**CSE6243: Advanced Machine Learning** **Fall 2023**

# Lecture 7: Sampling: Markov Chain Monte Carlo (MCMC)

*Lecturer: Bo Dai* *Scribes: Wenbo Chen, Hangtian Zhu*

**Note**: *LaTeX template courtesy of UC Berkeley EECS Department.*



## 7.1 Recap

Given function $f(x)$ and the target distribution $p(x)$, we want to inference the empirical mean by sampling:

$$\mathbb{E}_{p(x)}[f(x)] \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i), \text{ where } x_i \sim p(x)$$

The previous lectures covered acceptance-rejection sampling and importance sampling. They sample from a proposal distribution $q(x)$ which is easier to sample. Acceptance-rejection sampling accepts the sample with probability $\frac{p(x)}{Mq(x)}$. Importance sampling reweights the sample with $\frac{p(x)}{q(x)}$.

However, both sampling methods have limitations. Acceptance-rejection sampling requires finding a $M$ such that $Mq(x) \geq p(x)\ \forall x$. $M$ could be very large for high-dimension distribution and thus waste a large number of samples.

Importance sampling could have very high variance:

$$Var[\frac{p(x)}{q(x)}f(x)] = \mathbb{E}_q[\frac{p^2(x)}{q^2(x)}f^2(x)] - \mathbb{E}_q^2[\frac{p(x)}{q(x)}f(x)] = \int f^2(x)\frac{p^2(x)}{q(x)}dx - \mathbb{E}_p^2[f(x)] \quad (7.1)$$

Specifically, $q(x) \to 0, p(x) \neq 0, Var[\frac{p(x)}{q(x)}f(x)] \to \infty$.

## 7.2 New Content

In this lecture, we introduce *Markov Chain Monte Carlo (MCMC)*.

### 7.2.1 Intuition

Define the conditional probability or transition kernel $T(\cdot)$, we want to construct a sequence of sampling:

$$x_0 \sim p_0(x), x_1 = T(x_0), x_0 = T(x_1), \cdots, x_T \sim p(x),$$

such that along the steps, the sampling converges to the target distribution.

**Algorithm 1** MCMC

$x_0 \sim p_0(x)$
**for** $t = 1 \cdots T$ **do**
  $x_{t+1} \sim T(\cdot|x_t)$
**end for**

### 7.2.2 MCMC

As mentioned, we want the samples converge to the target distribution:

$$p(x) = \lim_{t\to\infty} \int T^t(x|x_0)p(x_0)dx_0,$$

where

$$T^t(x|x_0) = \int T^{t-1}(x|x_1)T(x_1|x_0)dx_1 = \int \prod_{i=0}^{t} T(x_{i+1}|x_i)d\{x_i\}_{i=1}^{t-1}.$$

**Theorem 7.1** *The procedure converges to the target distribution if and only if the following conditions hold:*

$$\text{1) } p(x) \text{ is a stationary distribution of the Markov chain } T(x|x'), i.e., \tag{7.2}$$

$$p(x') = \int T(x|x')p(x)dx, \tag{7.3}$$

$$\text{2) There is only one stationary distribution } p(x). \tag{7.4}$$

Theorem 7.1 is typically hard to check and people typically look into the sufficient condition:

**Theorem 7.2** *The procedure converges to the target distribution if the following conditions hold:*

$$\text{1) Detailed balance: } p(x)T(y|x) = p(y)T(x|y), \tag{7.5}$$

$$\text{2) Ergodicity: } \forall x, T(\cdot|x) > 0 \text{ and } T^t(\cdot|x) > 0. \tag{7.6}$$

The intuition of 2) in Theorem 7.2 is the sample can go everywhere at every step.

**Proof:**

$$\int p(y)T(x|y)dy = \int p(x)T(y|x)dy \quad \text{(detailed balance in 7.5)} \tag{7.7}$$

$$= p(x)\int T(y|x)dy \tag{7.8}$$

$$= p(x) * 1 \tag{7.9}$$

$$= p(x) \tag{7.10}$$

■

### 7.2.3 Metropolis-Hastings (MH) algorithm

**Theorem 7.3** *Metropolis-Hastings in Algorithm 2 satisfies the detailed balance.*

**Algorithm 2** Metropolis-Hasting (MH)

$x_0 \sim p_0(x)$
**for** $t = 1 \cdots T$ **do**
  $x = x_t$
  $y \sim q(\cdot|x)$
  $A(x, y) = \min(\frac{p(y)q(x|y)}{p(x)q(y|x)}, 1)$
  $u \sim U[0, 1]$
  **if** $u \leq A(x, y)$ **then**
    $x_{t+1} = y$
  **else**
    $x_{t+1} = x$
  **end if**
**end for**

*The blue part is the transition kernel $T(\cdot|x)$

**Proof:**

$$
\begin{aligned}
p(x)T(y|x) &= p(x)A(x,y)q(y|x) \\
&= p(x)q(y|x)[\min(\frac{p(y)q(x|y)}{p(x)q(y|x)}, 1)] \\
&= \min(\frac{p(x)q(y|x)}{p(y)q(x|y)} \cdot \frac{p(y)q(x|y)}{p(x)q(y|x)}, \frac{p(x)q(y|x)}{p(y)q(x|y)})p(y)q(x|y) \\
&= \min(1, \frac{p(x)q(y|x)}{p(y)q(x|y)})p(y)q(x|y) \\
&= A(y,x)p(y)q(x|y) \\
&= p(y)T(x|y)
\end{aligned}
$$

■

Based on Theorem 7.2, we know MH converges to the target distribution. MH is a template algorithm, based on different designs of the distribution $q(\cdot|x)$, we get different instantiation of algorithms such as random walk, Gibbs sampling, and Metropolis Adjust Langevin Algorithm (MALA).

### 7.2.4 Random Walk

Random walk chooses

$$q(y|x) \propto \exp(-\frac{\|y-x\|^2}{2\sigma^2}) \propto U(\|y-x\| \leq \delta).$$

The acceptance rate is

$$A(x,y) = \min(\frac{p(y)q(x|y)}{p(x)q(y|x)}, 1) = \frac{p(y)}{p(x)},$$

since $q(y|x) = q(x|y)$. Random walk could be written as $y = x + \epsilon$, where $\epsilon \sim N(0, \sigma^2)$. The choice of $\sigma$ controls the tradeoff between the computational cost of getting a new sample and dependency i.e., how different the new sample is from the previous points.

### 7.2.5 Gibbs Sampling

Gibbs sampling only changes one entry in $x$ at a time. Recall $p(x) = p(x_0, \cdots, x_d), x \in \mathbb{R}^d$. We define

$$q(y|x) = p(x_i|x_{-i}),$$

where $x_{-i} = \{x_0, \cdots, x_{i-1}, x_{i+1}, \cdots, x_d\}$.

The acceptance rate is:

$$A(x, y) = \min(\frac{p(y)q(x|y)}{p(x)q(y|x)}, 1) = \min(\frac{p(x_i)p(x_{-i}|x_i)}{p(x_{-i})p(x_i|x_{-i})}, 1) = 1$$

### 7.2.6 Metropolis Adjusted Langevin Algorithm (MALA)

MALA could be viewed as injecting target probability into random walk:

$$y = x + \eta \nabla \log p(x) + \sqrt{\eta}\epsilon$$