

Lecture 20: Langevin Monte Carlo

Gautam Iyer (*scribe: Keenan Crane*)

November 9, 2023

Since we keep running out of time, we might be a little less ambitious today.

1 Recap

Let's just recall the SDE

$$dX_t = b(X_t)dt + \sigma(X_t)dW_t, \quad (1)$$

where everything is d -dimensional, i.e., $b : \mathbb{R}^d \rightarrow \mathbb{R}^d$ and $\sigma : \mathbb{R}^d \rightarrow \mathbb{R}^{d^2}$, where $\mathbb{R}^{d^2}$ denotes $d \times d$ matrices, and W_t is d -dimensional Brownian motion. Recall that X_t^x denotes the solution of our SDE for initial data x ; for each fixed t , this solution is an $\mathbb{R}^d$ -valued random variable, with associated probability density or *transition density* $\phi_t(x, y)$. What does this probability density tell us? It gives us, for any subset $A \subset \mathbb{R}^d$ the probability

$$P(X_t^x \in A) = \int_A \phi_t(x, y) dy.$$

We also defined a differential operator

$$\mathcal{L}f(x) = \sum_{i=1}^d b_i(x) \partial_i f(x) + \sum_{i=1}^d \sum_{j=1}^d a_{ij} \partial_i \partial_j f(x),$$

where

$$a_{ij} := \frac{1}{2} \sum_{k=1}^d \sigma_{ik} \sigma_{jk}.$$

The adjoint of this operator is given by

$$\mathcal{L}^*g(x) = - \sum_{i=1}^d \partial_i (b_i(x)g(x)) + \sum_{i=1}^d \sum_{j=1}^d \partial_i \partial_j (a_{ij}g(x)).$$

We then had two theorems.

Theorem (Kolmogorov equations). Consider the SDE defined in Equation 1. Then

1. in variables t, x , we have $\partial_t \phi = \mathcal{L}_x \phi$ (Kolmogorov backward equation), and
2. in variables t, y , we have $\partial_t \phi = \mathcal{L}_x^* \phi$ (Kolmogorov forward equation).

2 Motivation

Recall that our goal is to sample from some distribution, with density ρ . One strategy, in principle, is to solve the SDE (1), where we choose b and σ “cleverly” so that if we look at the distribution of X_t , then it goes to the target distribution ρ as $t \rightarrow \infty$. But how do we find such data?

To try to answer this question, let’s consider $\lim_{t \rightarrow \infty} \phi_t(x, y)$. Suppose in particular that

$$\lim_{t \rightarrow \infty} \phi_t(x, y) = \rho(y),$$

i.e., that the probability density function of our process approaches the target density ρ , as desired. More explicitly, we mean that

$$\lim_{t \rightarrow \infty} P(X_t^x \in A) = \lim_{t \rightarrow \infty} \int_A \phi_t(x, y) dy = \int_A \rho(y) dy$$

for all A . We don’t have a formula for ρ , but suppose that we do have an equation for ρ —like the Kolmogorov forward equation. Suppose moreover that this equation reaches a stationary state, i.e., that as $t \rightarrow \infty$, $\partial_t \phi = 0$. Then if ϕ_t converges to ρ , we must have

$$\mathcal{L}^* \rho = 0.$$

Theorem. If b, σ are “nice”, then for all $x \in \mathbb{R}^d$ we obtain a limit probability distribution

$$\lim_{t \rightarrow \infty} \phi_t(x, y) = \rho(y)$$

that satisfies the partial differential equation $\mathcal{L}^* \phi = 0$.

So, abstractly, we now know that ρ is the stationary distribution. But that still doesn’t give a sampling procedure.

Bad example. Let $b = 0$, and let σ be the identity matrix, so that our SDE becomes

$$dX_t = dW_t.$$

Then $X_t = X_0 + W_t$, and hence $X_t^x = x + W_t$. What’s the probability distribution of X_t^x as $t \rightarrow \infty$? Since W_t is a standard Brownian motion, we have

$$\text{dist}(X_t^x) = \mathcal{N}(x, t),$$

i.e., a normal distribution that gets “wider and wider” in time. But then as $t \rightarrow \infty$ there is no meaningful limit distribution: the probability just keeps “spreading out” forever. In terms of our PDE, we can apply the Kolmogorov forward and backward equations (in 1D) to see that

$$\mathcal{L}f = 0 \cdot \partial_y f + \frac{1}{2} \partial_y^2 f$$

and

$$\mathcal{L}^* f = 0 \cdot \partial_y f + \frac{1}{2} \partial_y^2 f.$$

We would now like to find functions ρ such that

$$\mathcal{L}^* \rho = \frac{1}{2} \partial_y^2 \rho.$$

In 1D, any such function must be an *affine function*, i.e., a function of the form

$$\rho(y) = m + ny.$$

But there is no choice of constants $m, n \in \mathbb{R}$ that makes this function a probability distribution function, i.e., such that $\int_{\mathbb{R}} \rho(y) dx = 1$.

If, on the other hand, we don’t have a “bad example,” i.e., the density of the stochastic process does approach some valid density ρ , then we can sample from ρ by just simulating the SDE for a very long time t (e.g., using Euler-Maruyama¹).

¹Though here of course you must be careful about discretization error!

3 Langevin Monte Carlo

We can now describe the *Langevin Monte Carlo (LMC)* algorithm for sampling from a distribution. Suppose are given a probability density function $\rho : \mathbb{R}^d \rightarrow [0, \infty)$ that we would like to sample from. Consider the function

$$U := -\ln \rho,$$

(i.e., $\rho = e^{-U}$), and let

$$b = -\nabla U$$

(i.e., $b_i = -\partial_i U$), and

$$\sigma = \sqrt{2}I,$$

where I is the $d \times d$ identity matrix. Without saying where this formulation came from, we can still go ahead and check that it gives us data for an SDE whose limit distribution is ρ . Hence, by simulating the corresponding SDE, we can sample from ρ . More explicitly, we can solve the SDE

$$dX_t = -\nabla U(X_t) dt + \sqrt{2}dW_t. \quad (2)$$

The claim is then that, as $t \rightarrow \infty$, $\text{dist}(X_t)$ goes to a distribution with probability density ρ .

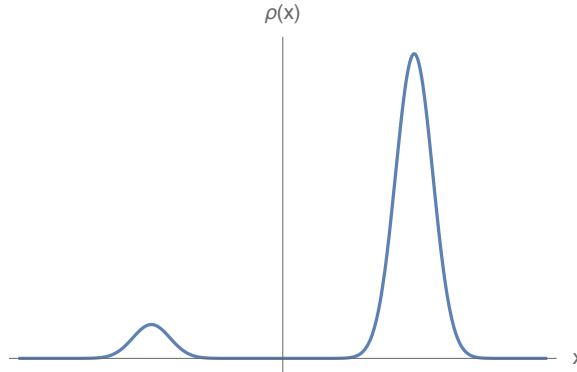
To actually solve the SDE, we can apply the following algorithm (basically just Euler-Maruyama):

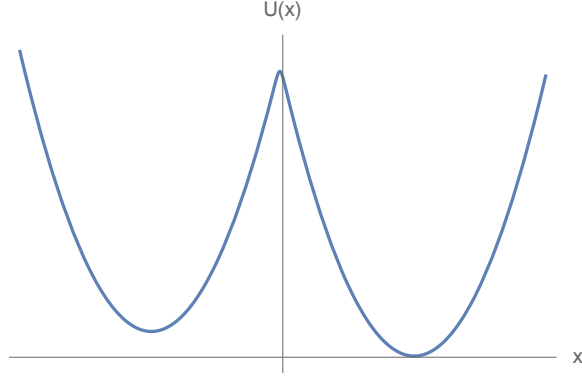
1. Choose some time step $\tau > 0$, $x_0 \in \mathbb{R}^d$.
2. Iteratively evaluate $X_{n+1} \leftarrow X_n - \tau \nabla U(X_n) + \sqrt{2\tau} \zeta_{n+1}$,

where for each time step n , ζ_n is sampled from a standard normal distribution $\mathcal{N}(0, 1)$.

To see that the LMC algorithm works (i.e., that it samples from the right distribution), we basically just have to compute $\mathcal{L}^*$, and check that $\mathcal{L}^* \rho = 0$ (which will be done in your homework).

3.1 What Can Go Wrong





On the homework, we considered a distribution

$$\rho = 0.1G_1 + 0.9G_2,$$

where G_1 and G_2 are two Gaussian distributions, centered at different locations. Notice that in between these two distributions, ρ can be extremely small—hence, $U = -\ln \rho$ becomes very large. Suppose we now start simulating the SDE starting in one of the two “wells” in U (corresponding to one of the two centers of ρ). Then we will have strong forces pushing us back into the well. In order to make our way into the other well, we will have to get extremely lucky: our random forces have to “collude”, in order for us to push over the energy barrier and make it into the next well. But just like always getting “heads” on a coin flip, this collusion will happen with exponentially small probability. So, this is an example of a problem where LMC does not work well: the stationary distribution is still correct, but we will not reach that distribution within our lifetime.

One thing we can perhaps do to improve the situation is to start from a *distribution* of points, rather than a single point. E.g., a uniform sample of the domain—or better yet, a distribution that roughly looks like the distribution ρ . Of course, the whole challenge in the first place is that we don’t know how to sample from ρ ! So it’s a bit of a chicken and egg problem. But certainly in practice people do try to mix other Monte Carlo sampling strategies with LMC.

Another thing to be cautious about is that the time step parameter τ will impact the stationary distribution of our discrete random process. If τ is too large, we might get a distribution that (say) looks “blurred out” relative to the true distribution. If τ is too small, we might get the right distribution—but at a very high cost. How to pick a “good” τ (i.e., that provides a good “bang for your buck”) is problem-dependent.

3.2 Non-normalized distributions

Suppose our target distribution is given by

$$\rho = \frac{\rho_u(x)}{Z},$$

where $\rho_u(x)$ is some positive non-normalized function, and $Z := \int_{\Omega} \rho_u(x) dx$ is the normalization factor. Then

$$U = -\ln(\rho) \neq -\ln(\rho_u).$$

But

$$\nabla U = -\nabla \ln \rho = -\nabla \ln(\rho_u).$$

So, to sample from ρ_u , just choose $U = -\nabla \ln(\rho_u)$ and run LMC as usual.

This feature is one of the great strengths of LMC: it allows us to sample from *any* positive function, not just one that has already been normalized. In many application contexts (e.g., machine learning, which involves very

large data sets, or other examples we've seen in this class), explicitly normalizing a distribution is not practical, or not feasible in the lifetime of the universe.

4 Metropolis Adjusted Langevin Monte Carlo

Suppose we want to sample from a density ρ (or from ρ_u/Z for some unnormalized positive function ρ_u). The basic approach of *Metropolis adjusted Langevin Monte Carlo (MALA)* is

1. Propose X_{n+1} according to the LMC update rule.
2. Accept or reject X_{n+1} via Metropolis-Hastings.

In particular, consider the acceptance ratio

$$A(x, y) := \frac{\rho(y)Q(y, x)}{\rho(x)Q(x, y)},$$

and let

$$Q(x, y) := \exp\left(\frac{-|x - \tau \nabla U(x) - y|^2}{4\tau}\right),$$

(i.e., a Gaussian distribution). Suppose we pick a time step $\tau > 0$, and some initial distribution X_0 . The MALA algorithm is then

1. Propose a new sample

$$\tilde{X}_{n+1} = X_n - \tau \nabla U(X_n) + \sqrt{2\tau} \zeta_{n+1}, \quad \zeta \sim N(0, 1).$$

2. Flip a biased coin that lands “heads” with probability $\min(1, A(X_n, \tilde{X}_n))$.
3. If the coin landed heads, accept the proposed sample ($X_{n+1} \leftarrow \tilde{X}_{n+1}$). Otherwise, keep the old sample ($X_{n+1} \leftarrow X_n$).