3 Markov Chain Monte Carlo (Lectures 5 & 6)

In previous lectures we motivated the computation of integrals $\mathbb{E}_{X\sim\pi}[f(X)]$ and showed how they can be estimated via Monte Carlo when sampling from π is feasible. Unfortunately, it is rarely the case that we know how to generate i.i.d. samples from π. In addition, π is usually known only up to a multiplicative constant.[7] Markov Chain Monte Carlo (MCMC) is a suite of techniques used to estimate expectations when simple Monte Carlo approaches are infeasible.

Notice that i.i.d. samples are not crucial to compute the integral correctly. What we are really after is a sequence $X_1, \dots, X_N$ that would cover the state space with proportions dictated by our target distribution π. Therefore, instead of sampling $X_1, \dots, X_N$ i.i.d. from π, one could consider generating this sequence sequentially, for instance, as a Markov Chain that has π as its stationary distribution. Intuitively, we expect this Markov Chain to perform a random walk around the state space, such that the proportion of time spent in different sets of the state space would correspond to the target measure. In that case, the sample average estimator $(1/N)\sum_{i=1}^{N} f(X_i)$ will be biased, but one can hope that the bias decreases to 0 as $N \to \infty$ (then the estimator is said to be asymptotically unbiased).

3.1 Discrete Markov Chains

Before we discuss general state space Markov Chains, it is helpful to illustrate the idea of Markov Chain Monte Carlo on finite state spaces. You are possibly already familiar with the following definition of a Markov Chain.

Definition 16. *Let $\{X_n\}_{n\in\mathbb{N}}$ be a random process taking values in a finite set $S = \{1, \dots, K\}$. We call $\{X_n\}_{n\in\mathbb{N}}$ a Markov Chain if for any values $x_0, \dots, x_n \in S$ we have*

$$\mathbb{P}(X_n = x_n \mid X_{n-1} = x_{n-1}, X_0 = x_0) = \mathbb{P}(X_n = x_n \mid X_{n-1} = x_{n-1}) \tag{73}$$

i.e. the future is independent of the past, given the present.

If, in addition, the distribution $\mathbb{P}(X_n = x_n | X_{n-1} = x_{n-1})$ is independent of n, it is a time-homogeneous Markov Chain. In that case, we define the transition matrix P as

$$P_{ij} = \mathbb{P}(X_n = j \mid X_{n-1} = i) \tag{74}$$

for all $i, j \in S$.

In other words, Markov Chains generate sequences of random values $X_0, \dots, X_n$ by looking only at the previously generated value. The distribution of the first value is called the initial distribution and will often be written as a vector μ, i.e. $\mu_i = \mathbb{P}(X_0 = i)$. Then it is easy to check that the marginal distribution of X_n has the following nice matrix form

$$\mathbb{P}(X_n = i) = (\mu P^n)_i \tag{75}$$

As $n \to \infty$ it might be the case that the vector μP^n converges to some fixed vector π. In that case, we might expect the sample average $(1/N)\sum_{i=1}^{N} f(X_i)$ to converge to the integral $\mathbb{E}_{X\sim\pi}[f(X)]$.

[7] Often this is a consequence of the Bayes' rule (1), when the marginal likelihood $p(y)$ is intractable.

Definition 17. *If $\{X_n\}_{n\in\mathbb{N}}$ is a time-homogeneous Markov Chain with transition matrix P, we call π a stationary distribution of $\{X_n\}_{n\in\mathbb{N}}$ if*

$$\pi P = \pi \tag{76}$$

That is, if X_{n-1} is distributed according to the stationary distribution π, X_n will also be distributed according to the stationary distribution π.

Definition 18. *A Markov Chain is irreducible if for any $i, j \in S$ there is an $n \in \mathbb{N}$ such that*

$$\mathbb{P}(X_n = j \mid X_0 = i) > 0 \tag{77}$$

That is, any state is reachable from any state.

Theorem 8. *If $\{X_n\}_{n\in\mathbb{N}}$ is a time-homogeneous Markov Chain on a finite state space, then it has a (one or more) stationary distribution π. If the Markov Chain is irreducible, π is unique.*

Theorem 9 (Ergodic Theorem)**.** *Let $\{X_n\}_{n\in\mathbb{N}}$ be an irreducible time-homogeneous Markov Chain on a finite state space with stationary distribution π. For each $i \in S$ write $V_i(n) = \sum_{j=0}^{n-1} I(X_j = i)$ for the number of visits to state i up until time n. Then*

$$\frac{V_i(n)}{n} \to \pi_i \textit{ almost surely} \tag{78}$$

That is, the proportion of time spent in state i converges to its stationary probability π_i as $n \to \infty$. This means that if we have a function $f : S \to \mathbb{R}$ on our state space, we could evaluate the expected value $\mathbb{E}_{X\to\pi}[f(X)]$ using a Markov Chain:

$$\begin{aligned}
\mathbb{E}[(1/n)\sum_{j=0}^{n-1} f(X_j)] &= \sum_{i\in S} f(i)\mathbb{E}\left[\frac{V_i(n)}{n}\right] && (79)\\
&\to \sum f(i)\pi_i && (80)\\
&= \mathbb{E}_{X\sim\pi}[f(X)] && (81)
\end{aligned}$$

This is the whole idea of Markov Chain Monte Carlo – we will build Markov Chains such that they have stationary distributions π that we want to sample from, simulate those Markov Chains and use the generated samples to compute the expectations.

Although the treatment above looks fairly simple, it is only because we looked at finite state spaces. Working with general state spaces the theory becomes much more complicated. In this course we will not discuss extensions to the above Ergodic theorem and will only be concerned with constructing Markov Chains with specific stationary distributions.

3.2 General state space Markov Chains

We now define general state space Markov Chains using measure-theoretic notation. In the following, we will consider Markov Chains on measurable spaces $(X, \mathcal{X})$. Just like transition matrices were used to describe transitions in discrete Markov Chains, transition kernels will describe transitions in general state spaces:

Definition 19. *A Markov transition kernel on a measurable space $(X, \mathcal{X})$ is a function $\mathcal{P} : X \times \mathcal{X} \to [0,1]$ such that:*

- *$\mathcal{P}(x, \cdot)$ is a probability measure for all $x \in X$*
- *$\mathcal{P}(\cdot, A)$ is measurable for all $A \in \mathcal{X}$.*

Intuitively, the first condition says that the for each state $x \in X$ the conditional probability distribution of the next state in the Markov Chain will be

$$\mathbb{P}(X_{n+1} \in A \mid X_n = x) = \mathcal{P}(x, A)$$

The second point is a technical condition that ensures we can talk about integrals of transition kernels.

Definition 20. *A random process $\{X_n\}_{n \in \mathbb{N}}$ is a time-homogeneous Markov Chain with transition kernel $\mathcal{P}$ if for any values $x_0, \ldots, x_{n-1} \in X$ and $A \in \mathcal{X}$ we have:*

$$\begin{aligned} \mathbb{P}(X_n \in A \mid X_0 = x_0, \ldots, X_{n-1} = x_{n-1}) &= \mathbb{P}(X_n \in A \mid X_{n-1} = x_{n-1}) && (82) \\ &= \mathcal{P}(x_{n-1}, A) && (83) \end{aligned}$$

We can now define a few operatations involving transition kernels.

Definition 21. *Let $f : X \to \mathbb{R}$ be a measurable function, $\mu : \mathcal{X} \to \mathbb{R}^+$ a measure, $\mathcal{P}, \mathcal{Q}$ transition kernels. Then:*

- *$\mathcal{P}\mathcal{Q}$ is a transition kernel, corresponding to first making a transition with respect to $\mathcal{P}$ and then $\mathcal{Q}$:*

  $$\mathcal{P}\mathcal{Q}(x, A) = \int \mathcal{Q}(y, A)\mathcal{P}(x, \mathrm{d}y)$$

  and $\mathcal{P}^n = \underbrace{\mathcal{P}\mathcal{P}\ldots\mathcal{P}}_{n \text{ times}}$.

- *$\mathcal{P}f : X \to \mathbb{R}$ is a function corresponding to the expected value of f after a one step transition according to $\mathcal{P}$ starting at some point $x \in X$:*

  $$\mathcal{P}f(x) = \int f(y)\mathcal{P}(x, \mathrm{d}y)$$

- *$\mu\mathcal{P}$ is a measure after one transition, defined as:*

  $$\mu\mathcal{P}(A) = \int \mathcal{P}(x, A)\mu(\mathrm{d}x)$$

  We say that μ is stationary for $\mathcal{P}$ if $\mu\mathcal{P} = \mu$.

When the transition kernel has a density (i.e. the Radon-Nikodym derivative) with respect to some measure, we will denote it as $p(x, y)$ (i.e. if it's a density w.r.t. the Lebesgue measure, we have $\mathcal{P}(x, A) = \int_A p(x, y)\mathrm{d}y$).

You could now check that the finite-state space Markov Chains are just a specific case of the above definitions, where the transition matrix P corresponds to the Radon-Nikodym derivative of $\mathcal{P}$ w.r.t. the counting measure.

Having the terminology for Markov Chains, we will now construct transition kernels $\mathcal{P}$ such that $\pi\mathcal{P} = \pi$ for a given target measure π.

Algorithm 1 Metropolis-Hastings algorithm. Given: target density π, proposal density q, first state X_0 and number of iterations N.

```
for $i = 1, \dots, N$ do
    $Y \sim \mathcal{Q}(X_{i-1})$
    $u \sim U[0, 1]$
    if $u < \alpha(X_{i-1}, Y)$ then
        $X_i = Y$
    else
        $X_i = X_{i-1}$
    end if
end for return $X_0, \dots, X_N$
```

3.3 The Metropolis-Hastings algorithm

The first MCMC algorithm, given in Algorithm 1, is the celebrated Metropolis-Hastings algorithm, which is still used incredibly widely to build Markov Chains that target a specific density. The idea behind the algorithm is to construct a Markov Chain that would be reversible with respect to the stationary measure and show it by verifying the detailed balance equations. Markov Chains that satisfy the detailed balance equations are useful because they also have π as their stationary measure.

Definition 22. *A Markov Chain with transition kernel $\mathcal{P}$ is reversible with respect to the measure π if for all $A, B \in \mathcal{X}$:*

$$\int_A \mathcal{P}(x, B)\pi(\mathrm{d}x) = \int_B \mathcal{P}(y, A)\pi(dy) \tag{84}$$

i.e. the probability of observing the transition $A \to B$ is the same as observing the transition $B \to A$. Equation (84) *is also called the detailed balance equation.*

Theorem 10. *If the transition kernel $\mathcal{P}$ satisfies the detailed balance equations for the measure π, then $\pi\mathcal{P} = \pi$.*

Suppose that we are given a target probability measure π from which we want to draw samples. It is assumed that π has a density, which we will denote ambiguously by $\pi(x)$. The construction of the Markov Chain in the Metropolis-Hastings algorithm involves an auxiliary transition kernel $\mathcal{Q}(x, \cdot)$ with a density $q(x, y)$ for each $x \in X$, called the proposal. In essence, the algorithm simulates a Markov Chain with the transition kernel $\mathcal{Q}(x, \cdot)$ by adjusting it on every step to satisfy the detailed balance equations.

To simulate X_{n+1} from X_n, we simulate the Markov Chain one step — $Y \sim \mathcal{Q}(X_n, \cdot)$. Then, to adjust for the detailed balance equations, with probability $\alpha(X_n, Y)$ we accept the proposal and set $X_{n+1} = Y$, or reject it and set $X_{n+1} = X_n$. Here, the acceptance probability is given by

$$\alpha(x, y) = \min\left\{1, \frac{\pi(y)q(y, x)}{\pi(x)q(x, y)}\right\}$$

The resulting Markov Chain has $\mathcal{P}(x, \cdot)$ as it's transition kernel, given by

$$\mathcal{P}(x, A) = r(x)\delta_x(A) + \int_A \alpha(x, y)q(x, y)\mathrm{d}y$$

where $r(x) = \int(1 - \alpha(x, y))q(x, y)\mathrm{d}y$ is the rejection probability.

We can easily show that $\mathcal{P}$ satisfies the detailed balance equations:

$$\begin{aligned}
\int_A \mathcal{P}(x,B)\pi(\mathrm{d}x) &= \int_A \Big(r(x)\delta_x(B) + \int_B \alpha(x,y)q(x,y)\mathrm{d}y \Big)\pi(x)\mathrm{d}x && (85)\\
&= \int_A r(x)\delta_x(B)\pi(x)\mathrm{d}x + \int_A\int_B \alpha(x,y)q(x,y)\pi(x)\mathrm{d}y\mathrm{d}x && (86)\\
&= \int_B r(x)\delta_x(A)\pi(x)\mathrm{d}x + \int_A\int_B \alpha(y,x)q(y,x)\pi(y)\mathrm{d}y\mathrm{d}x && (87)\\
&= \int_B r(x)\delta_x(A)\pi(x)\mathrm{d}x + \int_B\int_A \alpha(x,y)q(x,y)\pi(x)\mathrm{d}y\mathrm{d}x && (88)\\
&= \int_B \mathcal{P}(x,A)\pi(\mathrm{d}x) && (89)
\end{aligned}$$

This means that Algorithm 1 generates a Markov Chain with stationary measure π.

It is important to note that all that is needed to run the algorithm is the ability to sample from the proposal density $q(x,y)$, which the user can choose, and to compute the acceptance ratio $\alpha(x,y)$, which only requires us to know the *unnormalized* versions of π and q. It is because of this feature that the Metropolis-Hastings algorithms are so widely used — in most complicated Bayesian models only the unnormalized version of π is known and the normalizing constant is intractable.

A good question is how to choose the proposal density $q(x,y)$. Most commonly, $q(x,y)$ is chosen as a Normal distribution, i.e.

$$q(x,y) \propto \exp\Big(-\frac{1}{2\sigma^2}\left\|x-y\right\|^2 \Big)$$

However, it is not easy to optimally choose σ^2. If σ^2 is too small, then the Markov Chain will move very slowly across the state space. If σ^2 is too big, it will very rarely accept a proposal. Both of these cases will yield high correlation samples, which is usually referred to as bad mixing. Lectures further in the course will describe other proposals, based on gradient flow, which exhibit much faster convergence behaviour.

Another issue with the Metropolis-Hastings algorithm is that it is ineffective for sampling from multimodal distributions. Specifically, if the target distribution π has two highly concentrated and distant modes, then the constructed Markov Chain will be stuck in one of them. The reason is that the valleys between the modes usually have low π values, meaning that the acceptance rate for proposals to the valleys are low and crossing them is incredibly unlikely. In addition, the average "step size" of the chain has to be small to explore the modes individually, meaning that multiple lucky steps have to be made to jump between the modes. Sampling from multimodal distributions is usually tackled with parallel tempering (PT) algorithms which will also appear later in the course. PT algorithms build a ladder of distributions with different temperatures and swap samples between them to jump between modes.

Finally, while reversibility allows us to easily construct Markov Chains with specific stationary distributions, the feature itself is somewhat undesirable. The reason is that we prefer Markov Chains that explore the state space as quickly as possible. However, by definition, reversible Markov Chains are likely to make steps back, and therefore explore the state space in a slow, diffusive fashion.

Algorithm 2 Gibb's sampler. Given: target measure π, first state X_0 and number of iterations N.

```
for $i = 1, \dots, N$ do
    for $j = 1, \dots, d$ do
        $X_i^j \sim \pi(\,\cdot\,|X_i^1, \dots, X_i^{j-1}, X_{i-1}^{j+1}, \dots, X_{i-1}^d)$
    end for
end for return $X_0, \dots, X_N$
```

3.4 Gibb's sampler

A popular alternative algorithm to construct Markov Chains is Gibb's sampler, given in Algorithm 2. Gibb's sampler assumes that the state space can be decomposed as

$$
\begin{aligned}
X &= X^1 \times \dots \times X^d \\
\mathcal{X} &= \mathcal{X}^1 \otimes \dots \otimes \mathcal{X}^d
\end{aligned}
$$

i.e. each $x \in X$ can be written as $x = (x^1, \dots, x^d)$ with $x^i \in X^i$. In addition, it assumes that the full conditional distributions of π, i.e. $\pi(X^i \in \cdot \mid X^{-i})$, are possible to sample from (we use $-i$ notation to denote exclusion of the i-th component). Subsequently, to produce a new sample, the algorithm iterates over all dimensions and updates only the corresponding component given all others by sampling from the full conditional distribution.

A single coordinate transition kernel in the Gibb's sampler can be written as

$$
\begin{aligned}
\mathcal{P}_i(x, A) &= \pi(X^i \in A_{x^{-i}} \mid X^{-i} = x^{-i}) \tag{90} \\
A_{x^{-i}} &= \{y \in X^i : (x^1, \dots, x^{i-1}, y, x^{i+1}, \dots, x^d) \in A\} \tag{91}
\end{aligned}
$$

While it is a rather cumbersome definition, the set $A_{x^{-i}}$ simply represents all points in A which can be obtained from changing the i-th component of x.

When π has a density, it is not too difficult to check that $\mathcal{P}_i$ is reversible:

$$
\begin{aligned}
\int_A \mathcal{P}_i(x, B)\pi(\mathrm{d}x) &= \int\int_{A_{x_{-i}}} \mathcal{P}_i(x, B)\pi(x^i|x^{-i})\pi(x^{-i})\mathrm{d}x^i\mathrm{d}x^{-i} \tag{92} \\
&= \int\int_{A_{x_{-i}}}\int_{B_{x_{-i}}} \pi(y|x^{-i})\pi(x^i|x^{-i})\pi(x^{-i})\mathrm{d}y\mathrm{d}x^i\mathrm{d}x^{-i} \tag{93} \\
&= \int\int_{B_{x_{-i}}}\int_{A_{x_{-i}}} \pi(y|x^{-i})\pi(x^i|x^{-i})\pi(x^{-i})\mathrm{d}x^i\mathrm{d}y\mathrm{d}x^{-i} \tag{94} \\
&= \int\int_{B_{x_{-i}}}\int_{A_{x_{-i}}} \pi(y|x^{-i})\pi(x^i|x^{-i})\pi(x^{-i})\mathrm{d}x^i\mathrm{d}y\mathrm{d}x^{-i} \tag{95} \\
&= \int_B \mathcal{P}_i(x, A)\pi(\mathrm{d}x) \tag{96}
\end{aligned}
$$

Even given the reversibility of $\mathcal{P}_i$, using a single $\mathcal{P}_i$, obviously, is not enough since $\mathcal{P}_i$ updates a single coordinate. The transition kernel for the full Gibb's sampler is given by

Algorithm 3 Two-stage Metropolis-within-Gibbs. Given: target measure π, first state X_0, proposal Q for $\pi(\,\cdot\,|X^1)$ and number of iterations N.

```
for $i = 1, \dots, N$ do
    $X_i^1 \sim \pi(\,\cdot\,|X_{i-1}^2)$
    $Y \sim q(X_i^1, X_{i-1}^2)$
    $u \sim U[0,1]$
    if $u < \alpha((X_i^1, X_{i-1}^2), (X_i^1, Y))$ then
        $X_i = Y$
    else
        $X_i = X_{i-1}$
    end if
end for return $X_0, \dots, X_N$
```

$$\mathcal{P} = \mathcal{P}_1 \mathcal{P}_2 \dots \mathcal{P}_d \tag{97}$$

and hence preserves the stationary.

This particular version of Gibb's sampler deterministically scans over components, and for this reason is in general not reversible. Other versions of Gibb's sampler are possible, for example, where components to be updated are chosen randomly and not sequentially. Then the resulting Markov Chain would be reversible, and the transition kernel would be given by

$$\mathcal{P} = \frac{1}{d} \sum_{i=1}^{d} \mathcal{P}_i$$

Although Gibb's sampler is more restrictive than Metropolis-Hastings, it has been very successfully used in numerous applications. In particular, the transition kernel of Gibb's sampler is fully derived from the target distribution π in contrast to the user-engineered proposals in MH. In addition, Gibb's sampler has the advantage of having a constant acceptance rate of 1, meaning users do not have to worry about the mixing of the chain (except for multi-modal target distributions).

Even if some full conditionals of π are impossible to sample from and possibly known only up to a multiplicative constant, it is possible to use a hybrid of the two algorithms, called Metropolis-within-Gibbs. For those full conditionals, which we cannot sample from, we can instead use a Metropolis-Hastings transition kernel and combine the kernels as in (97). An example of the algorithm is given in Algorithm 3.