Markov Chains and Algorithmic Applications: WEEKS 12&13

1 Exact Simulation

In the previous lectures, we saw several procedures to generate samples from a distribution π on S. Moreover, the Metropolis-Hastings procedure allows us to do so with only partial knowledge of π, but this approach is merely approximate, as we do not know how long to run the algorithm to reach the stationary distribution π. In other words, the best we can do is upper-bound the mixing-time T_ϵ so as to sample from a distribution that is arbitrarily close to π.

Today, we will see another procedure named *coupling from the past* (introduced by James Propp and David Wilson in 1996) which allows us to sample *exactly* from π. In order to study this method, we need a tool called *random mapping representation of a Markov chain.*

1.1 Random Mapping Representation

So far we used to define a (time-homogeneous) Markov chain by a matrix of transition probabilities $P = [p_{i\to j}]$ where $p_{i\to j} = \mathbb{P}(X_{n+1} = j | X_n = i)$. Alternatively one can represent a Markov chain as

$$X_{n+1} = \Phi(X_n, U_{n+1})$$

where $\Phi(\cdot,\cdot)$ is a cleverly chosen function and $(U_n, n \geq 1)$ is a sequence of i.i.d. random variables

Proposition 1.1. Every Markov chain has a random mapping representation.

Proof. We assume the U_n's are uniform random variables in $[0,1]$ (we denote this as $U_n \sim \mathcal{U}[0,1]$) and construct $\Phi(\cdot,\cdot)$ such that $p_{i\to j} = \mathbb{P}\left(X_{n+1} = j | X_n = i\right) = \mathbb{P}\left(\Phi(i, U_{n+1}) = j\right)$ for any arbitrary set of transition probabilities $p_{i\to j}$.

Define

$$F_{i\to k} \triangleq \sum_{j=1}^{k} p_{i\to j}, \qquad \forall i,k \in S$$

(where S is the state space) and set

$$\Phi(i,u) \triangleq \sum_{j\in S} j \cdot \mathbb{1}\left\{F_{i\to j-1} < u \leq F_{i\to j}\right\}.$$

We hence have

$$\mathbb{P}(\Phi(i,U_{n+1}) = j) = \mathbb{P}(F_{i\to j-1} < U_{n+1} \leq F_{i\to j}) = F_{i\to j} - F_{i\to j-1} = p_{i\to j}. \qquad \square$$

Remark: In general, there may exist many different random mapping representations for a particular chain. In the above proof we just constructed *one of* these representations.

1.2 Forward Coupling

Suppose we take two copies of a Markov chain X_n and Y_n having stationary distribution π. Their random mapping representations are

$$X_{n+1} = \Phi(X_n, U_{n+1})$$
$$Y_{n+1} = \Phi(Y_n, U_{n+1}).$$

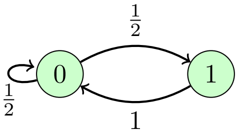


Figure 1: Markov chain of Example **??**

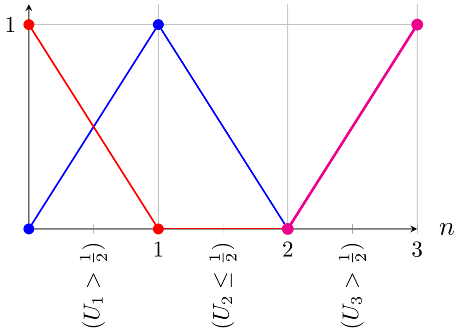


Figure 2: Two copies of the chain considered in Example **??**.

In general, the U_n's used in the chains X_n and Y_n are two independent samples. However, *if we use the same samples U_{n+1} for updating $X_n \to X_{n+1}$ and $Y_n \to Y_{n+1}$*, we will impose *grand coupling* between those chains.

Now suppose we start $|S|$ copies of the chain $\left(X_n^{(i)}, i \in S\right)$, each starting at a different state (i.e. $X_0^{(i)} = i$), and update them using the same samples of U_n (i.e. we establish pairwise grand coupling). This situation is called *forward coupling*.

One may think that once all chains coalesce at some time $T^* > 0$, the initial state has been "forgotten", so that the stationary distribution has been reached (i.e. $\forall i, j \in S,\ \ \mathbb{P}\left(X_{T^*}^{(i)} = j\right) = \mathbb{P}\left(X_{T^*} = j\right) = \pi_j$). Unfortunately, this is not the case as we will see in the following examples:

Example 1.2. Consider the Markov chain of Figure **??**. A random mapping representation of this chain (using $U_n \sim \mathcal{U}[0,1]$) is

$$\Phi(0,u) = \begin{cases} 0 & \text{if } u \leq \frac{1}{2}, \\ 1 & \text{if } u > \frac{1}{2}, \end{cases}$$
$$\Phi(1,u) = 0.$$

It is easy to check that coalescence always happens at state 0. Indeed, the only way to get $\Phi(0,u) = \Phi(1,u)$ is to have $0 \leq u \leq \frac{1}{2}$ implying $\Phi(0,u) = \Phi(1,u) = 0$. For example, consider the situation depicted in Figure **??**. That is to say, $(\pi_0\,(T_c) = 1, \pi_1\,(T_c) = 0)$ where T_c is the coalescence time. However, the stationary distribution is $\left(\pi_0 = \frac{2}{3}, \pi_1 = \frac{1}{3}\right)$. Therefore, the chains are not in the stationary distribution when they coalesce. They are also not in the stationary distribution after the coalescence time.

The choice of the random mapping representation can even lead to situations where we do not have coalescence at all.

Example 1.3. Consider the Markov chain of Figure **??**. One possible candidate for its random mapping representation (still assuming $U_n \sim \mathcal{U}[0,1]$) is

$$\Phi(i,u) = \begin{cases} i & \text{if } u \leq \frac{1}{3}, \\ 1-i & \text{if } u > \frac{1}{3} \end{cases}$$

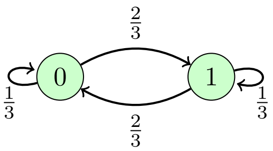
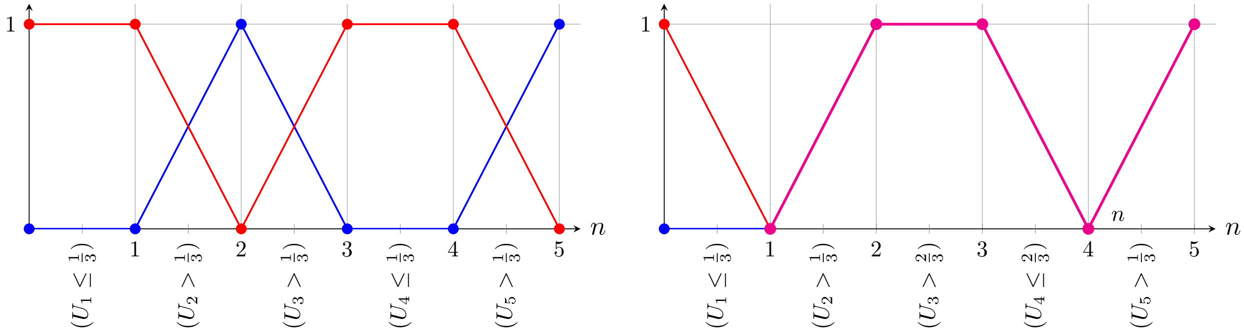


Figure 3: Markov chain of Example **??**



(a) Coalescence does not happen.

(b) Coalescence happens fast.

Figure 4: The choice of random mapping representation can change the coalescence time.

Using this mapping, the two chains will never coalesce (see Figure **??** for example).

However, if we pick another random mapping representation

$$\Phi(0,u) = \begin{cases} 0 & \text{if } u \leq \frac{1}{3}, \\ 1 & \text{if } u > \frac{1}{3}, \end{cases} \quad \Phi(1,u) = \begin{cases} 0 & \text{if } u \leq \frac{2}{3}, \\ 1 & \text{if } u > \frac{2}{3}, \end{cases}$$

with the same realization of the U_n's, coalescence will take place in a few steps (see Figure **??**).

1.3 Coupling From The Past

Surprisingly perhaps, a slight modification of the idea of forward coupling leads to a criterion to check whether the chain is in the stationary distribution. The idea is called *coupling from the past* and is formalized as follows (the algorithm is known as the Propp-Wilson algorithm):

1. Generate once and for all $(U_{-n}, n \geq 1)$.
2. Set $T_0 = -1$.
3. Start the experiment at all states $i \in S$ at time T_0 and update $X_{n+1}^{(i,T_0)} = \Phi\left(X_n^{(i,T_0)}, U_{n+1}\right)$ for $n = T_0, T_0+1, \ldots, -1$ ($X_n^{(i,T_0)}$ denotes the state of the chain at time n knowing that $X_{T_0} = i$).
4. Check coalescence at time $n = 0$: If $X_0^{(i,T_0)}$ is independent of i (i.e. $\forall i,j \in S,\ \ X_0^{(i,T_0)} = X_0^{(j,T_0)}$), $X_0^{(i,T_0)}$ is the output and the algorithm terminates. If not, set $T_0 \leftarrow T_0 - 1$ and return to step 2.

We will see that the distribution of $X_0^{(i,T_0)}$ is *exactly* the stationary distribution of the Markov chain.

Let us define the event

$$A_{T_1,T_2} = \left\{ X_{T_2}^{(i,T_1)} = X_{T_2}, \forall i \in S \right\} = \{\text{all chains started at time } T_1 \text{ have coalesced at time } \leq T_2\}.$$

Theorem 1.4. Let X_n be a Markov chain having stationary distribution π. If $\exists L > 0$ such that $\mathbb{P}\left(A_{0,L}\right) > 0$[1], then

1. With probability 1, the Propp-Wilson algorithm outputs a value X_0 in finite time.
2. $X_0 \sim \pi$.

Proof. We first need to check that the algorithm terminates in finite time with probability 1. To this end, we will show:

$$\mathbb{P}\left(\bigcup_{k\geq 1} A_{-kL}^{-(k-1)L}\right) = 1.$$

Because of Markovity, the events $A_{-kL}^{-(k-1)L}$ are i.i.d. Hence,

$$\begin{aligned}\mathbb{P}\left(\bigcup_{k\geq 1} A_{-kL}^{-(k-1)L}\right) &= 1 - \mathbb{P}\left(\bigcap_{k\geq 1} \overline{A_{-kL}^{-(k-1)L}}\right)\\ &= 1 - \prod_{k\geq 1} \mathbb{P}\left(\overline{A_{-kL}^{-(k-1)L}}\right)\\ &= 1 - \lim_{n\to\infty}\left(1 - \mathbb{P}\left(A_0^L\right)\right)^n = 1,\end{aligned}$$

since $\mathbb{P}\left(A_0^L\right) > 0$ by assumption.

It remains to show that $X_0^{(i,T_0)} \sim \pi$ (when the algorithm stops). Let $T_0 = -\inf\left\{n > 0,\ X_0^{(i,-n)} = X_0\ \ \forall i \in S\right\}$ and assume $X_0 \sim \mu$. We will show that $\mu = \pi$:

We have $X_0^{(i,T_0)} \sim \mu$, so $X_1^{(i,T_0)} \sim \mu P$, and by time-homogeneity $X_1^{(i,T_0)} \sim X_0^{(i,T_0-1)}$. Moreover, since T_0 is the minimum coalescence time, we have

$$X_0^{(i,T_0-1)} = X_0^{(j,T_0)} = X_0,\ \ \forall i, j \in S$$

Hence $\mu = \mu P$ and therefore $\mu = \pi$.

□

1.3.1 Propp-Wilson Algorithm in Practice

In the Propp-Wilson algorithm depicted above, T_0 is replaced by $T_0 - 1$ at each iteration, so finding the coalescence time is a linear-time operation which is too slow in practice. By replacing T_0 with $2T_0$ instead, finding the coalescence time becomes a logarithmic-time operation.

Moreover, at first glance, the Propp-Wilson algorithm seems to be useless when the state space is huge: we need to keep track of $|S|$ copies of the chain in order to generate one sample distributed according to π. For example, the Ising model would require running 2^N chains simultaneously, which is not practically feasible. However, if we have

1. A partial ordering in the state space,
2. A random mapping representation that preserves this ordering:
$$i \preceq j \Longrightarrow \Phi(i,u) \preceq \Phi(j,u) \quad \forall i, j \in S,$$
3. Two extremal states $\underline{i}$ and $\bar{i}$ such that $\underline{i} \preceq j \preceq \bar{i},\ \forall j \in S$,

[1] If the chain is ergodic, then it is more than reasonable to assume that there exists a mapping Φ satisfying this assumption.

then we only need to keep track of the *two* chains $X^{(\underline{i},T_0)}$ and $X^{(\bar{i},T_0)}$. Indeed, under this assumption, all intermediary chains remain "sandwiched" between the two extreme chains as time goes by, so that at the time these two coalesce, all the others must have coalesced also. This is called *monotone coupling from the past.*

Example 1.5. Consider the classical random walk on $S = \{0, 1, \dots, N\}$, with transition probabilities

$$p_{00} = p_{01} = p_{N,N} = p_{N,N-1} = \frac{1}{2} \quad \text{and} \quad p_{i,i\pm 1} = \frac{1}{2} \quad \text{for } i = 1, \dots, N-1$$

and consider the following random mapping representation (with U_n i.i.d.$\sim \mathcal{U}[0,1]$):

$$\begin{cases} \text{if } u \leq \frac{1}{2}: \quad \Phi(0,u) = 0, \quad \Phi(i,u) = i-1 \quad \text{for } i = 1, \dots, N \\ \\ \text{if } u > \frac{1}{2}: \quad \Phi(N,u) = N, \quad \Phi(i,u) = i+1 \quad \text{for } i = 0, \dots, N-1 \end{cases}$$

(Note that with such a mapping, the chain only possibly coalesces in states 0 or N). It is the case that this random mapping representation is monotone. So in this case, it sufficient to run the Propp-Wilson algorithm from initial states 0 and N only. Of course, there are more interesting examples one might be interested in! In the next paragraph we take up the Glauber dynamics for the Ising model.

1.4 Application to the Ising model

For the Ising model we can define an ordering on the state space which is preserved during the Glauber dynamics (in a random mapping representation of this dynamics). We discuss here this interesting application.

Recall the state space is the set of spin assignments $(\sigma_1, \dots, \sigma_N) = \underline{\sigma}$ with $\sigma_v \in \{-1, +1\}$. The *partial* order is

$$\underline{\sigma} \preceq \underline{\sigma}' \Leftrightarrow \sigma_v \leq \sigma_v'$$

for all $v = 1, \dots, N$. In particular $\underline{\sigma}_{\text{min}} = (-1, \dots, -1)$ is the "smaller" of all assignments and $\underline{\sigma}_{\text{max}} = (+1, \dots, +1)$ is the "larger" of all assignments.

We want the random mapping representation of the MCMC chain to preserve this ordering. This means

$$\underline{\sigma} \preceq \underline{\sigma}' \Rightarrow \Phi(\underline{\sigma}, U) \preceq \Phi(\underline{\sigma}', U).$$

for some appropriate random variable U. We say that such a mapping is monotone and that it leads to a monotone CFTP.

Then in the monotone CFTP algorithm of Propp and Wilson all trajectories are sandwiched between the trajectory emanating from $\underline{\sigma}_{\text{min}}$ and $\underline{\sigma}_{\text{max}}$. Therefore to check coalescence it is enough to check for coalescence for these two trajectories (instead of the 2^N trajectories corresponding to all spin assignments).

Claim: *The Glauber dynamics (or Gibbs sampler) for the Ising ferromagnetic model is monotone.*

We check the claim. Recall the probability distribution (of Gibbs) is

$$\pi(\underline{\sigma}) = \frac{1}{Z} \exp(\beta \sum_{v,w=1}^{N} J_{vw} \sigma_v \sigma_w + \sum_{v=1}^{N} h_v \sigma_v)$$

and the Gibbs sampler reduces to:

- Initialize at $\underline{\sigma}$
- Select vertex $v \in \{1, \dots, N\}$ uniformly at random.

- Update $\underline{\sigma} \to \underline{\sigma}'$ where $\sigma'_w = \sigma_w$ for $w \neq v$ and $\sigma'_v = \pm 1$ with probability

$$\frac{1}{2}(1 \pm \tanh\{\sum_w \beta J_{vw}\sigma_w + \beta h_v\}).$$

A random mapping representation of the update is $\underline{\sigma}' = \Phi(\underline{\sigma}, U)$ where $U = (v, u)$ where v is uniform random in $\{1, \dots, N\}$, u is a uniform r.v over the real interval $[0, 1]$, and

$$\Phi(\underline{\sigma}, U) = \begin{cases} \sigma'_w = \sigma_w, w \neq v, \\ \sigma'_v = +1, \text{ for } 0 \leq u \leq \frac{1}{2}(1 + \tanh\{\sum_w \beta J_{vw}\sigma_w + \beta h_v\}). \\ \sigma'_v = -1, \text{ for } \frac{1}{2}(1 + \tanh\{\sum_w \beta J_{vw}\sigma_w + \beta h_v\}) < u \leq 1. \end{cases}$$

This is in fact how you could implement a numerical simulation of this dynamics.

To check monotonicity we have to show that $\underline{\sigma} \preceq \underline{\tau} \Rightarrow \underline{\sigma}' \preceq \underline{\tau}'$. Select v and u. Certainly for $w \neq v$ we have $\sigma_w \leq \tau_w \Rightarrow \sigma'_w \leq \tau'_w$ since $\sigma_w = \sigma'_w$ and $\tau_w = \tau'_w$.

For $w = v$ we proceed as follows. Note that the generated v is the same for the two states $\underline{\sigma}$ and $\underline{\tau}$. So if $0 \leq u < \frac{1}{2}(1 + \tanh\{\sum_w \beta J_{vw}\sigma_w + \beta h_v\})$ for $J_{vw} > 0$ (ferromagnetic model) then it is also true that $u \leq \frac{1}{2}(1 + \tanh\{\sum_w \beta J_{vw}\tau_w + \beta h_v\})$. Therefore we get $\sigma'_v = +1$ and $\tau'_v = +1$, i.e. $\sigma'_v \leq \tau'_v = +1$. Finally for $\frac{1}{2}(1 + \tanh\{\sum_w \beta J_{vw}\sigma_w + \beta h_v\}) < u \leq 1$ then $\sigma'_v = -1$ so it is anyway true that $\sigma'_v \leq \tau'_v$ whatever is the update τ'.

This ends the check of the claim.