Proposal of a Score Based Approach to Sampling Using Monte Carlo Estimation of Score and Oracle Access to Target Density

Curtis J. McDonald
Department of Statistics and Data Science
Yale University
curtis.mcdonald@yale.edu

Andrew Barron
Department of Statistics and Data Science
Yale University
andrew.barron@yale.edu

Abstract

Score based approaches to sampling have shown much success as a generative algorithm to produce new samples from a target density given a pool of initial samples. In this work, we consider if we have no initial samples from the target density, but rather 0^{th} and 1^{st} order oracle access to the log likelihood. Such problems may arise in Bayesian posterior sampling, or in approximate minimization of non-convex functions. Using this knowledge alone, we propose a Monte Carlo method to estimate the score empirically as a particular expectation of a random variable. Using this estimator, we can then run a discrete version of the backward flow SDE to produce samples from the target density. This approach has the benefit of not relying on a pool of initial samples from the target density, and it does not rely on a neural network or other black box model to estimate the score.

1 Introduction

Producing samples from a known probability measure is a core problem in statistics and many machine learning applications. Computing complicated integrals, determining the volume of a high dimensional set, or performing Bayesian inference on a posterior are all problems which require generating samples from a probability measure in question. In this work, we consider sampling from a probability measure P_0 over $\theta \in \mathbb{R}^d$ absolutely continuous with respect to Lebesgue measure and admitting a probability density proportional to $p_0(\theta) \propto e^{f(\theta)-\frac{1}{2}\theta^T\theta}$ for some function $f(\theta)$. The form of the density is known and we are able to query $f(\theta)$ and $\nabla f(\theta)$ at will.

When $f(\theta)$ is concave, such a density is called log concave and can easily be sampled by known methods such as Langevin diffusion or other classical Markov Chain Monte Carlo (MCMC) methods. However, we are particularly interested in when $f(\theta)$ is non-concave having multiple separated maximizers. Traditional MCMC methods for such a density have no guarantee of convergence to invariance in finite time. Two examples are if $f(\theta)$ represents the log likelihood of a mixture model (e.g. Gaussian mixture model) and $p_0(\theta)$ is the posterior under a normal prior. Under such a model, the posterior has multiple modes of equal likelihood which are difficult for traditional methods to sample. If $f(\theta)$ represents a non-concave objective we want to maximize under a L^2 penalty, $\theta^* = \text{argmax}_\theta f(\theta) - \frac{1}{2}\theta^T\theta$, we may settle not for finding θ^*, but rather sampling $\theta \sim p_0(\theta)$ and anticipating that $f(\theta^*) \approx f(\theta)$, that is our sampled point, while not the global maximizer, produces a function value close to the optimal value.

Thus, there remains non-log concave sampling problems yet to be sufficiently addressed by existing MCMC algorithms. In recent years, score based methods have shown remarkable success as a generative model on complex multi-modal densities in high dimensions. In this paper, we would like to transfer this success as a generative model into a successful algorithm for sampling a known

NeurIPS 2022 Workshop on Score-Based Methods.

density function. The main difference between a generative modelling problem and the sampling problem studied here is what "information" about the target density $p_0(\theta)$ the algorithm may access. In the current applications of score based methods as a generative algorithm, one starts with an initial pool of samples drawn from the target measure, $\theta_0, \cdots, \theta_N \sim P_0$, but one does not know the functional form of the target density nor can one query $f(\theta)$ and it's gradient $\nabla f(\theta)$ directly. In this work, we have no initial samples from the target density, but do have direct axis to the density function $p_0 = e^{f(\theta) - \frac{1}{2}\theta^T \theta}$ as well as access to $f(\theta)$ and $\nabla f(\theta)$.

What connects both problems is the core theory of score based sampling. Underlying all score based approaches to sampling is the dual relationship between the "forward flow" and "backward flow" stochastic differential equations (SDE). Consider the forward flow SDE

$$d\theta_t = f(\theta_t, t)dt + g(t)dW_t, \theta_0 \sim P_0, \tag{1}$$

where the marginal distribution of θ_t at time t is denoted P_t with density p_t. Paired with this forward flow SDE is a backward flow SDE [1]

$$d\theta_t = [f(\theta_t, t) - g(t)^2 \nabla \log p_t(\theta_t)]dt + g(t)dW_t, \tag{2}$$

where we run time in reverse from $t = T$ to $t = 0$. The forward flow SDE takes $\theta_0 \sim P_0$ to $\theta_T \sim P_T$, while the backward flow SDE takes $\theta_T \sim P_T$ to $\theta_0 \sim P_0$. If we can sample P_T directly and implement the backward flow SDE then we can produce samples from P_0.

Crucial to the backward flow SDE is the score function, $\nabla \log p_t(\theta)$, which is not known and must be estimated. Popular generative approaches have a large pool of initial samples drawn from P_0. They then train a model such as a neural network to approximate the score at different time scales t by minimizing an objective function using the initial samples and noisy perturbations as training data [10, 11]. Various related methods such as [3], [4] propose improved methods to estimate the score, but are also ultimately generative and thus rely on initial samples. [5] applies score based diffusions as an approach to improve annealed importance sampling. It is not a generative approach as it does not rely on initial samples from the target density, but it ultimately trains a neural network to estimate the score by minimizing relative entropy. In this paper, we desire a method to estimate the score without any initial samples from P_0 and without appealing to a complex secondary model such as a neural network to estimate the score.

As a summary of what is to come, with knowledge of the functional form of the density p_0 and application of stochastic calculus, we can analytically express how p_t changes over time and express p_t as an expectation of a known function over a known density. With some further analysis, the score $\nabla \log p_t(\theta)$ can also be expressed as the expectation of a function over a density, and thus the score can be computed by estimating this integral via a sub Monte Carlo (MC) sampling problem. This replaces the need to estimate the score via a neural network with estimating the score via a MC average.

Specifically, we will consider the forward flow SDE as the Ornstein–Uhlenbeck process with drift $f(\theta, t) = -\theta$ and constant diffusion $g(t) = \sqrt{2}$

$$d\theta_t = -\theta_t dt + \sqrt{2} dW_t. \tag{3}$$

The invariant measure is standard normal and the conditional measure $p_t(\theta_t|\theta_0)$ can be expressed as

$$\theta_t = e^{-t}\theta_0 + \sqrt{1 - e^{-2t}} Z, Z \sim N(0, I). \tag{4}$$

Thus the density $p_t(\theta_t)$ up to normalization can be computed by integrating $p_0(\theta_0)p_t(\theta_t|\theta_0)$ over θ_0 and the score, which removes any dependence on normalizing constants, can be expressed as

$$\nabla_{\theta_t} \log p_t(\theta_t) = \frac{\int p_0(\theta_0) \nabla_{\theta_t} p_t(\theta_t|\theta_0) d\theta_0}{\int p_0(\theta_0) p_t(\theta_t|\theta_0) d\theta_0}. \tag{5}$$

With some manipulations, we can express the numerator and denominator as expectations over normal random variables, and thus can approximate each via averaging the integrand at normally drawn points, i.e. a MC estimator.

2 Results

Simplifying equation (5) (see appendix for full derivation) we can express the score as a ratio of two expectations,

$$\nabla \log p_t(\theta) = -\theta + \frac{e^{-t}}{\sqrt{1-e^{-2t}}} \frac{E[Ue^{f(\sqrt{1-e^{-2t}}U + e^{-t}\theta)}]}{E[e^{f(\sqrt{1-e^{-2t}}U+e^{-t}\theta)}]} \quad U \sim N(0, I), \tag{6}$$

applying integration by parts (see appendix), this expression can equivalently be written as

$$\nabla \log p_t(\theta) = -\theta + e^{-t} \frac{E[\nabla f(\sqrt{1-e^{-2t}}U + e^{-t}\theta)e^{f(\sqrt{1-e^{-2t}}U+e^{-t}\theta)}]}{E[e^{f(\sqrt{1-e^{-2t}}U+e^{-t}\theta)}]} \quad U \sim N(0, I). \tag{7}$$

Computing expressions (6) and (7) then amounts to computing expectations over normal random variables, and we will appeal to a MC estimator. Say we draw K points $U_1, \cdots, U_K \sim N(0, I)$ from a standard normal, and approximate the score in one of two ways. Define weights w_k which sum to 1

$$w_k = \frac{e^{f(\sqrt{1-e^{-2t}}U_k+e^{-t}\theta)}}{\sum_{j=1}^{K} e^{f(\sqrt{1-e^{-2t}}U_j+e^{-t}\theta)}}, \tag{8}$$

as the relative heights of the exponential of f evaluated at the sampled points. We can then approximate (6) and (7) as weighted averages using these weightings

$$\hat{s}_1(\theta, t) = -\theta + \frac{e^{-t}}{\sqrt{1-e^{-2t}}} \sum_{k=1}^{K} U_k w_k \tag{9}$$

$$\hat{s}_2(\theta, t) = -\theta + e^{-t} \sum_{k=1}^{K} \nabla f(\sqrt{1-e^{-2t}}U_k + e^{-t}\theta) w_k. \tag{10}$$

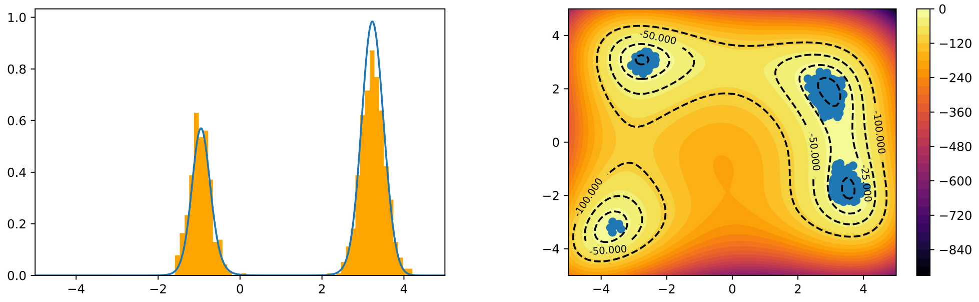
There are two perspectives on how to estimate the ratio of expectations that define (6) and (7). One could use a separate pool of samples to evaluate the numerator and denominator expectations and then take the ratio of these two estimates. However, we may be dividing by a potentially small number in this method and have high variance as a result. The authors here propose using the same pool of samples to estimate both the numerator and denominator, resulting in the weights w_k where the normalizing constant of the weights represents the denominator in (6) and (7). This has the advantage that the estimates will never become unbounded since all estimates are weighted averages of well behaved values, however proper analysis must be conducted on how such a sampling procedure effects the independence, bias, and variance of the estimators.

For small t, say $t < 0.1$, we conjecture that $\hat{s}_2(\theta, t)$ is the better estimator as it will have lower variance since $\frac{e^{-t}}{\sqrt{1-e^{-2t}}}$ can grow large as $t \to 0$. Additionally, as $t \to 0$, $\nabla f(\sqrt{1-e^{-2t}}U + e^{-t}\theta) \to \nabla f(\theta)$ and estimate $\hat{s}_2(\theta, t)$ approaches $-\theta + \nabla f(\theta)$ for small t, that is the score of the target density.

For larger t, it would seem $\hat{s}_1(\theta, t)$ will have lower variance since U_k is less variable than the gradient of the target $\nabla f(U_k)$. Additionally, as $t \to \infty$ we have the weights w_k approach a θ independent value due to the $e^{-t}\theta$, so the expectation is some finite vector while $\frac{e^{-t}}{\sqrt{1-e^{-2t}}} \to 0$ thus the score approaches $-\theta$, the score of a standard normal.

The algorithm to generate a sample from P_0 is as follows. Pick a terminal time $T > 0$, step size $\delta > 0$, and MC sample size $K \in \mathbb{N}$. Initialize $\theta_T \sim N(0, I)$ and $t = T$. While $t > 0$, in each iteration draw K values $U_1, \cdots, U_K \sim N(0, I)$ and compute weights w_k. For large times $t > 0.1$ estimate score as $\hat{s}(\theta_t, t) = \hat{s}_1(\theta, t)$, otherwise set $\hat{s}(\theta_t, t) = \hat{s}_2(\theta, t)$. Update the point as $\theta_{t-\delta} = \theta_t - \delta(-\theta_t - 2\hat{s}(\theta_t, t)) + \sqrt{2\delta}Z, \quad Z \sim N(0, I)$ and decrease $t = t - \delta$. Finally, return θ_0.

One could propose more complex methods such Stein's method [2],[9] or multi-modal sampling methods such as Simulated Tempering [7] to estimate the quantities in (6) and (7), as these represent expectations over un-normalized densities (the denominator is the normalizing constant). However, estimating these values is itself a sub problem of sampling the un-normalized density p_0 in the first place, thus if we employ sufficiently complex methods to sample the sub problems we may have simply employed said methods on the original problem. Additionally, for large $t >> 0$ and small $t \approx 0$ the simple estimators will have small variance, thus it is the middle region of t where it may require more involved estimation of quantities (6) and (7) to keep the estimates accurate.



(a) One Dimensional Sampling Histogram

(b) Himmelblau Sampled Points

Figure 1: Samples produced for 1 and 2 dimensional sampling problems.

3 Example

Here we demonstrate the performance of the algorithm on a series of example problems. These examples while low dimensional represent multi-modal targets with future work considering higher dimensional implementations. First, we sample a one dimensional multi-modal target density with asymmetric modes. Using the log likelihood function

$$f(\theta) = 100\sum_{i=1}^{4}(\tanh(\theta + 0.05 - \mu_i) - \tanh(\theta - 0.05 - \mu_i)), \mu_1 = -5, \mu_2 = -1, \mu_3 = 3, \mu_4 = 4.$$

In Figure 1a we see the histogram of $n = 1000$ points ran in parallel with terminal time $T = 2$, $K = 3000$ samples per estimate, and step size $\delta = 0.01$. This requires $K * (T/\delta) = 600,000$ evaluations of f and ∇f per point, a significantly high number of evaluations per sample but the polynomial nature may see better gains in high dimensions compared with existing methods.

Another typical problem to test multi-modal sampling is to sample the Himmelblau function [6] in two dimensions

$$f(\theta_1, \theta_2) = -(\theta_1^2 + \theta_2 - 11)^2 - (\theta_1 + \theta_2^2 - 7)^2$$

which has modes at $(3, 2), (-2.81, 3.13), (-3.78, -3.28), (3.58, -1.85)$ which are all separated by areas of low probability and asymmetric in their relative heights. Sampling $n = 2000$ points in parallel with $T = 3$, $K = 1000$, $\delta = 0.01$ this algorithm requires $K * (T/\delta) = 300,000$ evaluations of f and ∇f per point. To test if each mode is sampled proportionally correct, we compute the proportion of points within ± 0.5 in each coordinate around each mode and normalize to sum to 1. We then evaluate the target pdf at $10,000$ points sampled uniformly around each mode as an estimate of the probability of the region around each mode and normalize to sum to 1. If each more is sampled proportionally correctly, then these two vectors should be the same. The probability estimates of each mode's immediate vicinity are (0.831, 0.009, 0.002, 0.158) while the proportion of points near each mode is (0.741, 0.053, 0.005, 0.201). We note that the modes are not sampled exactly in the correct proportions, but the proportions are reasonable close to each mode, and each mode is visited, so as an approximate sampling algorithm this may suffice in certain situations.

4 Concluding remarks

This paper presents a score based sampling approach using Monte Carlo estimation of the score and knowledge of the functional form of the target density. This method does not rely on any initial samples from the target density, nor on a model such as a neural network to estimate the score function. Much work remains to be done on the the analysis of such a method, specifically on high dimensional target densities of interest. The MC estimators may be of high variance, and theoretical guarantees on this variance must be provided. Furthermore, the estimates of the score will never be perfect, they will have some standard error around the true value, thus analysis of the backward flow SDE under discretization and approximate drift such as the analysis in [8] will be also be required to guarantee convergence to the correct target measure.

References

[1] Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

[2] Wilson Ye Chen, Alessandro Barp, François-Xavier Briol, Jackson Gorham, Mark Girolami, Lester Mackey, and Chris Oates. Stein point markov chain monte carlo. In *International Conference on Machine Learning*, pages 1011–1021. PMLR, 2019.

[3] Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*, 34:17695–17709, 2021.

[4] Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Score-Based Generative Modeling with Critically-Damped Langevin Diffusion. *International Conference on Learning Representations (ICLR)*, 2022.

[5] Arnaud Doucet, Will Grathwohl, Alexander GDG Matthews, and Heiko Strathmann. Score-based diffusion meets annealed importance sampling. *arXiv preprint arXiv:2208.07698*, 2022.

[6] Farhan Feroz and John Skilling. Exploring multi-modal distributions with nested sampling. In *AIP Conference Proceedings*, volume 1553, pages 106–113. American Institute of Physics, 2013.

[7] Rong Ge, Holden Lee, and Andrej Risteski. Simulated tempering langevin monte carlo ii: An improved proof using soft markov chain decomposition. *arXiv preprint arXiv:1812.00793*, 2018.

[8] Holden Lee, Jianfeng Lu, and Yixin Tan. Convergence for score-based generative modeling with polynomial complexity. *arXiv preprint arXiv:2206.06227*, 2022.

[9] Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. *Advances in neural information processing systems*, 29, 2016.

[10] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019.

[11] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *International Conference on Learning Representations (ICLR)*, 2020.

5 Appendix

Here we derive the form of the expectations that define the score. Define $\sigma_t = \sqrt{1-e^{-2t}}$, we have

$$\begin{aligned}
\nabla_{\theta_t} \log p_t(\theta_t) &= \frac{\int p_0(\theta_0)\nabla_{\theta_t} p_t(\theta_t|\theta_0)d\theta_0}{\int p_0(\theta_0)p_t(\theta_t|\theta_0)d\theta_0} \\
&= \frac{\int e^{f(\theta_0)-\frac{1}{2}\theta_0^T\theta_0}\nabla_{\theta_t} e^{-\frac{1}{2\sigma_t^2}(\theta_t-e^{-t}\theta_0)^T(\theta_t-e^{-t}\theta_0)}d\theta_0}{\int e^{f(\theta_0)-\frac{1}{2}\theta_0^T\theta_0} e^{-\frac{1}{2\sigma_t^2}(\theta_t-e^{-t}\theta_0)^T(\theta_t-e^{-t}\theta_0)}d\theta_0} \\
&= \frac{\int e^{f(\theta_0)}\left(-\frac{\theta_t-e^{-t}\theta_0}{\sigma_t^2}\right) e^{-\frac{1}{2}\theta_0^T\theta_0-\frac{1}{2\sigma_t^2}(\theta_t-e^{-t}\theta_0)^T(\theta_t-e^{-t}\theta_0)}d\theta_0}{\int e^{f(\theta_0)} e^{-\frac{1}{2}\theta_0^T\theta_0-\frac{1}{2\sigma_t^2}(\theta_t-e^{-t}\theta_0)^T(\theta_t-e^{-t}\theta_0)}d\theta_0}
\end{aligned}$$

we now perform some completing the square on the θ_0 terms in the exponent to express it as a normal density over θ_0. Note by cancellation on the top and bottom of the fraction, we can ignore any constant terms in the integral and only focus on those terms which relate to θ_0.

$$
\begin{aligned}
&e^{f(\theta_0)-\frac{1}{2}\theta_0^T\theta_0}e^{-\frac{1}{2\sigma_t^2}(\theta_t-e^{-t}\theta_0)^T(\theta_t-e^{-t}\theta_0)} \\
&= e^{f(\theta_0)}e^{-\frac{1}{2}[\theta_0^T\theta_0+\frac{1}{\sigma_t^2}(\theta_t^T\theta_t-2e^{-t}\theta_t^T\theta_0+e^{-2t}\theta_0^T\theta_0)]} \\
&\propto e^{f(\theta_0)}e^{-\frac{1}{2}((1+\frac{e^{-2t}}{1-e^{-2t}})\theta_0^T\theta_0-2\frac{e^{-t}}{1-e^{2t}}\theta_t^T\theta_0)} \\
&\propto e^{f(\theta_0)}e^{-\frac{1}{2\sigma_t^2}(\theta_0-e^{-t}\theta_t)^T(\theta_0-e^{-t}\theta_t)}
\end{aligned}
$$

thus, the above function is proportional to $e^{f(\theta_0)}N(e^{-t}\theta_t, \sigma_t^2 I)$ and we can write the score as

$$
\frac{-E[e^{f(\theta_0)}\frac{\theta_t-e^{-t}\theta_0}{\sigma_t^2}]}{E[e^{f(\theta_0)}]} \quad \theta_0 \sim N(e^{-t}\theta_t, \sigma_t^2 I)
$$

then, consider if instead $U \sim N(0, I)$ and $\theta_0 = \sigma_t U + e^{-t}\theta_t$, we could write the above as

$$
\begin{aligned}
&\frac{-E[e^{f(\sigma_t U+e^{-t}\theta_t)}\frac{\theta_t-e^{-t}\sigma_t U-e^{-2t}\theta_t}{1-e^{-2t}}]}{E[e^{f(\sigma_t U+e^{-t}\theta_t)}]} \\
&= \frac{-E[e^{f(\sigma_t U+e^{-t}\theta_t)}(\theta_t - \frac{e^{-t}}{\sigma_t}U)]}{E[e^{f(\sigma_t U+e^{-t}\theta_t)}]} \\
&= -\theta_t + \frac{e^{-t}}{\sigma_t}\frac{E[Ue^{f(\sigma_t U+e^{-t}\theta_t)}]}{E[e^{f(\sigma_t U+e^{-t}\theta_t)}]}
\end{aligned}
$$

We can also apply integration by parts in the numerator to arrive at an equivalent expression,

$$
\begin{aligned}
\frac{e^{-t}}{\sigma_t}E[Ue^{f(\sigma_t U+e^{-t}\theta_t)}] &= \frac{e^{-t}}{\sigma_t}\int ue^{f(\sigma_t u+e^{-t}\theta_t)}\left(\frac{1}{2\pi}\right)^{\frac{d}{2}}e^{-\frac{1}{2}u^Tu}du \\
&= -\frac{e^{-t}}{\sigma_t}\int e^{f(\sigma_t u+e^{-t}\theta_t)}\left(-u\left(\frac{1}{2\pi}\right)^{\frac{d}{2}}e^{-\frac{1}{2}u^Tu}\right)du \\
&= -\frac{e^{-t}}{\sigma_t}\int e^{f(\sigma_t u+e^{-t}\theta_t)}\nabla_u\left(\left(\frac{1}{2\pi}\right)^{\frac{d}{2}}e^{-\frac{1}{2}u^Tu}\right)du \\
&= \frac{e^{-t}}{\sigma_t}\int \nabla_u\left(e^{f(\sigma_t u+e^{-t}\theta_t)}\right)\left(\frac{1}{2\pi}\right)^{\frac{d}{2}}e^{-\frac{1}{2}u^Tu}du \\
&= \frac{e^{-t}}{\sigma_t}\int \sigma_t\nabla f(\sigma_t u+e^{-t}\theta_t)e^{f(\sigma_t u+e^{-t}\theta_t)}\left(\frac{1}{2\pi}\right)^{\frac{d}{2}}e^{-\frac{1}{2}u^Tu}du \\
&= e^{-t}E[\nabla f(\sigma_t U+e^{-t}\theta_t)e^{f(\sigma_t U+e^{-t}\theta_t)}]
\end{aligned}
$$

and thus can express the score as

$$
-\theta_t + e^{-t}\frac{E[\nabla f(\sigma_t U+e^{-t}\theta_t)e^{f(\sigma_t U+e^{-t}\theta_t)}]}{E[e^{f(\sigma_t U+e^{-t}\theta_t)}]}
$$