\text{sign}(h_T)) = \text{dist}(\text{sign}(h_{N\delta}))$ denote the law of the Euler discretization of the SDE as in Equation 3. Recalling from Proposition 2 that for a β-semi-log-concave distribution μ, $\nabla\mathcal{L}_\mu$ is β-Lipschitz, we have from [5] that

$$d_{TV}(\text{dist}(\text{sign}(h_T)), \mu) \leq \left(\beta\sqrt{n\delta} + \beta\delta\mathbb{E}_{x\sim\mu}[\|x\|_2^2]\right)\sqrt{T}$$

Technically, this bound applies to the distribution of $\text{sign}(h_T/T)$, but it should be clear that this distribution is equal to that of $\text{sign}(\mu_T)$. Furthermore, $\mathbb{E}_{x\sim\mu}[\|x\|_2^2] = n$ because we are on the hypercube, thus we have

$$d_{TV}(\text{dist}(\text{sign}(h_T)), \mu) \leq \left(\beta\sqrt{n\delta} + \beta n\delta\right)\sqrt{T}$$

Thus, setting $T = O\left(\frac{\varepsilon^2}{n^2\beta^2}\right)$, $\delta = O\left(\frac{1}{n^2}\right)$, and $N = O\left(\frac{\varepsilon^2}{\beta^2}\right)$ gives a total variation distance bounded by $O(\varepsilon)$. Recalling that for semi-log-concave distributions, we have $\beta = O(1)$, we can furthermore ignore factors of β in the preceding bounds (importantly, $N = O(1)$).

3.3 Parallelization

We now analyze a parallelization technique for the Euler discretization introduced in the preceding section, this being Picard iteration. Sequential computation of the Euler discretization of our SDE in Equation 3 would yield a trajectory $h^* = (h_0^*, h_\delta^*, \ldots, h_T^*)$. In our Picard iteration, we initialize the trajectory at iteration 0, then compute the entire trajectory in parallel according to Equation 3 for each subsequent iteration using the values of $h_{j\delta}$ that we computed in the previous iteration. For a more precise description, see Algorithm 1 or [10].

Let $h_t^{(k)} = (h_0^{(k)}, h_\delta^{(k)}, \ldots, h_T^{(k)})$ denote the trajectory after k Picard iterations. It should be clear by construction that $(h_0^{(k)}, h_\delta^{(k)}, \ldots, h_{k\delta}^{(k)}) = (h_0^*, h_\delta^*, \ldots, h_{k\delta}^*)$. This implies a trivial bound of $N = O(1)$ iterations until obtaining the sequential answer h^*. Importantly, notice that after reaching h^* after N iterations, subsequent Picard iterations will still output h^*, thus we do indeed have convergence.

In order to evaluate Equation 3, we'll need to find the mean of arbitrary exponential tilts of μ; as an intermediate step in the proof of Lemma 6 in Anari et al. [3], it is shown that this can be done with $n^{O(1)}$ processors and $(\log n)^{O(1)}$ time per processor given an oracle for $\mathcal{L}_\mu$. In the worst case, one processor will need to sum over the mean of $N = O(1)$ exponential tilts of μ, however this still maintains a total of $(\log n)^{O(1)}$ time per Picard iteration. Recalling that we need at most $N = O(1)$ Picard iterations to converge, and the convergence is exact convergence to the sequential h^*, this finally implies that Algorithm 1 is in RNC by setting $K = N$.

4 Conclusion

We've shown that, given semi-log-concave distribution μ on the hypercube and an oracle for computing its logarithmic Laplace transforms, we can sample from it in RNC time complexity. Anari et al. [3] show that many important classes of distributions satisfy these assumptions, such as arborescences, determinantal point processes, Eulerian tours on digraphs, and planar perfect matchings. Importantly, per the analysis of Anari et al. [3], this work gives an RNC algorithm for sampling Eulerian tours on digraphs, the last remaining open reduction from approximate parallel counting to approximate parallel sampling raised by Anari et al. [2]. Thus, using techniques from stochastic localization and insights from diffusion models, this work resolves the last remaining open question in this regard. Given the success of algorithmic stochastic localization in this work and [1], future directions may consider algorithmic stochastic localization for sampling from other distributions. This work shows that using standard techniques (Euler discretization, Picard iteration) to solve the arising SDEs can yield efficient parallel sampling algorithms. With more sophisticated numerical methods for solving SDEs (eg. implicit Euler, Runge-Kutta), we may be able to obtain tighter bounds; this is another interesting direction for future work.

References

[1] Ahmed El Alaoui, Andrea Montanari, and Mark Sellke. Sampling from the sherrington-kirkpatrick gibbs measure via algorithmic stochastic localization, 2022.

[2] Nima Anari, Nathan Hu, Amin Saberi, and Aaron Schild. Sampling arborescences in parallel. *CoRR*, abs/2012.09502, 2020.

[3] Nima Anari, Yizhi Huang, Tianyu Liu, Thuy-Duong Vuong, Brian Xu, and Katherine Yu. Parallel discrete sampling via continuous walks. STOC 2023, pages 103–116, New York, NY, USA, 2023. Association for Computing Machinery.

[4] Sitan Chen, Sinho Chewi, Holden Lee, Yuanzhi Li, Jianfeng Lu, and Adil Salim. The probability flow ode is provably fast, 2023.

[5] Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru R. Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions, 2023.

[6] Ronen Eldan and Omer Shamir. Log concavity and concentration of lipschitz functions on the boolean hypercube. *Journal of Functional Analysis*, 282(8):109392, 2022.

[7] Robert S. Liptser and Albert N. Shiryaev. *Statistics of Random Processes I: General Theory*, volume 394. Springer, 1977.

[8] Andrea Montanari. Sampling, diffusions, and stochastic localization, 2023.

[9] Ruoqi Shen and Yin Tat Lee. The randomized midpoint method for log-concave sampling. *CoRR*, abs/1909.05503, 2019.

[10] Andy Shih, Suneel Belkhale, Stefano Ermon, Dorsa Sadigh, and Nima Anari. Parallel sampling of diffusion models, 2023.

[11] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *CoRR*, abs/2011.13456, 2020.

[12] Linqi Zhou, Andy Shih, Chenlin Meng, and Stefano Ermon. Dreampropeller: Supercharge text-to-3d generation with parallel sampling, 2023.