1 Markov Chain Monte Carlo

We have seen Monte Carlo methods in the previous lecture that generates sample to either approximate probability density function itself, or quantity of interest (e.g. expectation, numerical integration). The *Markov Chain Monte Carlo*(MCMC) approach is simply the Monte Carlo approach applied to Markov Processes. Namely, it is sampling from a distribution defned via a stochastic process known as a Markov Process.

1.1 Markov Chain

A discrete time, discrete space stochastic process $(X_0, X_1, X_2, \ldots)$ is a *Markov Chain* (MC) if

$$\Pr[X_t = a_t | X_{t-1} = a_{t-1}, X_{t-2} = a_{t-2}, \ldots, X_0 = a_0] = \Pr[X_t = a_t | X_{t-1} = a_{t-1}].$$

This is known as the *Markov property*.

A discrete time stochastic process is a collection of random variables n $(X_t : t \in \{0, 1, 2, \ldots\})$, where t is referred as time. X_t is the state variable at time t. Discrete time refers to t taking values $\{0, 1, 2, \ldots\}$ and discrete space refers to X_i being samples from a countable set.

We can define the transition probability for a (time-homogeneous) Markov Chain as

$$p_{ij} = \Pr[X_t = j | X_{t-1} = i].$$

Then, the 1-step transition matrix is

$$P = \begin{pmatrix} p_{00} & p_{01} & p_{02} & \cdots & p_{0j} & \cdots \\ p_{10} & p_{11} & p_{12} & \cdots & p_{1j} & \cdots \\ p_{20} & p_{21} & p_{22} & \cdots & p_{2j} & \cdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots \\ p_{i0} & p_{i1} & p_{i2} & \cdots & p_{ij} & \cdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots \end{pmatrix}.$$

The sum of each row equals 1. In other words, for all i, $\sum_{j \geq 0} p_{ij} = 1$.

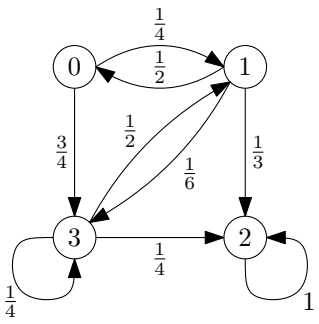


Figure 1: An example of a Markov chain represented as a weighted directed graph.

Let $p_i(t)$ be the probability that the process is at state i at time t. Then,

$$p_i(t) = \sum_{j \geq 0} p_j(t-1) p_{ji}.$$

Let $\mathbf{p}(t) = \begin{pmatrix} p_0(t) & p_1(t) & p_2(t) & \ldots \end{pmatrix}$. Then,

$$\mathbf{p}(t) = \mathbf{p}(t-1) P.$$

Figure 1 shows an example of a Markov chain with transition matrix

$$\begin{pmatrix} 0 & \frac{1}{4} & 0 & \frac{3}{4} \\ \frac{1}{2} & 0 & \frac{1}{3} & \frac{1}{6} \\ 0 & 0 & 1 & 0 \\ 0 & \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \end{pmatrix}.$$

Let $p_{ij}^m = \Pr[X_{t+m} = j | X_t = i]$ be the m-step transition probability, i.e. the probability that the Markov chain moves from state i to state j in exactly m steps. Then,

$$p_{ij}^m = \sum_{k \geq 0} p_{ik} p_{kj}^{m-1}.$$

In the above example, p_{03}^3 will be the total probability of all paths from state 0 to state 3 of length exactly 3. These paths are $(0,1,0,3), (0,1,3,3), (0,3,1,3), (0,3,3,3)$. Alternatively, p_{03}^3 can be calculated from P^3 by cubing the matrix.

A finite Markov Chain is *irreducible* iff its graph is strongly connected. (Every vertex i can each every vertex j by some path.) A state j in a discrete Markov chain is *periodic* if there exists integer $k > 1$ such that $\Pr[X_{t+s} = j | X_t = j] = 0$ unless s is divisible by k. Any finite, irreducible and aperiodic Markov chain is an *ergodic* chain. One reason to introduce these concepts is to find the state probability distributions that do not change after a transition, a.k.a. stationary distribution.

Definition ([MU, Definition 7.8]). *A **stationary distribution** (also called an equilibrium distribution) of a Markov Chain is a probability distribution $\boldsymbol{\pi}$ of n finite states such that $\pi_i = \sum_{j=0}^{n-1} \pi_j p_{ij}$, i.e.$\boldsymbol{\pi} = \boldsymbol{\pi} P$.*

If a chain ever reaches a stationary distribution then it maintains that distribution for all future time, and thus a stationary distribution represents a steady state or an equilibrium in the chain's behavior. This is the desired property we need from a Markov Chain. We introduce the following basic property (without proof) as a fundamental theorem of Markov chains that characterizes chains that converge to stationary distributions.

Theorem ([MU, Theorem 7.7]). *Any finite (n states), irreducible and ergodic Markov chain with transition matrix P has the following properties:*

- *The chain has a unique stationart distribution $\boldsymbol{\pi} = (\pi_0, \pi_1, \ldots, \pi_{n-1})$;*
- *For all pair of (i,j), $\lim_{t\to\infty} p_{ij}^t$ exists and independent of j.*
- *$\pi_i = \lim_{t\to\infty} p_{ij}^t$.*

Theorem ([MU, Theorem 7.10]). *Consider a finite, irreducible and ergodic Markov chain with transition matrix P. If there exists a non-negative vector $\boldsymbol{\pi} = \begin{pmatrix} \pi_0 & \pi_1 & \ldots & \pi_n \end{pmatrix}$ with n states in the state space such that $\sum_{i=0}^n \pi_i = 1$ and $\pi_i p_{ij} = \pi_j p_{ji}$ for all (i,j), then $\boldsymbol{\pi}$ is the stationary distribution corresponding to P.*

We call chains satisfies the condition $\pi_i p_{ij} = \pi_j p_{ji}$ *time reversible*. In the upcoming Assignment 2 you can prove [MU, Theorem 7.10] by yourself, by constructing the time reversible chain from scratch.

1.2 Random Walk and Markov Chain Monte Carlo

The Monte Carlo method is based on sampling. It is often difficult to generate a random sample with the required probability distribution. For example, how would you sample from a Gamma distribution? How one can generate uniform sample given a complex manifold? The MCMC method provides a very general approach to sampling from a desired probability distribution. The basic idea is to define an ergodic Markov chain whose set of states is the sample space and whose stationary distribution is the required sampling distribution. Let $X_0, X_1, \ldots, X_n$ be the Markov Chain. After r time steps, we claim state X_r will be close to stationary distribution (The bigger r, the better), so it can be used as a sample drawn from stationary distribution. Therefore, $\{X_r, X_{2r}, \ldots\}$ generates an almost independent sequence of samples from the stationary distribution of the Markov chain.

Example Task: Independent Set Let $G = (V, E)$ be a graph. An *independent set* is a set of vertices with no edges between them. Finding the maximum independent set is NP-hard.

Theorem ([MU, Theorem 6.5]). *Suppose G has n vertices and m edges. Then, the maximum independent set of G has size at least $\frac{n^2}{4m}$.*

Proof. Let $d = \frac{2m}{n}$ be the average degree of vertices in G.

(1) Delete each vertex of G (along with its incident edges) with probability $(1 - \frac{1}{d})$.

(2) For each remaining edge, remove it and one of its adjacent vertices.

The remaining vertices will form an independent set.

Let X be the number of vertices that survive step 1.2. Let Y be the number of edges that survive step 1.2. Then, $\mathrm{E}[X] = \frac{n}{d}$ and $\mathrm{E}[Y] = \frac{nd}{2}\frac{1}{d^2} = \frac{n}{2d}$. Hence, $\mathrm{E}[X - Y] = \frac{n}{d} - \frac{n}{2d} = \frac{n^2}{4m}$. □

Let Ω be the state space. The first step is to design a set of moves that ensures the state space is irreducible under the Markov chain. Denote $N(x) \subset \Omega$ as the neighborhood of $x \in \Omega$ that is one step reachable. Once the neighborhoods are established, we need to establish transition probabilities. One natural approach to try would be performing a random walk on the graph of the state space.

Theorem ([MU, Theorem 7.13]). *A random walk on a non-bipartite graph G converges to a stationary distribution $\pi_v = \frac{d(v)}{2|E|}$.*

The following lemma shows that, if we modify the random walk by giving each vertex an appropriate self-loop probability, then we can obtain a uniform stationary distribution.

Theorem ([MU, Lemma 10.7]). *Let Ω be a finite state space and $N(x)$ denote the neighbors of x. Let $M \geq max_{x\in\Omega}|N(x)|$. Consider the Markov chain where*

$$p_{xy} = \begin{cases} \frac{1}{M} & \text{if } x \neq y \text{ and } y \in N(x) \\ 0 & \text{if } x \neq y \text{ and } y \notin N(x) \\ 1 - \frac{|N(x)|}{M} & \text{if } x = y \end{cases}.$$

If this chain is irreducible and aperiodic, then the stationary distribution is uniform.

In some cases, however. we may want to sample from a chain with a nonuniform stationary distribution. The Metropolis algorithm refers to a general construction that transforms any irreducible Markov chain on a state space Ω to a time-reversible Markov chain with a required stationary distribution. The approach generalizes the idea we used before to create chains with uniform stationary distributions: add self-loop probabilities to states in order to obtain the desired stationary distribution. Now we want to construct a Markov chain on this state space with a stationary distribution π_x. As we see in the following lemma (which generalizes [MU, Lemma 10.7]), we only need the ratios between the required probabilities.

Theorem (Metropolis Algorithm [MU, Lemma 10.8]). *Let Ω be a finite state space and $N(x)$ denote the neighbors of x. Let $M \geq max_{x\in\Omega}|N(x)|$. Let $\pi_x > 0$ be the desired probability of state x in the stationary distribution. Consider the Markov chain where*

$$p_{xy} = \begin{cases} \frac{1}{M}\min(1, \frac{\pi_y}{\pi_x}) & \text{if } x \neq y \text{ and } y \in N(x) \\ 0 & \text{if } x \neq y \text{ and } y \notin N(x) \\ 1 - \sum_{y\neq x} p_{xy} & \text{if } x = y \end{cases}.$$

If this chain is irreducible and aperiodic, then the stationary distribution equal π_x.

Example: *Metropolis Hasting* method is a special case of Metropolis Algorithm that introduces *acceptance-rejection* concept when sampling over the Markov Chain. It's workflow is:

- Draws a sample x' from $Q(x'|x)$, where $x = x_t$ is the previous time snap sample and Q is the transition matrix (or function with infinite states).
- The new sample x' is accepted or rejected with some probability. This acceptance probability is:

$$A(x'|x) := \min(1, \frac{P(x')Q(x|x')}{P(x)Q(x'|x)}). \tag{1}$$

  The $P(x)$ refers to the true distribution we wish to sample from.
- If we sample $u \sim U[0,1]$, from uniform distribution and $u < A(x'|x)$, we accept the transition and set next time sample $x_{t+1} = x'$, otherwise we keep x_{t+1} same as x_t.

Example: *Gibbs Sampling* method is a special case of Metropolis Algorithm that samples each random variable of a graphical model, one at a time. The state variable $x^{(0)}$ at time 0 is first sampled from a known distribution $q(x)$. Then at each time step, the algorithm would immediately use the new value of every sample $x_i^{(t)}$ for sampling other variables $x_j^{(t)}, i \neq j$, to update the state variable which contains all graph nodes. Figure 3 shows an illustrative example. There are five nodes in Bayesian graph so each state variable consists of five Boolean values, indicating whether or not the incident happens. The probability of each event is labeled on the graph.

At time step 0, we set all events to be false, as the initial state. Assume we sample variables in the order Burglary(B),Earthquake(E),Alarm(A),JohnCalls(J) and MaryCalls(M), then at each time step, we update the table on the right from left to right accordingly. Take $t = 1$ as the example, we first generate state of B with $P(B|A, E)$. By Bayesian rule:

$$P(B|A, E) = \frac{P(A|B, E)P(B)}{P(A)} \propto P(A|B, E)P(B).$$

Therefore,

$$P(B = T|A = F, E = F) \propto P(A = F|B = F, E = F)P(B = F) = 0.0006.$$
$$P(B = F|A = F, E = F) \propto P(A = F|B = T, E = F)P(B = T) = 0.9980.$$

Similarly,

$$P(E|A, B) \propto P(A|B, E)P(E).$$

Therefore,

$$P(E = T|A = F, B = F) \propto P(A = F|B = F, E = F)P(E = F) = 0.0142.$$
$$P(E = F|A = F, B = F) \propto P(A = F|B = T, E = F)P(E = T) = 0.9970.$$

Now if we sample $E = T$ at $t = 1$, when inferring Alarm state, by bayesian rule, we obtain

$$P(A|B, E, J, M) \propto P(J|A)P(M|A)P(A|B, E).$$

Nevertheless, we shall bring the state $E = T$, instead of $E = F$ to generate sample state A at $t = 1$:

$$P(A = T|B = F, E = T, J = F, M = F) \propto P(J = F|A = F)P(M = F|A = F)(A = F|B = F, E = T) = 0.0087.$$
$$P(A = F|B = F, E = T, J = F, M = F) \propto P(J = F|A = F)P(M = F|A = F)(A = F|B = F, E = T) = 0.6678.$$

The remaining sampling procedure is similar to the steps explained above. Note the sampled state at the same time step need to be applied instantly.

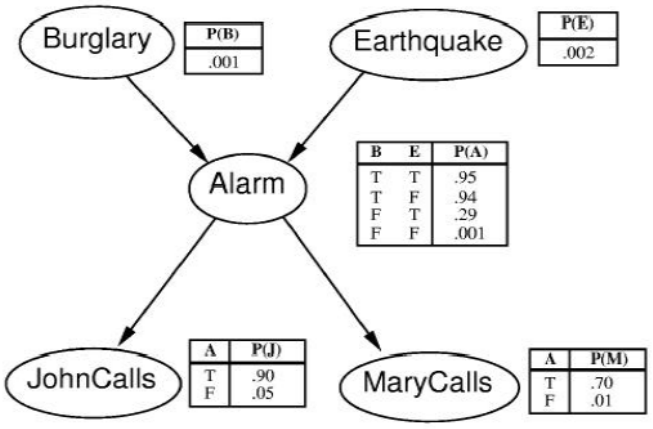
References

[MU] Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis.* Cambridge University Press, 2005.

Algorithm 1 Gibbs sampler

Initialize $x^{(0)} \sim q(x)$
for iteration $i = 1, 2, \ldots$ **do**
$x_1^{(i)} \sim p(X_1 = x_1 | X_2 = x_2^{(i-1)}, X_3 = x_3^{(i-1)}, \ldots, X_D = x_D^{(i-1)})$
$x_2^{(i)} \sim p(X_2 = x_2 | X_1 = x_1^{(i)}, X_3 = x_3^{(i-1)}, \ldots, X_D = x_D^{(i-1)})$
$\vdots$
$x_D^{(i)} \sim p(X_D = x_D | X_1 = x_1^{(i)}, X_2 = x_2^{(i)}, \ldots, X_D = x_{D-1}^{(i)})$
end for

Figure 2: Gibbs Sampling Algorithm



t	B	E	A	J	M
0	F	F	F	F	F
1	F	T	F	T	F
2	F	T	T	T	T
3	T	F	T	F	T
4	T	F	T	F	F

Figure 3: Alarm Clock System Example. The probability listed here refers to the probability of a state becomes *True*.