MCMC notes by Mark Holder

Bayesian inference

Ultimately, we want to make probability statements about true values of parameters, given our data. For example $\mathbb{P}(\alpha_0 < \alpha_1|X)$. According to Bayes' theorem:

$$\mathbb{P}(\theta|X) = \frac{\mathbb{P}(X|\theta)\mathbb{P}(\theta)}{\mathbb{P}(X)}$$

It is often the case that we cannot calculate $\mathbb{P}(X)$, the marginal probability of the data.

Markov chain Monte Carlo is (by far), the dominant computational approach to conducting Bayesian inference. The "Monte Carlo" part of the name reflects the fact that we are going to perform a random simulation – the algorithm uses a pseudo-random number generator to explore parameter space. The "Markov chain" part of the name refers to fact that we are going to conduct the simulation by iteratively updating the state of the simulator using rules that depend only on the current state of the system. This type of stochastic system (one which "forgets" its history) is referred to as a Markov chain.

Markov chains

A Markov chain connects a series of states. You can think of it as a walk through a "state space".

- a *state* is the complete description of the "value" of the chain as it walks. We'll be walking through parameter space, so the state space is the space of all parameter values in our model. Typically the state space is continuous, but I'll explain most of the theory in problems with discrete state space.
- an *index* refers to which step of the chain we are currently at. There are many uses for continuous-time Markov chains (in which the index can assume any numerical value), but the Markov chains that we will be using in MCMC are discrete time. This means that the index is simply a count of the number iterations that the algorithm has been running. The index is often referred to as the "iteration," the "generation," or the "step number." In a discrete time Markov chain, you update the state once per interval (although you can remain in the same state and still consider that an update).

There is a huge literature on Markov processes, but there are only a few crucial aspects of Markov chains that we need to understand if we want to understand MCMC.

- Under mild conditions, a Markov chain will converge to a stationary distribution (also called a steady-state distribution) after a large number of steps. This distribution will be the same regardless of the starting point of the chain.
- Markov chains that have a high probability of changing state in a single step will converge to the stationary distribution faster.

Transition probabilities

A Markov chain can be defined by describing the full set of probability statements that define the rules for the state of the chain in the next step (step $i+1$) given its current state (in step i). These transition probabilities are analogous to transition rates if we are working with continuous-time Markov processes.

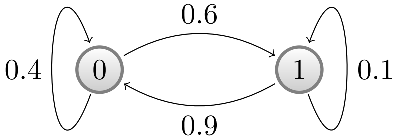


Figure 1: A graphical depiction of a two-state Markov process.

Consider the simplest possible Markov chain: one with two states (0, and 1) that operates in discrete time. The figure to the right shows the states in circles. The transition probabilities are shown as arcs connecting the states with the probabilities next to the line. The full probability statements that correspond to the graph are:

$$\begin{aligned}
\mathbb{P}(x_{i+1} = 0|x_i = 0) &= 0.4 \\
\mathbb{P}(x_{i+1} = 1|x_i = 0) &= 0.6 \\
\mathbb{P}(x_{i+1} = 0|x_i = 1) &= 0.9 \\
\mathbb{P}(x_{i+1} = 1|x_i = 1) &= 0.1
\end{aligned}$$

Note that, because these are probabilities, some of them must sum to one. In particular, if we are in a particular state at step i we can call the state x_i. In the next step, we must have *some* state so $1 = \sum_j \mathbb{P}(x_{i+1} = j|x_i)$ for every possible x_i.

Note that the state at step $i+1$ only depends on the state at step i. This is the Markov property. More formally we could state it as:

$$\mathbb{P}(x_{i+1}|x_i) = \mathbb{P}(x_{i+1}|x_i, x_{i-k})$$

where k is positive integer. What this probability statement is saying is that, conditional on x_i, the state at $i+1$ is independent on the state at any point before i. So when working with Markov chains we don't need to concern ourselves with the full history of the chain, merely knowing the state at the previous step is enough.[1]

Clearly if we know x_i and the transition probabilities, then we can make a probabilistic statement about the state in the next iteration (in fact the transition probabilities *are* these probabilistic statements). But we can also think about the probability that the chain will be in a particular state two steps form now:

$$\begin{aligned}
\mathbb{P}(x_{i+2} = 0|x_i = 0) &= \mathbb{P}(x_{i+2} = 0|x_{i+1} = 1)\mathbb{P}(x_{i+1} = 1|x_i = 0) + \mathbb{P}(x_{i+2} = 0|x_{i+1} = 0)\mathbb{P}(x_{i+1} = 0|x_i = 0) \\
&= 0.9 * 0.6 + 0.4 * 0.4 \\
&= 0.7
\end{aligned}$$

Here we are exploiting the fact that the same "rules" (transition probabilities) apply when we consider state changes between $i+1$ and $i+2$. If the transition probabilities are fixed through the running of the Markov chain, then we are dealing with a time-homogeneous Markov chain.

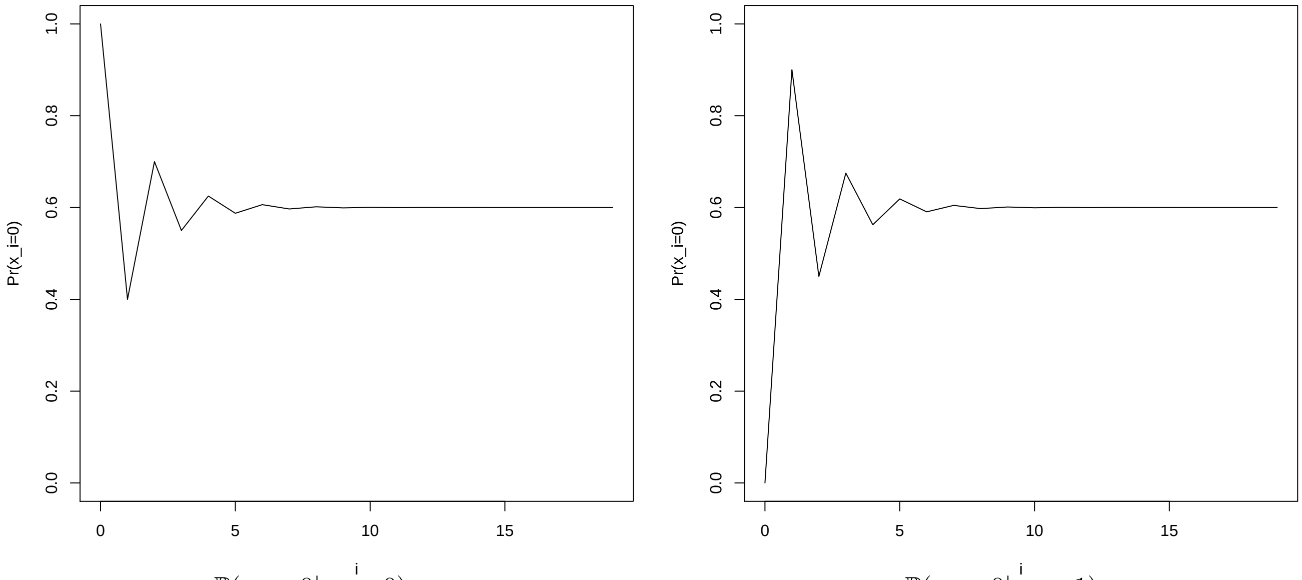
[1] There are second-order Markov processes that depend on the two previous states, and third-order Markov process *etc.*. But in this course, we'll just be dealing with the simplest Markov chains which only depend on the current state.

Note that:

$$\begin{aligned}
\mathbb{P}(x_{i+2}=1|x_i=0) &= \mathbb{P}(x_{i+2}=1|x_{i+1}=1)\mathbb{P}(x_{i+1}=1|x_i=0)+\mathbb{P}(x_{i+2}=1|x_{i+1}=0)\mathbb{P}(x_{i+1}=0|x_i=0) \\
&= 0.1*0.6+0.6*0.4 \\
&= 0.3
\end{aligned}$$

So, if we sum over all possible states at $i+2$, the relevant probability statements sum to one.

It gets tedious to continue this for a large number of steps into the future. But we can ask a computer to calculate the probability of being in state 0 for a large number of steps into the future by repeating the calculations (see the Appendix A for code). Figure (2) shows the probabilities of the Markov process being in state 0 as a function of the step number for the two possible starting states.



$\mathbb{P}(x_i=0|x_0=0)$ $\mathbb{P}(x_i=0|x_0=1)$

Figure 2: The probability of being in state 0 as a function of the step number, i, for two different starting states (0 and 1) for the Markov process depicted in Figure (1).

Note that the probability stabilizes to a steady state distribution. Knowing whether the chain started in state 0 or 1 tells you very little about the state of the chain in step 15. Technically, x_{15} and x_0 are *not* independent of each other. If you work through the math the probability of the state at step 15 does depend on the starting state:

$$\begin{aligned}
\mathbb{P}(x_{15}=0|x_0=0) &= 0.599987792969 \\
\mathbb{P}(x_{15}=0|x_0=1) &= 0.600018310547
\end{aligned}$$

But clearly these two probabilities are very close to being equal.

If we consider an even larger number of iterations (e.g. the state at step 100), then the probabilities are so close that they are indistinguishable.

After a long enough walk, a Markov chain in which all of the states are connected[2] will *converge* to its stationary distribution.

Many chains or one really long chain

When constructing arguments about Markov chains we often flip back and forth between thinking about the behavior of an arbitrarily large number of instances of the random process all obeying the same rules (but performing their walks independently) versus the behavior we expect if we sparsely sample a very long chain. The idea is that if sample sparsely enough, the sampled points from one chain are close to being independent of each other (as Figure 2 was meant to demonstrate). Thus we can think of a very long chain as if we had a large number of independent chains.

Stationary distribution

Notice in the discussion above that the probability of ending up in state 0 as the number of iterations increased approached 0.6. What is special about this value?

It turns out that in the stationary distribution (frequently denoted $\boldsymbol{\pi}$), the probability of sampling a chain in state 0 is 0.6. We would write this as $\boldsymbol{\pi} = \{0.6, 0.4\}$ or the pair of statements: $\boldsymbol{\pi}_0 = 0.6, \boldsymbol{\pi}_1 = 0.4$.

How can we show that this is the steady-state (equilibrium) distribution? The flux "out" of each state must be equal to the "flux" into that state for a system to be at equilibrium. Here it is helpful to think about running infinitely many chains. If we have an equilibrium probability distribution, then we can characterize the probability of a chain leaving state 0 in at one particular step in the process. This is the joint probability of being in state 0 at step i and the probability of an $0 \rightarrow 1$ transition at point i. Thus,

$$\mathbb{P}(0 \rightarrow 1 \text{ at } i) = \mathbb{P}(x_i = 0)\mathbb{P}(x_{i+1} = 1|x_i = 0)$$

At the equilibrium $\mathbb{P}(x_i = 0) = \boldsymbol{\pi}_0$, so the flux out of 0 at step i is $\boldsymbol{\pi}_0\mathbb{P}(x_{i+1} = 1|x_i = 0)$. In our simple (two-state) system the flux into state 0 can only come from state 1, so:

$$\begin{aligned}\mathbb{P}(1 \rightarrow 0 \text{ at } i) &= \mathbb{P}(x_i = 1)\mathbb{P}(x_{i+1} = 0|x_i = 1) \\ &= \boldsymbol{\pi}_1\mathbb{P}(x_{i+1} = 0|x_i = 1)\end{aligned}$$

If we force the "flux out" and "flux in" to be identical:

$$\begin{aligned}\boldsymbol{\pi}_0\mathbb{P}(x_{i+1} = 1|x_i = 0) &= \boldsymbol{\pi}_1\mathbb{P}(x_{i+1} = 0|x_i = 1) \\ \frac{\boldsymbol{\pi}_0}{\boldsymbol{\pi}_1} &= \frac{\mathbb{P}(x_{i+1} = 0|x_i = 1)}{\mathbb{P}(x_{i+1} = 1|x_i = 0)}\end{aligned}$$

[2] The fancy jargon is "irreducible." A reducible chain has some states that are disconnected from others, so the chain can be broken up (reduced) into two or more chains that operate on a subset of the states.

We can then solve for the actual frequencies by recognizing that $\boldsymbol{\pi}$ is a probability distribution:

$$\sum_k \boldsymbol{\pi}_k = 1$$

For the system that we are considering:

$$\begin{aligned}
\frac{\boldsymbol{\pi}_0}{\boldsymbol{\pi}_1} &= \frac{0.9}{0.6} = 1.5 \\
\boldsymbol{\pi}_0 &= 1.5\boldsymbol{\pi}_1 \\
1.5\boldsymbol{\pi}_1 + \boldsymbol{\pi}_1 &= 1.0 \\
\boldsymbol{\pi}_1 &= 0.4 \\
\boldsymbol{\pi}_0 &= 0.6
\end{aligned}$$

Thus, the transition probabilities determine the stationary distribution.

If we can choose the transition probabilities then we can construct a sampler that will converge to any distribution that we would like.

When we have a state set $\mathcal{S}$ with more than 2 states, we can express a general statement about the stationary distribution:

$$\boldsymbol{\pi}_j = \sum_{k\in\mathcal{S}} \boldsymbol{\pi}_k \mathbb{P}(x_{i+1} = j | x_i = k) \tag{1}$$

The connection to the "flux out" and "flux in" argument may not seem obvious, but note that the "flux out" can be quantified in terms of the 1 - the "self-transition." Let $\mathcal{S}_{\neq j}$ be the set of states that excludes state j, so the "flux out" is:

$$\begin{aligned}
\text{flux out of } j &= \boldsymbol{\pi}_j \mathbb{P}(x_{i+1} \in \mathcal{S}_{\neq j} | x_i = j) \\
&= \boldsymbol{\pi}_j \left[1 - \mathbb{P}(x_{i+1} \in j | x_i = j)\right]
\end{aligned}$$

The flux in is:

$$\text{flux into } j = \sum_{k\in\mathcal{S}_{\neq j}} \pi_k \mathbb{P}(x_{i+1} = j | x_i = k)$$

Forcing the fluxes to be equal gives us:

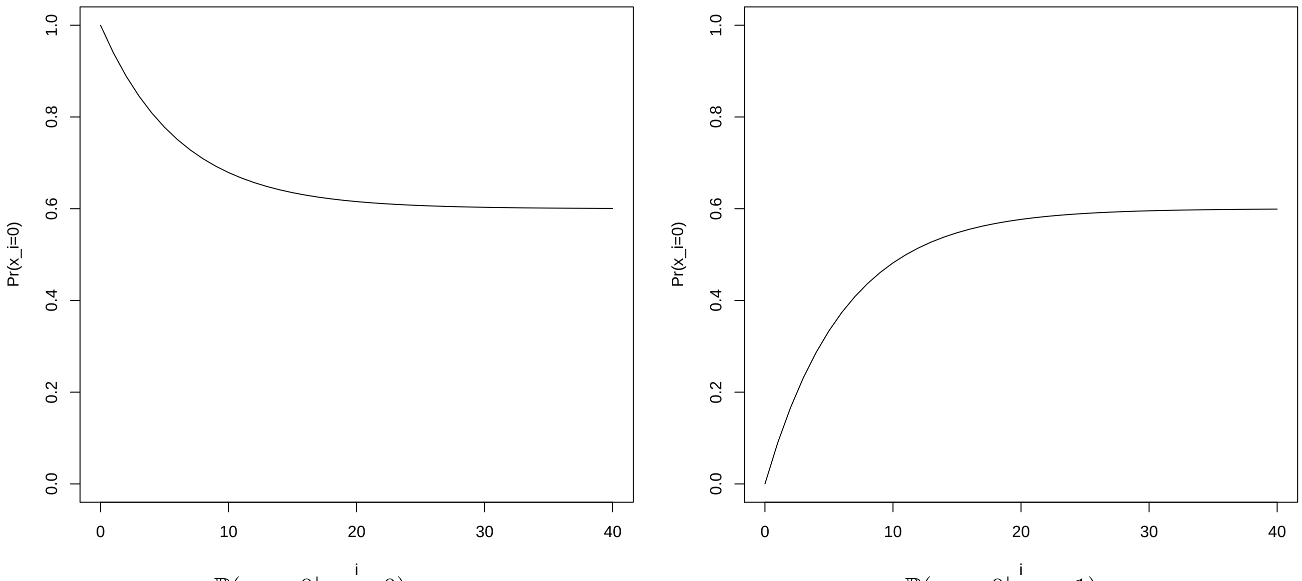
$$\begin{aligned}
\boldsymbol{\pi}_j \left[1 - \mathbb{P}(x_{i+1} = j | x_i = j)\right] &= \sum_{k\in\mathcal{S}_{\neq j}} \boldsymbol{\pi}_k \mathbb{P}(x_{i+1} = j | x_i = k) \\
\boldsymbol{\pi}_j &= \boldsymbol{\pi}_j \mathbb{P}(x_{i+1} = j | x_i = j) + \sum_{k\in\mathcal{S}_{\neq j}} \boldsymbol{\pi}_k \mathbb{P}(x_{i+1} = j | x_i = k) \qquad (2) \\
&= \sum_{k\in\mathcal{S}} \boldsymbol{\pi}_k \mathbb{P}(x_{i+1} = j | x_i = k)
\end{aligned}$$

Mixing

We just saw how changing transition probabilities will affect the stationary distribution. Perhaps, there is a one-to-one mapping between transition probabilities and a stationary distribution. In the form of a question: if we were given the stationary distribution could we decide what the transition rates must be to achieve that distribution?

It turns out that the answer is "No." Which can be seen if we examine equation (2). Imagine dropping the "flux out" of each state by a factor of 10. Because the "self-transition" rate would increase by the appropriate amount, we can still end up in the same stationary distribution.

How would such a chain behave? The primary difference is that it would "mix" more slowly. Adjacent steps would be more likely to be in the same state, and it would take a larger number of iterations before the chain "forgets" its starting state. Figure (3) depicts the same probability statements as Figure (2) but for a process in which $\mathbb{P}(x_{i+1} = 1|x_i = 0) = 0.06$ and $\mathbb{P}(x_{i+1} = 0|x_i = 1) = 0.09$.



$\mathbb{P}(x_i = 0|x_0 = 0)$

$\mathbb{P}(x_i = 0|x_0 = 1)$

Figure 3: The probability of being in state 0 as a function of the step number, i, for two different starting states (0 and 1) for the Markov process depicted in Figure (1) but with the state-changing transition probabilities scaled down by a factor of 10. Compare this to Figure (2).

Thus, that the rate of convergence of a chain to its stationary distribution is an aspect of a Markov chain that is separate from what the stationary distribution is.

In MCMC we will design a Markov chain such that its stationary distribution will be identical to the posterior probability distribution over the space of parameters. We will try to design chains that have high transition probabilities, so that our MCMC approximation will quickly converge to the posterior. But even a slowly mixing chain will (in theory) eventually be capable of providing an approximation to the posterior probability.

Detailed Balance

In principle, we could use the information in Equation (1) to inform us as to how to construct a chain with a desired $\boldsymbol{\pi}$. In practice, this is difficult when we have a large number of states. In the continuous parameter case, we have an infinite number of states and the summation becomes an integral. Setting transition probabilities in a general way such that the total flux into and out of any state satisfies the property shown in Equation (1) is tricky.

Typically, we restrict ourselves to a subset of possible Markov chains: those that satisfy detailed balance for all pairs of states j and k:

$$\boldsymbol{\pi}_j \mathbb{P}(x_{i+1} = k | x_i = j) = \boldsymbol{\pi}_k \mathbb{P}(x_{i+1} = j | x_i = k) \tag{3}$$

Detailed balance means that for any two possible states, the flux between them balances out.

Clearly, if the flux out of j into k is exactly matched by the flux from k into j, then Equation (1) will also be satisfied (the total flux out of j will be balanced by the total flux into j).

Another way to express the detailed balance restriction is to put the transition probabilities on one side of the equation and the stationary frequencies on the other:

$$\frac{\boldsymbol{\pi}_j}{\boldsymbol{\pi}_k} = \frac{\mathbb{P}(x_{i+1} = j | x_i = k)}{\mathbb{P}(x_{i+1} = k | x_i = j)} \tag{4}$$

MCMC

In the Metropolis-Hastings algorithm, we choose rules for constructing a stochastic walk through parameter space. We adopt transition probabilities such that the stationary distribution of our Markov chain is equal to posterior probability distribution. In notation we would like:

$$\boldsymbol{\pi}_{\theta_j} = \mathbb{P}(\theta_j | \text{Data})$$

Let's work through an example for which we can calculated the posterior probabilities analytically, and then construct a MCMC algorithm that will converge to those posterior probabilities.

Double-headed coin

Imagine a store that sells four-packs of quarters, but in some of the packs some of the coins have George Washington's head on both sides ($\mathbb{P}(H) = 1$ for these coins) while the remaining coins are fair coins ($\mathbb{P}(H) = \mathbb{P}(T) = 0.5$).

Further more there are equal numbers of packages that contain 0, 1, 2, 3, or 4 double-headed coins, and the packages are not labelled.

Someone grabs a package at random and records the number of heads from two experiments in which all for coins are flipped. In the first experiment all four flips are heads. In the second experiment three of the four flips are heads.

We want to make a probability statement about the number of coins in the pack that are double-headed. Thus our data is $D = \{d_1 = 4, d_2 = 3\}$. We have one parameter, θ, which can take on 5 values: $\theta \in \{0, 1, 2, 3, 4\}$.

If the five times of packs are equally likely, then we can express the prior probability of each value of θ as 1/5.

We can calculate the probability of different number of heads, h, given each possible value of θ by calculating the probability of $d - \theta$ heads in $4 - \theta$ flips. We use these likelihood calculations and Bayes formula:

$$\mathbb{P}(\theta|D) = \frac{\mathbb{P}(D|\theta)\mathbb{P}(\theta)}{\mathbb{P}(D)} = \frac{\mathbb{P}(D,\theta)}{\mathbb{P}(D)}$$

to produce the following table:

	θ					
	0	1	2	3	4	Sum
$\mathbb{P}(d = 4\|\theta)$	1/16	2/16	4/16	8/16	16/16	
$\mathbb{P}(d = 3\|\theta)$	4/16	6/16	8/16	8/16	0	
$\mathbb{P}(d = 2\|\theta)$	6/16	6/16	4/16	0	0	
$\mathbb{P}(d = 1\|\theta)$	4/16	2/16	0	0	0	
$\mathbb{P}(d = 0\|\theta)$	1/16	0	0	0	0	
$\sum_{i=0}^{4} \mathbb{P}(d = i\|\theta)$	1	1	1	1	1	
$\mathbb{P}(d_1 = 4\|\theta)$	1/16	2/16	4/16	8/16	16/16	
$\mathbb{P}(d_2 = 3\|\theta)$	4/16	6/16	8/16	8/16	0	
$\mathbb{P}(D\|\theta)$	4/256	12/256	32/26	64/256	0	
$\mathbb{P}(\theta)$	0.2	0.2	0.2	0.2	0.2	1
$\mathbb{P}(\theta, D)$	4/1280	12/1280	32/1280	64/1280	0	
$\mathbb{P}(D) = \sum_\theta \mathbb{P}(\theta, D)$						112/1280
$\mathbb{P}(\theta\|D) = \frac{\mathbb{P}(D,\theta)}{\mathbb{P}(D)}$	1/28	3/28	2/7	4/7	0	1

Table 1: Probability calculation for four coins, and data consisting of $d_1 = 4, d_2 = 3$

MCMC

In this case, we can simply calculate the posterior probability of θ given the data. But, in more complicated scenarios we cannot perform calculations for all possible values of θ. When we cannot try all possible values of θ, then we can't sum them to get $\mathbb{P}(D)$. In fact, our inability to calculate

$\mathbb{P}(D)$, the marginal probability of the data, is the primary reason that we usually have to resort to MCMC to perform Bayesian inference.

Recall, that in MCMC we are going to explore the space of parameters. In the case of the coin-example, there are 5 possible values for θ. So we need to construct a Markov chain with 5 states. Figure (4) shows one example. Note that not all of the states are adjacent, in the sense that there is not an arc between all possible pairs. Thus, it would not be possible to move from state 0 to state 2 in a single step. But that is fine - a Markov chain with this architecture could be constructed to sample the posterior probability distribution. The $m_{j,k}$ labels on the arcs mean "the probability of moving from j to k."

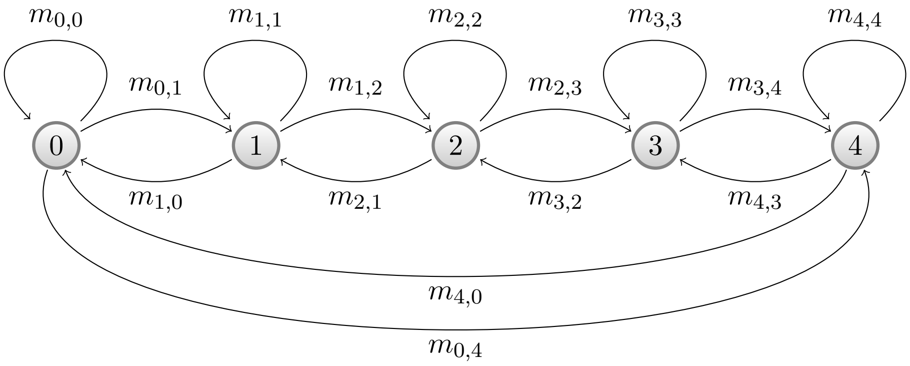


Figure 4: A graphical depiction of a five-state Markov process.

How can we construct a chain with the desired stationary distribution? We have to choose transition probabilities that obey detailed balance (equation 4). The other constraint is that the sum of the probabilities over all states at $i+1$ (conditional on the state at i) must be 1.

Specifically, we would like:

$$\begin{aligned}\boldsymbol{\pi}_0 &= \mathbb{P}(\theta=0|D) = \frac{\mathbb{P}(D|\theta=0)\mathbb{P}(\theta=0)}{\mathbb{P}(D)}\\ \boldsymbol{\pi}_1 &= \mathbb{P}(\theta=1|D) = \frac{\mathbb{P}(D|\theta=1)\mathbb{P}(\theta=1)}{\mathbb{P}(D)}\\ \ldots\boldsymbol{\pi}_4 &= \mathbb{P}(\theta=4|D) = \frac{\mathbb{P}(D|\theta=4)\mathbb{P}(\theta=4)}{\mathbb{P}(D)}\end{aligned}$$

Note that in our compact notation on the graph, $m_{0,1} = \mathbb{P}(x_{i+1}=1|x_i=0)$, thus we can rephrase equation (4) as:

$$\frac{\boldsymbol{\pi}_j}{\boldsymbol{\pi}_k} = \frac{m_{k,j}}{m_{j,k}}$$

So if we want to set $m_{0,1}$ and $m_{1,0}$ so that the transitions between 0 and 1 obey detailed balance and the stationary distribution is the posterior distribution we have:

$$\begin{aligned}\frac{m_{1,0}}{m_{0,1}} &= \frac{\boldsymbol{\pi}_0}{\boldsymbol{\pi}_1}\\ &= \left(\frac{\mathbb{P}(D|\theta=0)\mathbb{P}(\theta=0)}{\mathbb{P}(D)}\right)\Big/\left(\frac{\mathbb{P}(D|\theta=1)\mathbb{P}(\theta=1)}{\mathbb{P}(D)}\right)\end{aligned}$$

$$= \frac{\mathbb{P}(D|\theta = 0)\mathbb{P}(\theta = 0)}{\mathbb{P}(D|\theta = 1)\mathbb{P}(\theta = 1)}$$

A mathematically boring, but important, thing happened in the last step: the $\mathbb{P}(D)$ crossed out. Thus in order to express the ratio of move probabilities we don't need to be able to calculate $\mathbb{P}(D)$.

This is crucial – in fact this is the reason we use MCMC. Recall that we cannot calculate $\mathbb{P}(D)$ for most "real-world" problems because it involves summing over states. Fortunately, we don't have to calculate it to do MCMC.

In our coin example, we can see that $\mathbb{P}(D|\theta = 1)\mathbb{P}(\theta = 1)$ is three times higher than $\mathbb{P}(D|\theta = 0)\mathbb{P}(\theta = 0)$. Thus we need to ensure that the probability of moving from $0 \rightarrow 1$ is 1/3 the probability of moving from $1 \rightarrow 0$. For instance, we could set $m_{0,1} = 1$ and $m_{1,0} = 1/3$. Or we could choose $m_{0,1} = 1/2$ and $m_{1,0} = 1/6$. Which should we use? We'd like an MCMC simulation that converges quickly so we should set the transition probabilities as high as possible. So, using $m_{0,1} = 1$ and $m_{1,0} = 1/3$ sounds best.

Similar arguments lead us to the conclusion that $m_{1,2} = 1$ and $m_{2,1} = 3/8$. However this reveals a problem: setting these four transition probabilities in this way means that $\mathbb{P}(x_{i+1} = 0|x_i = 1) = 1/3$ and $\mathbb{P}(x_{i+1} = 2|x_i = 1) = 1$. But this violates fundamental rules of probability - the probability of all mutually-exclusive events must sum to 1.0. With our transition probabilities it exceeds 1.

On solution is to view a transition probability as a joint event: the move is proposed and the move is accepted. We can denote the probability of proposing a move from j to k as $q(j, k)$. We can denote the acceptance probability of a move from j to k (given that the move was proposed) as $\alpha(j, k)$. If we were to use x'_{i+1} to denote the state proposed at step $i + 1$ then

$$\begin{aligned} q(j,k) &= \mathbb{P}(x'_{i+1} = k|x_i = j) \\ \alpha(j,k) &= \mathbb{P}(x_{i+1} = k|x'_i = j, x'_{i+1} = k) \end{aligned}$$

This means that we can choose proposal probabilities that sum to one for all of the state-changing transitions. Then, we can multiply them by the appropriate acceptance probability (keeping that as high as we can, but ≤ 1). So, we get

$$\frac{m_{1,0}}{m_{0,1}} = \frac{q(1,0)\alpha(1,0)}{q(0,1)\alpha(0,1)} = \frac{\mathbb{P}(D|\theta = 0)\mathbb{P}(\theta = 0)}{\mathbb{P}(D|\theta = 1)\mathbb{P}(\theta = 1)} \tag{5}$$

We have a great deal of flexibility in selecting how we perform proposals on new states in MCMC. We have to assure that $q(j, k) > 0$ whenever $q(k, j) > 0$; failing to do this means that the ratio of proposal probabilities will be 0 in one direction and infinite in the other. It is fine for $q(j, k) = q(k, j) = 0$ for some pairs of states (we already said that it was OK if not all states were adjacent in the Markov chain – note that two states that are not adjacent still fulfill detailed balance because both fluxes are 0).

Once we have chosen a proposal scheme, though, we do not have as much flexibility when choosing whether or not to accept a proposal. Rearranging terms in equation (5) to put the acceptance probabilities on one side gives us:

$$\begin{aligned}\frac{\alpha(1,0)}{\alpha(0,1)} &= \frac{\mathbb{P}(D|\theta=0)\mathbb{P}(\theta=0)q(0,1)}{\mathbb{P}(D|\theta=1)\mathbb{P}(\theta=1)q(1,0)}\\ &= \left(\frac{\mathbb{P}(D|\theta=0)}{\mathbb{P}(D|\theta=1)}\right)\left(\frac{\mathbb{P}(\theta=0)}{\mathbb{P}(\theta=1)}\right)\left(\frac{q(0,1)}{q(1,0)}\right)\end{aligned}$$

or, in words,

$$\begin{array}{c}\text{acceptance}\\ \text{ratio}\end{array} = \left(\begin{array}{c}\text{likelihood}\\ \text{ratio}\end{array}\right)\left(\begin{array}{c}\text{prior}\\ \text{ratio}\end{array}\right)\left(\begin{array}{c}\text{Hastings}\\ \text{ratio}\end{array}\right) \tag{6}$$

here the name "Hastings ratio" comes from Hastings (1970). We can generalize this to any pair of adjacent states, j and k, by substituting in j for 0 and k for 1:

$$\frac{\alpha(k,j)}{\alpha(j,k)} = \left(\frac{\mathbb{P}(D|\theta=j)}{\mathbb{P}(D|\theta=k)}\right)\left(\frac{\mathbb{P}(\theta=j)}{\mathbb{P}(\theta=k)}\right)\left(\frac{q(j,k)}{q(k,j)}\right) \tag{7}$$

It also applies in the case of continuous parameters (with few restrictions), however then we have ratios of probability densities rather than probabilities. The MCMC algorithm that exploits Equation (7) is referred to as the Metropolis-Hastings algorithm (Metropolis et al., 1953).

Metropolis-Hastings in a program

Equation (7) express the fundamental constraint that we must obey when constructing an MCMC sampler. The last step toward being able to write a piece of software to perform this algorithm is to think about how this constraint can be implemented. As we simulate a Markov chain, we start from a valid state. Thus, at each step, i, we have a current state, x_i. In an iteration:

1. We propose a new state, x'_{i+1} using code that proposes move according to the probability statements encapsulated in $q(x_i, j)$.
2. We calculate an acceptance probability, $\alpha(x_i, x'_{i+1})$, for the move we proposed.
3. If we draw a uniform random number less than this acceptance probability, then we set $x_{i+1} = x'_{i+1}$, otherwise we set $x_{i+1} = x_i$.
4. We add one to the step count $i \leftarrow i + 1$

Note that when we reject we stay in the same state. Thus for every "destination" state, k, from starting state, j, we add $q(j,k)\,(1-\alpha(j,k))$ to the $j \rightarrow j$ transition probability. We don't have to worry about enforcing detailed balance for the $j \rightarrow j$ transition (because the "flux out" has to be the same as the "flux in" for any probability we assign).

Note that we have to have an acceptance probability, while equation (7) gives us the ratio of probabilities between states. The insight that we use to proceed is that we want to maximize the probability that the Markov chain move through state space. We can calculate the acceptance ratio for any pair of states because we can calculate the likelihood, prior probability (or prior probability density), and proposal probability (or proposal probability density) at any state.

To make things slightly more concrete, let's use A to represent the acceptance ratio. Imagine that $\alpha(x_i, x'_{i+1})/\alpha(x_i, x'_{i+1}) = A \geq 1$. This means that $\alpha(x_i, x'_{i+1}) > \alpha(x'_{i+1}, x_i)$. There will be an infinite number of choices of acceptance probabilities that satisfy this, but we want to make the probabilities as big as possible. The maximum value for a probability is 1, so we can simply set:

$$\begin{aligned}\alpha(x_i, x'_{i+1}) &= 1\\ \alpha(x'_{i+1}, x_i) &= \frac{1}{A}\end{aligned}$$

In the alternative scenario, $\alpha(x_i, x'_{i+1})/\alpha(x_i, x'_{i+1}) = A \leq 1$. We can satisfy this an maximize changing state by

$$\begin{aligned}\alpha(x'_{i+1}, x_i) &= 1\\ \alpha(x_i, x'_{i+1}) &= A\end{aligned}$$

As we simulate a Markov chain, we can figure what condition we are in by imagine if we had proposed the reverse move ($x'_{i+1} \rightarrow x_i$ instead of $x_i \rightarrow x'_{i+1}$). We can calculate the acceptance ratio for the pair of moves (forward and reverse) using equation (7). If the forward probability must be higher, then we set the acceptance probability to 1. If the forward probability must be the lower probability, then we set the acceptance probability to be equal to the acceptance ratio.

This leads to the following rule:

$$\alpha(x_i, x'_{i+1}) = \min\left[1, \left(\frac{\mathbb{P}(D|\theta = x'_{i+1})}{\mathbb{P}(D|\theta = x_i)}\right)\left(\frac{\mathbb{P}(\theta = x'_{i+1})}{\mathbb{P}(\theta = x_i)}\right)\left(\frac{q(x'_{i+1}, x_i)}{q(x_i, x'_{i+1})}\right)\right]$$

Double-headed coin MCMC

Returning to the example of estimating the number of double-headed coins. We can use a proposal distribution in which we propose either of the adjacent states in the five-state Markov chain over the number of coins that are double-headed with probability 0.5. This means that when we $q(j, k) = q(k, j) = 0.5$ for all adjacent states. Thus the Hastings ratio is 1 for all possible moves.

We can use the likelihoods, priors (from the table 1) along with this Hastings ratio to calculate acceptance probabilities. Multiplying acceptance probabilities by the proposal probabilities yields the Markov chain shown in Figure (5). Note that we do not generally calculate transition probabilities when doing MCMC - we resort to MCMC when we cannot visit all of the possible states.

Figure 6 shows a trace plot of the sampled values for θ.

In this example some of the transitions probabilities are 0. This is not a problem here because these probabilities are all associated with the state $\theta = 4$ which has posterior probability of 0 (because it conflicts with the data, resulting in a likelihood of 0). If state $\theta = 2$ and $\theta = 4$ had *both* had posterior probabilities of 0, then we would not have constructed a valid (irreducible) Markov chain for the purposes of MCMC. There would have been no way to get from state 1 to state 3, because all of the paths would be "blocked" by a transition with probability 0. This is rarely a problem, because

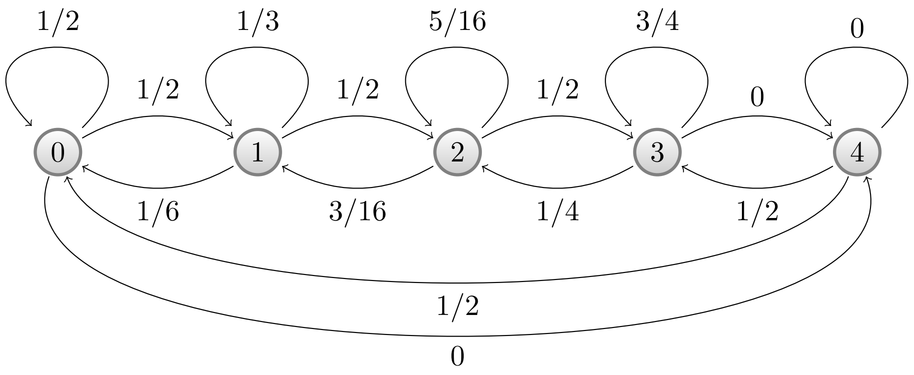


Figure 5: A graphical depiction of a five-state Markov process with transition probabilities calculated for a equiprobable proposal distribution and the data-based calculations from table 1.

most parameters combinations in "real-world" problems cannot result in posterior probabilities of 0. If you are working with a problem in which some of the posteriors can be 0, then you have to assure that the states with non-zero posterior probability are connected in your Markov chain.

If the posteriors of some states are very low, then you can end up with very small transition probabilities. Technically, if the transition probabilities are greater than zero, then the chain is irreducible and eventually the MCMC simulation will converge to the stationary distribution. In practice, very low transition probabilities between different "islands" of states represents a very demanding inference problem. Mixing will almost certainly be very slow, and your inference from MCMC can be misleading because any finite sampling of the Markov chain can miss very appreciable parts of parameter space. Running independent simulations from different starting points is one way to diagnose a Markov chain that is mixing slowly.

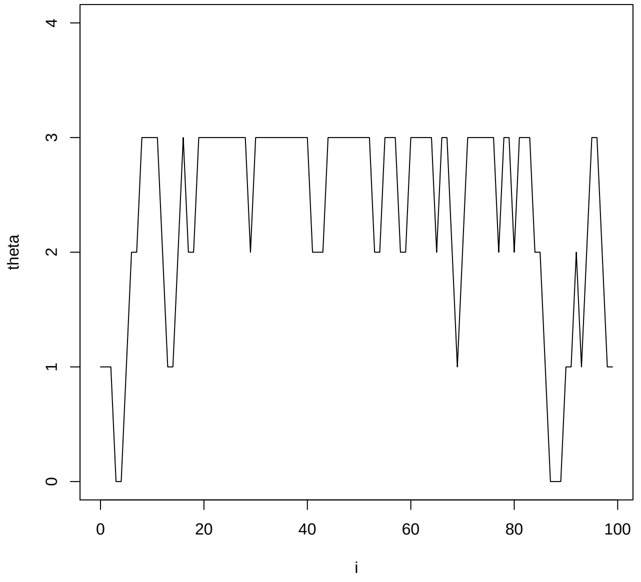


Figure 6: Trace of θ values for the first 100 iterations of a MCMC run

A Code to calculate probabilities for two-state Markov chain

Python code, `two_state.py`, to generate probabilities for a certain number of iterations:

```python
#!/usr/bin/env python
import sys
from random import random as uniform

a_to_b = float(sys.argv[1])
assert(a_to_b > 0.0 and a_to_b <= 1.0)
b_to_a = float(sys.argv[2])
assert(b_to_a > 0.0 and b_to_a <= 1.0)
ti_prob_list = [a_to_b, b_to_a]
b_to_b = 1 - b_to_a
state = int(sys.argv[4])
assert(state in [0,1])
if state == 0:
    prob_list = [1.0, 0.0]
else:
    prob_list = [0.0, 1.0]
num_it = int(sys.argv[3])
assert(num_it > 0)

print "Gen\tPrZero\tPrOne"
for i in xrange(num_it):
    print "\t".join([str(i)] + [str(p) for p in prob_list])
    pb = prob_list[0]*a_to_b + prob_list[1]*b_to_b
    prob_list = [1-pb, pb]
```

R code, `plotProbTrace.R`, to create plots from a file with columns labelled “Gen” and “PrZero”:

```r
fn = commandArgs(TRUE)
d = read.table(fn, header=TRUE, sep="\t");
pdf(paste(fn, '.pdf', sep=""));
plot(d$Gen, d$PrZero, xlab="i", ylab="Pr(x_i=0)", type="l", ylim=c(0,1));
dev.off();
```

bash invocation using the scripts to create pdf documents:

```bash
$ python two_state.py 0.6 0.9 20 0 >start_from_0.txt
$ R --no-save -f plotProbTrace.R --args start_from_0.txt
$ python two_state.py 0.6 0.9 20 1 >start_from_1.txt
$ R --no-save -f plotProbTrace.R --args start_from_1.txt
```

B Snippet of code for double-headed coin MCMC

```
likelihood = calc_likelihood(state)
assert(likelihood > 0.0)
counts = [0]*(num_coins + 1)
sys.stderr.write(”Gen\tlike\ttheta\n”)
for i in xrange(num_it):
    sys.stderr.write(”%d\t%f\t%d\n” % (i, likelihood, state))
    counts[state] += 1
    prev_likelihood = likelihood
    if random.random() < 0.5:
        proposed = state + 1
        if state > num_coins:
            proposed = 0
    else:
        proposed = state - 1
        if state < 0:
            proposed = num_coins
        # Prior ratio is 1.0, se we can ignore it

        # Hastings ratio is 1.0, se we can ignore it

    likelihood = calc_likelihood(proposed)
    if likelihood > prev_likelihood:
        state = proposed
    else:
        if random.random() < likelihood/prev_likelihood:
            state = proposed
        else:
            likelihood = prev_likelihood

print ”Posterior probabilities:”
for state in range(num_coins + 1):
    print state, float(counts[state])/num_it
```

Full program at http://phylo.bio.ku.edu/slides/coin_contamination.py.txt

References

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *J. Chem. Phys.*, 21(6):1087–1092.