Markov Chain Monte Carlo
马尔可夫链蒙特卡洛

Chunpai Wang

March 2018

目录

---

1 Introduction

MCMC techniques are often applied to solve integration and optimisation problems in large dimensional spaces. For example, in Bayesian inference and learning, given some unknown variable $x \in \mathcal{X}$ and $y \in \mathcal{Y}$,

1. to obtain the posterior $p(x \mid y)$ given the prior $p(x)$ and likelihood $p(y \mid x)$, the normalization term is intractable to compute

$$p(x|y) = \frac{p(y|x)p(x)}{\int_{\chi} p\left(y|x'\right) p\left(x'\right) dx'} \tag{1}$$

2. give the joint posterior $(x, z) \in \mathcal{X} \times \mathcal{Z}$, the marginal posterior is also intractable

$$p(x|y) = \int_{z} p(x, z|y)dz \tag{2}$$

3. expectation is also intractable to compute

$$\mathbb{E}_{p(x|y)}(f(x)) = \int_{\chi} f(x)p(x|y)dx \tag{3}$$

2 Monte Carlo Simulation

Now we will introduce how Monte Carlo method could help solve the intractable integration

$$\theta = \int_{a}^{b} f(x)dx \tag{4}$$

which could be viewed as the area under the curve $f(x)$ in figure (1).

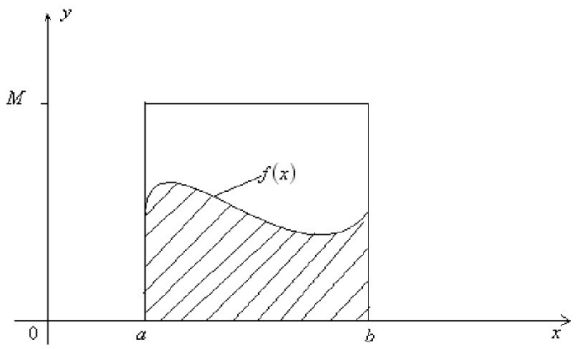

图 1

As we know the integration could be intractable, and we could resort to the Monte Carlo method to approximate it. The simplest way is to randomly sample a value $x_0 \in [a, b]$ and approximate the integration with

$$(b - a)f(x_0) \tag{5}$$

However, this method may lead to huge approximation error. We could sample multiple values $x_0, \cdots, x_{n-1} \in [a, b]$ and get a better approximation with

$$\frac{b - a}{n} \sum_{i=0}^{n-1} f(x_i) \tag{6}$$

with the assumption that x is uniformly distribution on the interval $[a, b]$. What if the x is not uniformly distributed on the interval $[a, b]$, but follows a p.d.f $p(x)$, what should we do ? If we could obtain the pdf of $p(x)$ on interval $[a, b]$, then we could still approximate the integration by

$$\theta = \int_a^b f(x)dx = \int_a^b \frac{f(x)}{p(x)}p(x)dx \approx \frac{1}{n}\sum_{i=0}^{n-1}\frac{f(x_i)}{p(x_i)} \tag{7}$$

which is the general form of Monte Carlo method. As we can see, if we assume $p(x)$ to be uniform distribution, that is $p(x_i) = \frac{1}{b-a}$, we could obtain the right hand side of formula above

$$\frac{1}{n}\sum_{i=0}^{n-1}\frac{f(x_i)}{1/(b-a)} = \frac{b-a}{n}\sum_{i=0}^{n-1}f(x_i) \tag{8}$$

which is same as formula (6).

Sampling on uniform distribution could be easily obtained by linear congruential generator (线性同余发生器), and the question remains here is how to obtain the samples from other probabilistic distribution $p(x)$.

3 Generating Samples from Probability Distribution

3.1 Inversion Method

When $p(x)$ has standard form, e.g. Gaussian, it is straightforward to sample from it using easily available routines. However, when this is not the case, we need to introduce more sophisticated techniques based on rejection sampling, importance sampling and MCMC. Here, we will show one example on negative exponential distribution to generate random samples.

The p.d.f of negative exponential distribution of random variable X could be written as

$$p(x) = \lambda e^{-\lambda x} \quad \text{for} \quad x \geq 0 \tag{9}$$

and the cumulative probability distribution (cdf) is

$$F(x) = \int_0^x \lambda e^{-\lambda x}dx = 1 - e^{-\lambda x} \quad \text{for} \quad x \geq 0 \tag{10}$$

Since the value of cdf is in $[0, 1]$, we could set a random number r (uniformly distributed between 0 and 1) equal to $F(x)$, that is

$$r = F(x) = 1 - e^{-\lambda x} \tag{11}$$

or, equivalently,

$$x = \frac{-\ln(1-r)}{\lambda} \tag{12}$$

As we can see, the equation (12) could be used for generating the sample of random variable X under the negative exponential distribution.

3.2 Rejection Sampling

Rejection sampling could be used for generating sample values for any random variable that:

1. Assumes values only within a finite range.
2. Has a p.d.f. that is bounded (i.e., does not go to infinity for any value of the random variable).

Let X be such a random variable. Let the maximum value of the p.d.f. $f_X(x)$ be denoted as c, and let X assume values in the range $[a, b]$. To generate random observations of X through the rejection method (see figure (2)):

1. Enclose the pdf $f_X(x)$ in the smallest rectangle that fully contains it and whose sides are parallel to the x and y axes. This is a $(b-a) \times c$ rectangle.
2. Using two random numbers, $R \sim Uniform(0,1)$ and $U \sim Uniform(0,1)$, and scaling each to the appropriate dimension of the rectangle [by multiplying one by $(b-a)$ and the other by c] generate a point that is uniformly distributed over the rectangle. Notice that, random variable R follows a proposal distribution $q(x)$, which is uniform distribution in our case here.
3. If this point is "below" the pdf, accept the x-coordinate of the point as an appropriate sample value of X. Otherwise, reject and return to Step 2.

The reason why this method works is quite simple. The points (x, y) obtained through the procedure of Step 2 are uniformly distributed over the area of the rectangle, $(b-a) \times c$. Therefore, for any point whose x-coordinate is between x_0 and $x_0 + dx$ (see Figure 2), we have

$$P\{\text{ point is accepted } | x_0 \le x \le x_0 + dx\} = \frac{f_X(x_0)\, dx}{c\, dx} = \frac{f_X(x_0)}{c} \tag{13}$$

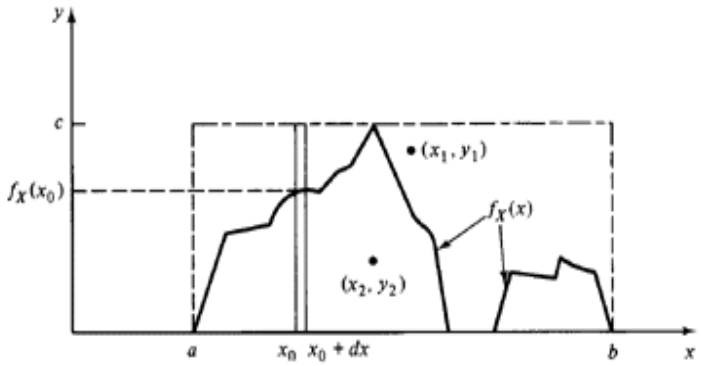

图 2: Rejection Sampling with Uniform Proposal Distribution

总结：c 是用来使得 $f_X(x)$ 总是在 $c \cdot q(x)$ 的下面，那么我们只需要先用 proposal distribution 先生一个任意的 $x^{(i)}$, 然后在 $[0, c \cdot q(x)]$ 之间随机均匀生成一个数值. 如何这个数值小于 $f_X(x)$，那么我们就把这个 $x^{(i)}$ 当作是 $f_X(x)$ 的抽样。

It is not always possible to bound $f_X(x)/q(x)$ with a reasonable constant c over the whole space X.If c is too large, the acceptance probability

$$P\{\text{ point is accepted } | x_0 \le x \le x_0 + dx\} = \frac{f_X(x_0)\, dx}{c\, dx} = \frac{f_X(x_0)}{c} \tag{14}$$

will be too small. This makes the method impractical in high-dimensional scenarios.

3.3 Importance Sampling

Suppose that our problem becomes to find $\mathbb{E}_p[f(x)]$ where p is a p.d.f on $\mathcal{D} \subseteq \mathbb{R}^d$, that is $p(x) = 0$ for all $x \notin \mathcal{D}$. Thus, we have

$$\mathbb{E}_p[f(x)] = \int_{x \in \mathcal{D}} p(x) f(x) dx \tag{15}$$

The problem here is $p(x)$ would be very difficult to sample, and we have to resort to a proposal distribution $q(x)$ which is more easier to sample. Therefore, we could directly sample on the proposal distribution to compute the $\mathbb{E}_p[f(x)]$ according to

$$\begin{aligned}
\mathbb{E}_p\left[f(x)\right] &= \int_{x \in \mathcal{D}} q(x) \frac{p(x)}{q(x)} f(x) dx && (16)\\
&= \int_{x \in \mathcal{D}} q(x) \left[\frac{p(x)}{q(x)} f(x)\right] dx && (17)\\
&= \mathbb{E}_q \left[\frac{p(x)}{q(x)} f(x)\right] && (18)
\end{aligned}$$

where $\frac{p(x)}{q(x)}$ is called the importance weight, and we should make sure $q(x) > 0$ if $p(x) > 0$.

4 Markov Chain and Limiting Distribution

The property of Markov chain is defined as

$$P(X_{t+1} = x \mid X_t, X_{t-1}, \cdots) = P(X_{t+1} = x \mid X_t) \tag{19}$$

where X_t denotes the random variable of state at time t. That means, future state at time $t+1$ is only effected by the current state at time t.

4.1 Example of Markov Chain

Sociologists typically categorize people into 3 classes based on their economic conditions: 1. lower-class, 2. middle-class, and 3. upper-class. They have found that the most important factor that determines a person ' s income class is the income class of their parents, and build the class transition probabilities as below:

$$P = \begin{bmatrix} 0.65 & 0.28 & 0.07 \\ 0.15 & 0.67 & 0.18 \\ 0.12 & 0.36 & 0.52 \end{bmatrix} \tag{20}$$

where each row sums to 1, and $P_{13} = 0.07$ is the probability of being in upper class if their parents are in lower class.

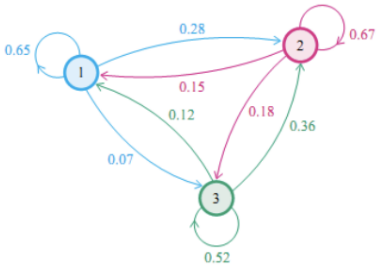

图 3: Transition Probabilities

Assume that the probability distribution of classes in current generation is denoted by $\pi_0 = [\pi_0(1), \pi_0(2), \pi_0(3)]$, the probability distribution of classes of their children is denoted by $\pi_1 = [\pi_1(1), \pi_1(2), \pi_1(3)]$, which could be computed by the transition probability matrix:

$$\pi_1 = \pi_0 P \tag{21}$$

Therefore, we could also obtain their grandchildren's distribution

$$\pi_2 = \pi_1 P = \pi_0 P^2 \tag{22}$$

and the distribution of their n^{th} generations:

$$\pi_n = \pi_{n-1} P = \pi_{n-1} P^2 = \cdots = \pi_0 P^n \tag{23}$$

Now, assuming the probability distribution for current generation is $\pi_0 = [0.21, 0.68, 0.11]$, and we could compute the $\pi_1, \cdots, \pi_n$ as below:

n^{th} Generation	0 lower-class	1 middle-class	2 upper-class
0	0.210	0.680	0.110
1	0.252	0.554	0.194
2	0.270	0.512	0.218
3	0.278	0.497	0.225
4	0.282	0.490	0.226
5	0.285	0.489	0.225
6	0.286	0.489	0.225
7	0.286	0.489	0.225
8	0.289	0.488	0.225
9	0.286	0.489	0.225
10	0.286	0.489	0.225
...	...	...	...

We find that the probability distribution stay unchanged starting from 7^{th} generation, and we call this as stationary distribution. Is this a coincident ? Let's try another initial distribution $\pi_0 = [0.75, 0.15, 0.1]$, and compute the distribution again:

n^{th} Generation	0 lower-class	1 middle-class	2 upper-class
0	0.75	0.15	0.1
1	0.522	0.347	0.132
2	0.407	0.426	0.167
3	0.349	0.459	0.192
4	0.318	0.475	0.207
5	0.303	0.482	0.215
6	0.295	0.485	0.220
7	0.291	0.487	0.222
8	0.289	0.488	0.225
9	0.286	0.489	0.225
10	0.286	0.489	0.225
...	...	...	...

We could see that the distribution become stable again starting from 9^{th} generation. The most stunning thing is, both converge to the same distribution $\pi = [0.286, 0.489, 0.225]$ with different initial distributions. We call this distribution as limiting distribution of Markov chain.

4.2 Limiting Distribution

A *limiting distribution* π, is a distribution over the states such that whatever the starting distribution π_0 is, the Markov chain converges to π. Therefore, we can conclude that the limiting distribution is determined by the transition matrix rather than the initial distribution, with the evidence:

$$P^{100} = \begin{bmatrix} 0.286 & 0.489 & 0.225 \\ 0.286 & 0.489 & 0.225 \\ 0.286 & 0.489 & 0.225 \end{bmatrix} \tag{24}$$

Theorem 1 (Ergodic Theorem of Markov Chain)**.** *We say that*

- *a Markov chain has period $k > 1$ if it can only return to its present state X_t at times $t + k, t + 2k, ...$ We say a Markov chain is aperiodic if does not have period k for any $k > 1$. That is $P_{ii} > 0$.*
- *the Markov chain is irreducible if we can get from any state to any other states (possibly in several steps). That means there exists a n such that $P_{ij}^n > 0 \ \ \forall \ i, j$.*
- *the Markov chain is positive recurrent if we are sure to come back to any state with finite expected time*

A Markov chain which aperiodic, irreducible and positive recurrent has a limiting distribution π, and

1. $\lim_{n\to\infty} P^n = \begin{bmatrix} \pi(1) & \pi(2) & \cdots & \pi(j) & \cdots \\ \pi(1) & \pi(2) & \cdots & \pi(j) & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \pi(1) & \pi(2) & \cdots & \pi(j) & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix}$

2. $\pi_j = \sum_{i=1}^{\infty} \pi(i) P_{ij}$

3. *π is the only non-negative solution of $\pi P = \pi$*

5 MCMC

5.1 Intuition

Recall that X_t denotes the random variable of state at time t, and follows the distribution π_t. By the ergodic theorem, we have

$$\begin{aligned} X_0 &\sim \pi_0(x) \\ X_1 &\sim \pi_1(x) \\ X_2 &\sim \pi_2(x) \\ &\vdots \\ X_n &\sim \pi_n(x) = \pi(x) \\ X_{n+1} &\sim \pi(x) \\ X_{n+2} &\sim \pi(x) \\ &\vdots \end{aligned}$$

where $X_n, X_{n+1}, \cdots$ are random variables following the same distribution $\pi(x)$, but not independent. Now, assuming starting from a specific initial state x_0, with certain transition matrix P, we could obtain the next following states $x_1, \cdots, x_n, x_{n+1}, \cdots$. Moreover, the $x_n, x_{n+1}, \cdots$ can be viewed as the samples of stationary distribution $\pi(x)$.

Now, you may see a way to generate samples from any distribution $p(x)$. That is, if we could create a Markov train with a transition matrix P that leads to limiting distribution which is $p(x)$, then no matter what initial distribution we start with, we will eventually generate a sequence of states $x_0, \cdots, x_n, x_{n+1}, x_{n+2}, \cdots$. If the Markov chain converge to the stationary distribution $\pi(x) = p(x)$ at time n, then we obtain the samples of $p(x)$: $x_n, x_{n+1}, \cdots$

5.2 Detailed Balance Condition

The question is how to create a transition matrix that will lead to the desired limiting distribution. We will leverage the *detailed balance condition.*

Theorem 2 (Detailed Balance Condition)**.** *Given the transition matrix Q for a Markov chain and a distribution $p(x)$, if the Markov chain is aperiodic, irreducible and positive recurrent and satisfies*

$$p(i)Q(i,j) = p(j)Q(j,i) \quad \forall \quad i, j \tag{25}$$

then $p(x)$ is the stationary distribution of this Markov chain.

证明.

$$\sum_{i=1}^{\infty} p(i)Q(i,j) = \sum_{i=1}^{\infty} p(j)Q(j,i) = p(j)\sum_{i=1}^{\infty} Q(j,i) = p(j) \tag{26}$$

Therefore, we have $pQ = p$. Since p is the solution of $pQ = p$, we conclude that p is the stationary distribution of Markov chain with transition matrix Q. □

Now it is straightforward that for generating samples from any distribution $p(x)$, we could achieve it by creating a transition matrix Q of Markov chain that satisfies the detailed balance condition. Denote $\beta(i, j)$ or $\beta(i \to j)$ or $\beta(j \mid i)$ by the transition probability from state i to state j in a random transition matrix β. Of course, the random transition matrix β will not satisfies the detailed balance condition in general, that is

$$p(i)\beta(i, j) \neq p(j)\beta(j, i) \tag{27}$$

However, we may observe that it is possible to modify the Markov chain by introducing additional probability $\alpha(i, j)$ to make the following condition holds:

$$p(i)\beta(i, j)\alpha(i, j) = p(j)\beta(j, i)\alpha(j, i) \tag{28}$$

where the most simple choice of $\alpha(i, j)$ and $\alpha(j, i)$ are:

$$\alpha(i, j) = p(j)\beta(j, i) \tag{29}$$

$$\alpha(j, i) = p(i)\beta(i, j) \tag{30}$$

Therefore, the transition matrix Q that makes our target distribution $p(x)$ as the limiting distribution satisfies:

$$Q(i, j) = \beta(i, j)\alpha(i, j) \tag{31}$$

5.3 General MCMC Sampling

The probability $\alpha(i, j)$ that transforms the random Markov chain β could be interpreted as the acceptance rate as in the rejection-sampling method. That means, when state i transits to state j with original probability $\beta(i, j)$, we add an additional probability $\alpha(i, j)$ to accept the transition. Therefore, we could summarize the general MCMC sampling method as below:

Algorithm 1: MCMC Sampling

Input: a random proper transition matrix β, the target distribution $p(x)$, number of samples M

Output: M samples from distribution $p(x)$

generate an initial state $X_0 = x_0$
for $t = 0$ *to* $N + M - 1$ **do**
  sample $y \sim \beta(x_t, x)$
  sample $\mu \sim Uniform(0, 1)$
  if $\mu < \alpha(x_t, y) = p(y)\beta(y, x_t)$ **then**
    accept the transition $x_t \to y$, and $X_{t+1} = y$
  else
    reject the transition, and $X_{t+1} = x_t$
  end
end
return *the sequence of states at time* $t \geq N$.

5.4 Metropolis-Hastings

One potential issue of MCMC sampling is the acceptance probability $\alpha(i, j)$ could be very small, which leads to extremely slow convergence to stationary distribution, that mean N will be very large. How can we resolve this issue ?

Assume $\alpha(i, j) = 0.1$ and $\alpha(j, i) = 0.2$, and the detailed balance condition is satisfied:

$$p(i)Q(i, j) \times 0.1 = p(j)Q(j, i) \times 0.2, \tag{32}$$

if we times both sides with 5, the detailed balance condition still holds:

$$p(i)Q(i, j) \times 0.5 = p(j)Q(j, i) \times 1, \tag{33}$$

but the acceptance probabilities have been increased from 0.1 to 0.5 and 0.2 to 1, respectively. Therefore, we could simply magnify both $\alpha(i, j)$ and $\alpha(j, i)$ by a constant c such that

$$c \cdot \max(\alpha(i, j), \alpha(j, i)) = 1 \tag{34}$$

Algorithm 2: Metropolis-Hastings

Input: a random proper transition matrix β, the target distribution $p(x)$, number of samples M

Output: M samples from distribution $p(x)$

generate an initial state $X_0 = x_0$
for $t = 0$ *to* $N + M - 1$ **do**
  sample $y \sim \beta(x_t, x)$
  sample $\mu \sim Uniform(0, 1)$
  if $\mu < \alpha(x_t, y) = \min\left\{\frac{p(y)\beta(y,x_t)}{p(x_t)\beta(x_t,y)}, 1\right\}$ **then**
    accept the transition $x_t \to y$, and $X_{t+1} = y$
  else
    reject the transition, and $X_{t+1} = x_t$
  end
end
return *the sequence of states at time* $t \geq N$.

5.5 Gibbs Sampling

There are some challenges when applying Metropolis-Hastings to high dimensional space: 1) it is time-consuming to compute the acceptance rate in high dimensional space, 2) some acceptance rates are always less than 1, which may requires many iterations to converge to stationary distribution, and 3) sometimes, the joint probability of high dimensional features is difficult to compute than the conditional distribution.

Now, we are looking for a way to solve those 3 issues. For example, in 2-dimensional case, we have probability distribution $p(x, y)$. That means we use 2 random variables to represent one state. Now consider two states or two points $A = (x_1, y_1)$ and $B = (x_1, y_2)$ in the line $x = x_1$, we have

$$p(x_1, y_1)\, p(y_2|x_1) = p(x_1)\, p(y_1|x_1)\, p(y_2|x_1) \tag{35}$$

$$p(x_1, y_2)\, p(y_1|x_1) = p(x_1)\, p(y_2|x_1)\, p(y_1|x_1) \tag{36}$$

and

$$p\left(x_1, y_1\right) p\left(y_2 | x_1\right) = p\left(x_1, y_2\right) p\left(y_1 | x_1\right) \tag{37}$$

That is

$$p(A) p\left(y_B | x_1\right) = p(B) p\left(y_A | x_1\right) \tag{38}$$

Recall the detail balance condition, we would like to find a transition matrix Q such that

$$p(A) Q(A \rightarrow B) = p(B) Q(B \rightarrow A) \tag{39}$$

We can easily find this transition probability, that is

$$Q(A \rightarrow B) = p(y_B \mid x_1) \quad \text{if} \quad x_A = x_B = x_1 \tag{40}$$

$$Q(B \rightarrow A) = p(y_A \mid x_1) \quad \text{if} \quad x_A = x_B = x_1 \tag{41}$$

Similarly, consider two states $A = (x_1, y_1)$ and $C = (x_2, y_1)$ in the line $y = y_1$, we could set the transition probability as

$$Q(A \rightarrow C) = p(x_C \mid y_1) \quad \text{if} \quad y_A = y_C = y_1 \tag{42}$$

$$Q(C \rightarrow A) = p(x_A \mid y_1) \quad \text{if} \quad y_A = y_C = y_1 \tag{43}$$

If two states $A = (x_1, y_1)$ and $D = (x_2, y_2)$ are not in the line which is parallel to $x - axis$ or $y - axis$, then we could simply set

$$Q(A \rightarrow D) = Q(D \rightarrow A) = 0 \quad \text{if} \quad y_A \neq y_D \ \text{ and } \ x_A \neq x_D \tag{44}$$

Hence, according to the detailed balance condition, the transition matrix Q we have constructed could lead to stationary distribution $p(x, y)$.

The idea could be easily extend to the higher dimensional cases. For higher dimensional case, we could simply change x_1 to $\mathbf{x}_1$, and we will find that detailed balance condition still holds:

$$p\left(\mathbf{x}_1, y_1\right) p\left(y_2 | \mathbf{x}_1\right) = p\left(\mathbf{x}_1, y_2\right) p\left(y_1 | \mathbf{x}_1\right) \tag{45}$$

and we could construct the transition matrix Q in the similar way. That is, if a state $A = (x_1, \cdots x_i, \cdots, x_n)$ transits along the axis x_i to another state B, then the transition probability could be defined as $p(x_i \mid x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n)$. Otherwise, the transition probability is set to 0.

Algorithm 3: Gibbs Sampling

Input: the target distribution $p(x_1, \cdots, x_n)$, number of samples M

Output: M samples from distribution $p(x_1, \cdots, x_n)$

generate an initial state $(X_1^{(0)}, \cdots, x_n^{(0)})$

for $t = 0$ *to* $N + M - 1$ **do**

  sample $x_1^{t+1} \sim P(x_1 \mid x_2^t, \cdots x_n^t)$

  sample $x_2^{t+1} \sim P(x_2 \mid x_1^{t+1}, x_3^t, \cdots x_n^t)$

  $\vdots$

  sample $x_{n-1}^{t+1} \sim P(x_{n-1}^{t+1} \mid x_2^{t+1}, \cdots, x_{n-2}^{t+1}, x_n^t)$

  sample $x_n^{t+1} \sim P(x_n \mid x_2^{t+1}, \cdots x_{n-1}^{t+1})$

end

return *the sequence of states* $(x_1^t, x_2^t, \cdots, x_n^t)$ *at time* $t \geq N$.

参考文献

[1] https://statweb.stanford.edu/ owen/mc/ch-var-is.pdf.

[2] https://towardsdatascience.com/from-scratch-bayesian-inference-markov-chain-monte-carlo-and-metropolis-hastings-in-python-ef21a29e25a.

[3] https://towardsdatascience.com/markov-chain-monte-carlo-in-python-44f7e609be98.

[4] https://www.cnblogs.com/pinard/p/6625739.html.