CPSC 661: Sampling Algorithms in Machine Learning February 17, 2021

Lecture 6

Lecturer: Andre Wibisono *Scribe: Grigoris Velegkas*

1 Outline

Today's lecture is about algorithms that given a Markov chain P with stationary distribution μ and some target distribution ν, they simulate a reversible Markov chain $\hat{P}$ with stationary distribution ν. These lecture notes cover:

- Metropolis-Hastings algorithm
- More general class of such algorithms proposed by Billera and Diaconis [2]
- Optimization perspective of the Metropolis-Hastings algorithm in the space of probability distributions

2 Metropolis Algorithm

We start by explaining how the Metropolis algorithm works. For this algorithm, we assume access to a *symmetric* Markov chain $P = (P_x : x \in \mathcal{X})$, i.e. $P_x(y) = P_y(x)$, $\forall\, x, y \in \mathcal{X}$. We are also given some target distribution ν we want to sample from. In a subsequent section, we will relax the assumption that P is symmetric. The algorithm performs a random walk with the following accept/reject step:

1. From x, draw $y \sim P_x$. We call y the *proposal*.
2. Accept y with probability $\min\left\{1, \frac{\nu(y)}{\nu(x)}\right\}$. Else, reject y. Hence, for the next point x' we have:

$$x' = \begin{cases} y \sim P_x, & \text{with probability } \alpha_x(y) = \min\left\{1, \dfrac{\nu(y)}{\nu(x)}\right\} \\ x, & \text{with probability } 1 - \alpha_x(y) \end{cases}$$

We want to note the following about the Metropolis algorithm:

- This algorithm was introduced by Metropolis et al. [7] in 1953.

- This procedure is intuitive, in the sense that if the proposed point y has $\nu(y) > \nu(x)$ then the walk will always move there. Otherwise, it will stay at the same point with some probability.
- It is a zero-order algorithm, i.e. it depends only on $\nu(x)$ and not the gradient.
- It requires knowledge of $\nu(x)$ up to a *normalizing constant*. This is especially good for applications that require Bayesian computation. In these settings, after we observe some data we want to update our estimate of an unknown parameter θ. We know that

  $$p(\theta \mid \text{data}) \propto p(\theta) \cdot p(\text{data} \mid \theta).$$

  Usually, we can compute the quantities $p(\theta)$, $p(\text{data} \mid \theta)$ but we cannot compute the normalizing constant $p(\text{data}) = \int_{\theta \in \Theta} p(\theta) \cdot p(\text{data} \mid \theta)\, d\theta$ because the space is too large. Thankfully, we can run the Metropolis algorithm without computing the previous quantity.

  This has also been leveraged in other domains such as physics and machine learning. Typically, in these applications we have that $\nu(x) \propto e^{-f(x)}$. This is called the Gibbs distribution. We call $f : \mathcal{X} \to \mathbb{R}$ the *potential function* or the *energy*.

  - In physics, the Metropolis algorithm was initially used with f being the *Hamiltonian*, i.e. the kinetic energy of a configuration of N particles in $\mathbb{R}^2$. In that case, $\mathcal{X} = \mathbb{R}^{4N}$ because each particle x_i corresponds to a vector $x_i = \begin{pmatrix} q_i \\ p_i \end{pmatrix} \in \mathbb{R}^2 \times \mathbb{R}^2$, where q_i is the position of the particle and p_i is the momentum of the particle.
  - In convex geometry, we are dealing with a convex f which makes the distribution ν log-concave. We can use techniques from the field of convex optimization to study log-concave functions.

2.1 Metropolis Random Walk (MRW)

We now present a concrete example of the Metropolis algorithm, the Metropolis random walk (MRW). Here, we start with P to be the Brownian motion, when we are in the continuous setting, or the Gaussian noise with step size η, when we are in the discrete setting. In the MRW, when we are at some point $x \in \mathcal{X}$, we draw a new proposal point $y = x + \sqrt{2\eta}\, Z$, where $Z \sim \mathcal{N}(0, I)$. As described in the Metropolis algorithm, we accept y with probability $\min\left\{1, \frac{\nu(y)}{\nu(x)}\right\}$. We stay at the same point with the remaining probability.

- Consider the case where ν is the uniform distribution on some $K \subset \mathbb{R}^n$, i.e.

  $$\nu(x) = \begin{cases} \frac{1}{\text{Vol}(K)} & \text{if } x \in K \\ 0 & \text{if } x \notin K. \end{cases}$$

In this setting, $\nu(x) \propto e^{-f(x)}$, where $f(x) = -\log(\nu(x)) = \mathbb{1}_K(x)$ is the convex indicator function:

$$\mathbb{1}_K(x) = \begin{cases} 0 & \text{if } x \in K \\ \infty & \text{if } x \notin K. \end{cases}$$

We note that f is a convex function if the set K is convex.

- When our distribution ν is Gaussian, then f is a quadratic function.

3 Metropolis-Hastings Algorithm

In this section we describe the Metropolis-Hastings (MH) algorithm, which generalizes the Metropolis algorithm we presented before. The important difference between the two algorithms, is that Metropolis-Hastings does not to start with a symmetric Markov chain P; the initial Markov chain can be arbitrary. The algorithm works in the following way:

1. From x, draw $y \sim P_x$. We call y the *proposal*.
2. Accept y with probability $\min\left\{1, \frac{\nu(y)\cdot P_y(x)}{\nu(x)\cdot P_x(y)}\right\}$. Else, reject y. Hence, for the next point x' we have:

$$x' = \begin{cases} y & \text{with probability } \alpha_x(y) = \min\left\{1, \dfrac{\nu(y)\cdot P_y(x)}{\nu(x)\cdot P_x(y)}\right\} \\ x & \text{with probability } 1 - \alpha_x(y). \end{cases}$$

We note the following:

- This algorithm was proposed by Hastings [6] in 1970. In the case where P is symmetric it is the same as the Metropolis algorithm.
- It requires zero-order access to ν and we only need to know ν up to a constant.
- It is a universal algorithm in the sense that we can start with any Markov chain P that has stationary distribution μ and convert it to a Markov chain $\hat{P}$ with the "correct" stationary distribution ν.

We might also refer to the MH algorithm as the MH filter.

3.1 Why Does MH Work?

We now prove the correctness of the MH algorithm. Our proof relies on the fact that if a distribution ν satisfies the detailed balance equations for a Markov chain P, then it is a stationary distribution for P.

Lemma 1. *Let P be the initial Markov chain and $\hat{P}$ the Markov chain that is induced by applying the MH algorithm to P with respect to ν. Then, $\hat{P}$ is reversible with respect to ν.*

Proof. Let $x, y \in \mathcal{X}$. We define $\alpha_x(y) = \min\left\{1, \frac{\nu(y)\cdot P_y(x)}{\nu(x)\cdot P_x(y)}\right\}$, as in the algorithm. Then,

$$\hat{P}_x(y) = P_x(y) \cdot \alpha_x(y) + \delta_x(y) \cdot A(x),$$

where $A(x) = P_x(\{x\}) + \int_{\mathcal{X}\setminus\{x\}}(1-\alpha_x(y)) \cdot P_x(y)dy = 1 - \int_{\mathcal{X}\setminus\{x\}} \alpha_x(y) \cdot P_x(y)dy$. We need to show that $\nu(x)\hat{P}_x(y) = \nu(y)\hat{P}_y(x)$. We consider the following cases:

- If $x = y$, this is true.
- If $x \neq y$, then:

$$\begin{aligned}\nu(x)\hat{P}_x(y) &= \nu(x)P_x(y)\alpha_x(y)\\ &= \nu(x)P_x(y)\min\left\{1, \frac{\nu(y)\cdot P_y(x)}{\nu(x)\cdot P_x(y)}\right\}\\ &= \min\{\nu(x)P_x(y), \nu(y)P_y(x)\}\\ &= \nu(y)\hat{P}_y(x),\end{aligned}$$

  where the last equality follows by the fact that the expression is symmetric in x, y.

□

4 A More General Class of Algorithms

The problem we have been studying is given a Markov chain $P = (P_x : x \in \mathcal{X})$ and some target distribution ν, how do we create a Markov chain $\hat{P}$ such that $\hat{P}$ is reversible with respect to ν? The Metropolis-Hastings algorithm solves this problem, but as we will see now it is an instance of a more general class of algorithms that was proposed by Billera and Diaconis [2]. All the algorithms in this class work in the following way:

1. From x, draw $y \sim P_x$. We call y the *proposal*.

2. Accept y with some probability $\alpha_x(y)$. Else, reject y. Hence, for the next point x' we have:

$$x' = \begin{cases} y & \text{with probability } \alpha_x(y) \\ x & \text{with probability } 1 - \alpha_x(y). \end{cases}$$

Let

$$R_x(y) = \frac{\nu(y) \cdot P_y(x)}{\nu(x) P_x(y)} = \frac{1}{R_y(x)}.$$

Billera and Diaconis proved the following lemma [2, Proposition 3.1].

Lemma 2. *Choose any $0 \leq \alpha_x(y) \leq \min\{1, R_x(y)\}$ and set $\alpha_y(x) = \frac{\alpha_x(y)}{R_x(y)}$. Then the induced Markov chain $\hat{P}$ is reversible with respect to ν.*

The following examples are instantiations of the above algorithm:

- Metropolis et al. [7] set $\alpha_x(y) = \min\{1, R_x(y)\}$.
- Barker [1] sets

$$\alpha_x(y) = \frac{\nu(y) P_y(x)}{\nu(x) P_x(y) + \nu(y) P_y(x)} = \frac{R_x(y)}{1 + R_x(y)}.$$

In the previous two examples, when P is already reversible with respect to ν, the Metropolis algorithm always accepts the proposed point, whereas Barker's algorithm accepts it with probability 1/2.

5 Optimization Interpretation of Metropolis-Hastings

In this section we will see a precise sense in which the Metropolis-Hastings algorithm is optimal. We define $\mathcal{M}$ to be set of all Markov chains on space $\mathcal{X}$, i.e. $\mathcal{M} = \{P = (P_x : x \in \mathcal{X}), P_x \in \mathcal{P}(\mathcal{X})\} = \mathcal{P}(\mathcal{X})^{\mathcal{X}}$, where $\mathcal{P}(\mathcal{X})$ is the set of all probability distributions on space $\mathcal{X}$. Given some $\nu \in \mathcal{P}(\mathcal{X})$, we define $\mathcal{M}_\nu = \{P \in \mathcal{M} : P \text{ reversible w.r.t. } \nu\}$ to be the set of all Markov chains on $\mathcal{X}$ that are reversible with respect to ν. Then, the Metropolis Hastings filter takes as input some $P \in \mathcal{M}$ and maps it to some $\hat{P} \in \mathcal{M}_\nu$. We now define the following metric between Markov chains on $\mathcal{M}$:

$$d(P, P') = \int_{\mathcal{M} \times \mathcal{M} \setminus \Delta} |P_x(y) - P'_x(y)| \, \nu(x) \, dx \, dy$$

where $\Delta = \{(x, x) : x \in \mathcal{X}\}$. Billera and Diaconies proved the following theorem [2, Theorem 1].

Theorem 1. *The MH filter is projection in d-distance, i.e.*

$$\hat{P} = \arg \min_{\tilde{P} \in \mathcal{M}_\nu} d(\tilde{P}, P).$$

Here, P is the initial Markov chain and $\hat{P}$ is the one we get after applying the MH filter. We also remark that $\hat{P}$ is the unique minimizer such that $\hat{P}_x(y) \leq P_x(y)$.

One might wonder why this specific metric d is used to measure the distance between two Markov chains. We give the following motivation. Given P, ν we define the joint distributions:

$$\begin{aligned}(\nu P)(x, y) &= \nu(x) \cdot P_x(y) \in \mathcal{P}(\mathcal{X} \times \mathcal{X}) \\ (\overline{\nu P})(x, y) &= \nu(y) \cdot P_y(x) \in \mathcal{P}(\mathcal{X} \times \mathcal{X})\end{aligned}$$

Notice that P is reversible with respect to ν if and only if $(\nu P)(x, y) = (\overline{\nu P})(x, y), \forall x, y \in \mathcal{X} \times \mathcal{X}$. Then, one can show that for the metric $d(\cdot, \cdot)$ we defined before we have

$$d(P, \mathcal{M}_\nu) = \min_{\tilde{P} \in \mathcal{M}_\nu} d(P, \tilde{P}) = \text{TV}(\nu P, \overline{\nu P})$$

where TV is the total variation distance.

5.1 Example in 2-Point Space

We consider the following example where $\mathcal{X} = \{0, 1\}$. Here, every Markov P chain has the form

$$P = \begin{pmatrix} 1-a & a \\ b & 1-b \end{pmatrix}$$

Also, the stationary distribution is $\left(\frac{b}{a+b}, \frac{a}{a+b}\right)$. Given some target distribution $\pi = (\pi(0), \pi(1))$, we have that P is stationary with respect to π if and only if $\pi(0) \cdot a = \pi(1) \cdot b$. Figure 5.1 shows pictorically the spaces $\mathcal{M}, \mathcal{M}_\nu$. The figure is taken from Billera and Diaconis [2].

5.2 Extensions

The following list of papers includes other extensions and ideas that should be suitable for the class projects. Such extensions include projections in other distances such as KL-divergence and χ^2-divergence. Also, they study Markov chains in continuous time.

- Diaconis and Miclo [5].
- Choi [4].
- Chewi, Lu, Ahn, Cheng, Gouic, and Rigollet [3].

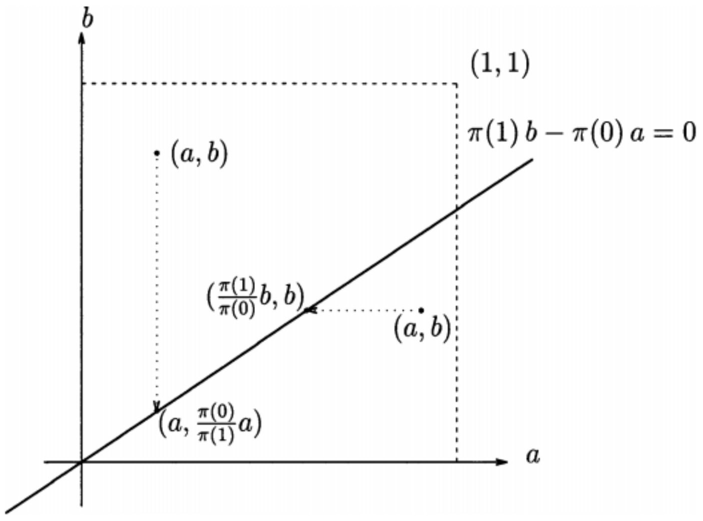
b
(1, 1)
$\pi(1)\,b - \pi(0)\,a = 0$
(a, b)
$(\frac{\pi(1)}{\pi(0)}b, b)$
(a, b)
$(a, \frac{\pi(0)}{\pi(1)}a)$
a

Figure 1: The 1×1 square represents the space of all Markov chains. The line represents the space of all reversible Markov chains with respect to π.

6 Basic Algorithms for Sampling

By combining the Metropolis-Hastings filter with some initial Markov chain P we can get different sampling algorithms for some target distribution ν. These sampling algorithms include:

- The Metropolis Random Walk (MRW).
- The Unadjusted Langevin Algorithm (ULA).
- The Metropolis-Adjusted Langevin Algorithm (MALA).

In Section 2.1, we saw how MRW works. We will briefly explain how the rest of the algorithms work.

6.1 Unadjusted Langevin Algorithm (ULA)

Here, the goal is to sample from $\nu \propto e^{-f}$. When we are at some point x_k, the point x_{k+1} that we move to is

$$x_{k+1} = x_k - \eta \nabla f(x_k) + \sqrt{2\eta}\, Z_k$$

where $Z_k \sim \mathcal{N}(0, I)$ is independent. We will talk about this algorithm more in future lectures. We want to point out that this algorithm has an asymptotic bias and the stationary distribution of the induced Markov chain is not equal to ν.

6.2 Metropolis Adjusted Langevin Algorithm (MALA)

This algorithm is a combination of ULA with the Metropolis-Hastings filter. This technique fixes the issue with the bias and the stationary distribution is equal to ν.

References

[1] A. A. Barker. Monte Carlo calculations of the radial distribution functions for a proton- electron plasma. *Australian Journal of Physics*, 18(2):119–134, 1965.

[2] L. J. Billera and P. Diaconis. A geometric interpretation of the Metropolis-Hastings algorithm. *Statistical Science*, pages 335–339, 2001.

[3] S. Chewi, C. Lu, K. Ahn, X. Cheng, T. L. Gouic, and P. Rigollet. Optimal dimension dependence of the Metropolis-adjusted Langevin algorithm. *arXiv preprint arXiv:2012.12810*, 2020.

[4] M. C. Choi. Metropolis–Hastings reversiblizations of non-reversible Markov chains. *Stochastic Processes and their Applications*, 130(2):1041–1073, 2020.

[5] P. Diaconis and L. Miclo. On characterizations of Metropolis type algorithms in continuous time. *ALEA: Latin American Journal of Probability and Mathematical Statistics*, 6:199–238, 2009.

[6] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. 1970.

[7] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.