http://www.math.nyu.edu/faculty/goodman/teaching/MonteCarlo15/

Class notes: Monte Carlo methods
Week 3, Markov chain Monte Carlo

Jonathan Goodman

March 3, 2015

1 Introduction

Markov chain Monte Carlo, or *MCMC*, is a way to sample probability distributions that cannot be sampled practically using direct samplers. Most complex probability distributions in more than a few variables are are sampled in this way. For us, a stationary Markov chain is a random sequence $X_1, X_2, \ldots$, where $X_{k+1} = M(X_k, \xi_k)$, where $M(x,\xi)$ is a fixed function and the inputs ξ are i.i.d. random variables. Mathematically, a stationary Markov chain is defined by a *transition distribution*, R, that describes distribution of X_{k+1} conditional on X_k. We use different related notations for this, including $R(y|x)$ for the probability density of $Y = M(X,\xi)$, or $R_{xy} = \mathrm{P}(x \to y)$ for the probability that $X_{k+1} = y$ given that $X_k = x$. An *MCMC sampler* is a code that implements M. It is a direct sampler for the conditional distribution $R(\cdot|X_k)$. A direct sampler is `X = fSamp()`, while the MCMC sampler is `X = fSamp(X)`. Both the direct and MCMC samplers can call `uSamp()` many times. The Markov chain is designed so that $f_k \to f$ as $k \to \infty$, where f_k is the distribution of X_k and f is the target distribution. It turns out to be possible to create suitable practical Markov chains for many distributions that do not have practical direct samplers.

Two theorems underly the application of MCMC, the *Perron Frobenius* theorem, and the *ergodic theorem for Markov chains*. A Markov chain *preserves* f if f is an *invariant distribution* for R. This means that if $X_k \sim f$, then $X_{k+1} \sim f$. Perron Frobenius says, among other things, a Markov chain preserves f, and if it is non-degenerate (aperiodic and irreducible, see below), then $f_k \to f$ as $k \to \infty$. It is usually easy to check the non-degeneracy conditions. The importance of the Perron Frobenius theorem is that we do not have to design MCMC samplers that make f_k converge to f, we only have to make them preserve f.

The MCMC samples X_k are not independent, but they suffice for estimating expected values. The *ergodic theorem* for Markov chains says that if R preserves f and is non-degenerate then

$$\widehat{A}_n = \frac{1}{n}\sum_{k=1}^{n} V(X_k) \;\to\; A = \mathrm{E}_f[V(X)] \quad \text{as } n \to \infty \; . \tag{1}$$

For this, it helps that $f_k \to f$ as $k \to \infty$. But this is not enough. It is also necessary that X_k and X_{k+1} become independent as $k \to \infty$ and $t \to \infty$. The statistical error $\mathrm{var}(\widehat{A}_n) \approx \mathrm{E}\Big[(\widehat{A}_n - A)^2\Big]$ depends on the rate of convergence $\mathrm{cov}(V(X_k), V(X_{k+t}) \to 0$ as $t \to \infty$. The MCMC does not produce any independent samples of f, but it produces enough samples with distribution close enough to f that are close enough to being independent.

There is no theorem that says that a given correct sampler is practical. On the contrary, there are distributions for which no practical sampler is known. When you use MCMC, even more than with a direct sampler, it is important to do convergence checks and estimate realistic error bars. The variance of $\widehat{A}_n$ with a direct sampler is $\frac{1}{n}\sigma^2$. With an MCMC sampler, the variance can be much larger.

This class covers generic MCMC methods. After today you will be able to create a correct sampler for more or less any distribution. We do this first for finite state space and discrete probability, then for continuous state space and continuous probability. The basic ideas in both cases are:

- *Detailed balance* achieved through rejection, called *Metropolis* or *Metropolis hastings.*
- *Partial resampling*, originally called the *heat bath* method, since renamed the *Gibbs sampler.*
- Compositions of the above that make the overall chain mixing.

2 MCMC error bars

Error bars for MCMC samples are harder to make than error bars for independent samples. There are two relaxation times that are relevant to knowing whether you have taken enough MCMC steps. The *burn-in time* is the number of steps it takes to achieve the convergence $f_k \to f$. This is not precisely defined because f_k is never equal to f. But it often takes many MCMC steps before X_k resembles typical samples from f. MCMC practitioners often throw away the first part of an MCMC sequence, hoping to avoid biased results that come from the burn-in steps. But it is hard to make a statistical estimate of the burn-in time, given that there is only one burn-in in a given MCMC run.

The *auto-correlation* time, τ, can be estimated from the MCMC run. Normally, the run length, n, should many times τ. The auto-correlation time is a way to understand the approximate formula for the variance of the estimator (1), which is

$$\mathrm{var}\left(\widehat{A}_n\right) \approx \frac{D}{n} \; , \tag{2}$$

where

$$D = \sum_{t=-\infty}^{\infty} C(t) = C(0) + 2\sum_{t=1}^{\infty} C(t) \; , \tag{3}$$

and

$$C(t) = \lim_{k\to\infty} \mathrm{cov}(V(X_k), V(X_{k+t})) \; . \tag{4}$$

We explain these in reverse order, starting with the *auto-covariance* function $C(t)$. As $k \to \infty$, the distribution of X_k converges to f. In the limit $k \to \infty$, and if $t > 0$, the joint distribution of (X,Y)=(X_k, X_{k+t}) converges to $f(x)R^t(y|x)$, where R^t is the transition matrix to the power t. The definition (4) shows that

$C(-t) = C(t)$. This justifies the second equality in the definition of D. The formula (3) is called the *Kubo* formula. Kubo derived it in the form

$$\mathrm{var}\left(\sum_{k=1}^{n} V(X_k)\right) \approx nD \; .$$

The left side is the sum of n random steps of size $V(X_k)$. The identity, which is a generalization of a formula of Einstein, suggests that D is the effective diffusion coefficient for a random walk process with correlated steps. That is because nD would be the mean square displacement after n independent steps, each having variance D.

The formula (3) may be written in a way that makes the relationship between independent sampling and MCMC sampling more clear. The time zero covariance, $C(0)$, is the *static variance*, which would be called σ^2 for independent sampling. We factor out the static variance and write the Kubo formula as

$$D = C(0)\left[1 + 2\sum_{t=1}^{\infty} \rho(t)\right] \; . \tag{5}$$

Here,

$$\rho(t) = \frac{C(t)}{C(0)} \; , \tag{6}$$

is the auto-correlation function. This is the correlation of $V(X_k)$ with itself at time lag t. In general, the correlation coefficient between random variables Y and Z is

$$\rho_{YZ} = \frac{\mathrm{cov}(Y, Z)}{\sqrt{\mathrm{var}(Y)\,\mathrm{var}(Z)}} \; .$$

This is a dimensionless measure of the relation between Y and Z. It ranges from -1 to 1, with extreme values only if $Y = CZ$ for some constant C. In the context here, $Y = V(X_k)$ and $Z = V(X_{k+t})$. For large k, the variances of $V(X_k)$ and $V(X_{k+t})$ are the same, hence the formula (6) for $\rho(t)$.

The quantity in square braces in the Kubo formula (5) is the auto-correlation time:

$$\tau = 1 + 2\sum_{t=1}^{\infty} \rho(t) \; . \tag{7}$$

It measures the number of MCMC steps needed to create an effectively independent sample of f, for the purpose of estimating $\mathrm{E}_f[V(X)]$. You can see this by rewriting the estimator variance formula (2) in terms of τ:

$$\mathrm{var}\left(\widehat{A}_n\right) \approx \frac{C(0)}{n/\tau} = \frac{\sigma^2}{n_{\mathit{eff}}} \; .$$

This formula defines the *effective number of samples*, which is the number of MCMC steps measured in units of the auto-correlation time. The variance of the estimator is the (static) variance of $V(X)$ divided by the effective number of samples.

<preach>In Monte Carlo practice, it is crucial to report n_{eff} rather than just n. For example, if you take $n = 10000$ MCMC steps and the auto-correlation time is $\tau = 1000$, then A is estimated to the accuracy of just $n_{\mathit{eff}} = 10$ independent steps. Unfortunately, $\tau = 1000$ may be the best we can do even with sophisticated MCMC algorithms. Taking a large number of MCMC steps does not necessarily mean small error bars. Understanding the correlations in $V(X_k)$ is crucial, if you want to know whether you have the right answer.</preach>

The auto-correlation time, τ, depends both on the MCMC algorithm and on the quantity, $V(X)$, being measured. The *spectral gap* of the MCMC algorithm may give a bound on τ for any V, if the chain satisfies *detailed balance*. A small spectral gap implies that there are observables $V(x)$ with large τ. It is possible that the Markov chain you are running has a small spectral gap but your V is not among the bad ones that have a correspondingly large τ. This seems to be uncommon in practice.

If we use the Kubo formula to estimate error bars (strongly recommended), we must estimate τ from Monte Carlo data. There does not seem to be a straightforward robust way to do this. It is natural to start by estimating the auto-covariance and auto-correlation functions, first

$$\widehat{C}(t) = \frac{1}{n-t-1} \sum_{k=1}^{n-t} \left(V(X_k) - \widehat{A}_n \right) \left(V(X_{k+t}) - \widehat{A}_n \right) \; ,$$

then

$$\widehat{\rho}(t) = \frac{\widehat{C}(t)}{\widehat{C}(0)} \; .$$

A typical $\widehat{\rho}$ goes from 1 to zero as t goes from 0 to several times τ. The exact $\rho(t)$ converges to zero as $t \to \infty$, but the estimated $\widehat{\rho}(t)$ fluctuates around zero in a noisy way. The following estimator is a natural implementation of auto-correlation sum (7):

$$\widehat{\tau}_{inc,n} = 1 + 2 \sum_{t=1}^{n/2} \widehat{\rho}(t) \; . \tag{8}$$

In statistics, an estimator, depending on the sample size n, is called *consistent* if it converges (in probability, or almost surely, depending) to the right answer as $n \to \infty$. In practice, we usually prove convergence in probability by showing that the bias and the variance of the estimator go to zero as $n \to \infty$. The estimator (8) is inconsistent in that sense. The bias goes to zero, but the variance does not go to zero. Most of the signal (values of t where $\rho(t)$ is significantly different from zero) is gone by, say $t = 5\tau$. But the noise continues all the way to $t = n/2$.

An informal quantitative analysis suggests that variance of $\widehat{\tau}_{inc,n}$ has a finite non-zero limit as $n \to \infty$. Suppose that the there were n_{eff} independent samples used for each value of $\widehat{\rho}(t)$. Then these estimates would have statistical error with variance of order $1/n_{\mathit{eff}}$. The sum (8) contains roughly n_{eff} independent terms with that variance, so the sum of the variances is order 1. It is easy (though time consuming) to verify this in specific examples.

We use a *self consistent window* strategy to capture most of the signal in $\widehat{\rho}$ without taking more noise than necessary. We do the sum in (7) over a *window* whose size is a multiple of τ.

$$\widehat{\tau}_{sc} = 1 + 2 \sum_{t=1}^{w\widehat{\tau}_{sc}} \widehat{\rho}(t) \; . \tag{9}$$

The parameter w is the *window size*, which I typically take to be $w = 5$ or $w = 10$. The estimator is *self consistent* because the estimated τ is used in the formula that estimates τ. The following code (which is not robust enough for a real MCMC code) does this:

```
tau_hat = 1.;
t       = 1;
while ( w*tau_hat > t )
   tau_hat += rho_hat[t++];
```

3 Discrete probabiltiy

We describe MCMC sampling first in a case that is not very technical so the main ideas are clear. Afterwards, we discuss continuous probability distributions. We start with two examples of discrete distributions. In both cases the state space is enormous but finite.

3.1 Example, a mixing/separation model

Two or more chemical species may mix or not mix. Oil and water do not mix. Sugar and water do mix, but the amount of sugar that can dissolve in water depends on the temperature. We give a qualitative model from statistical physics that explains the temperature dependence of mixing in some cases.

Suppose there are m atoms of species A. Each atom *occupies* a *site* in a *lattice* of $n \times n$ possible sites. A lattice site is described by integer coordinates (i, j) with $i \in \{1, \ldots, n\}$ and similarly for j. Each site is either occupied (by a species A atom) or *unoccupied.* A *configuration*, X, is the set of occupied sites, which is an m element subset of the n^2 sites. The *state space*, $\mathcal{S}$, is the set of all possible states. The number of states in this example is $|\mathcal{S}| = \binom{n^2}{m}$. If $n = 10$ and $m = 20$, this is about 2×10^{20}.

The *neighbors*, or *nearest neighbors*, of site (i, j) are the four sites $(i-1, j)$, $(i+1, j)$, $(i, j-1)$, and $(i, j+1)$. A *bond*, or *link*, or *edge*, in the lattice is the connection between neighboring sites. There is a bond between (i, j) and $(i+1, j)$, and a bond between (i, j) and $(i, j+1)$. The boundary of the lattice consists of sites with $i = 1$, $i = n$, $j = 1$, or $j = n$. *periodic* boundary conditions is the model specification that opposite edges are neighbors. That means, for example, (n, j) is a neighbor of $(1, j)$. *Open*, or *free*, boundary conditions is the specification that boundary sites have fewer neighbors. For example, the neighbors of $(4, n)$ would be $(3, n)$, $(5, n)$, and $(4, n-1)$, and the neighbors of $(n, 1)$ would be $(n-1, 1)$, and $(n, 2)$. The reason to use periodic boundary conditions is to avoid having to model side/void interactions and to make every lattice site look like every other site. It is not because we think a large lattice has periodic occupations.

A *graph* is an abstract generalization of this model. You define a graph by giving its *vertex set* and its *edges*. The vertex set is just a set. An *edge* is a subset of two vertices. Vertices v_1 and v_2 are *neighbors* in the graph if $\{v_1, v_2\}$ is an edge of the graph. In our mixing model, the vertex set is the set of lattice points. The edges are the bonds connecting nearest neighbors. The *complete graph* has n vertices and an edge connecting each pair of vertices – $n(n+1)/2$ edges in all. A *random graph* has n vertices. A pair $\{v_i, v_j\}$ is an edge with probability p, with all choices being independent.

Our lattice model of mixing has the physical hypothesis that atoms of species A attract each other. You model the attraction using the corresponding potential energy, which is the amount of energy it takes to pull apart a neighboring pair of A atoms. The overall energy of a configuration X is a sum over edges between neighbors. Each edge that has both sites occupied by A type atoms contributes $-h$ to the total energy. The energy is negative because you have to add energy to pull an A atom pair apart. The total potential energy is called $\phi(x)$ and is given by

$$\phi(x) = -h \sum_{e \in \text{edges}} \mathbf{1}_e \ , \tag{10}$$

where $\mathbf{1}_e = 1$ if both sites are occupied with A atoms, and $\mathbf{1}_e = 0$ otherwise. The maximum energy, which is zero, is achieved if the A type atoms are arranged in a checkerboard pattern. This is possible only if $m \leq n^2/2$. Low energy configurations have all occupied sites in a square of neighboring sites.

In a statistical equilibrium (technically, a *cannonical ensemble*), the probability of configuration X is given by

$$f(x) = \frac{1}{Z} e^{-\phi(x)/k_B T} \ . \tag{11}$$

The normalization constant Z is the *partition function* is a sum over all possible configurations

$$Z(T) = \sum_{x \in \mathcal{S}} e^{-\phi(x)/k_B T} \ . \tag{12}$$

The normalization constant is usually not known. It is important that the sampling algorithm be able to work without knowing Z. As usual, T is the temperature, in degrees above absolute zero. k_B is Boltzmann's constant, which is a conversion factor between degrees and units of energy.

The Gibbs distribution (11) says that low energy states are more likely than high energy states. The strength of this preference is determined by the temperature. At low temperature, low energy states are very strongly preferred over low energy states. In the limit $T \to \infty$, all states are equally likely. Ice is an example of a low temperature situation. Water molecules arrange themselves in a periodic crystal lattice that minimized their energy. Steam (gas phase water) is an example the same material at high temperature, where all configurations of water molecules are possible.

The behavior of this model, as with many models in statistical physics, is a competition between the influences of energy and entropy. *Entropy* refers to combinatorics: how many states are there with a given energy. Often there are more high energy states than low energy states. This can make it more likely to see a high energy state than a low energy state. Any particular high energy state is less likely than any particular low energy state. But there are so many more high energy states that high energy is more likely than low energy.

As an example, suppose there are only two type A atoms in the lattice. The two possible energy values are $-h$, if the two type A atoms are neighbors, and 0 if they are not. The number of "neighbor" configurations is $2n^2$ – the configuration could be vertical or horizontal (two possibilities), and the lower (if vertical) or left (if horizontal) occupied site can be any site. The total number of configurations is

$$|\mathcal{S}| = \binom{n^2}{2} = n^2(n^2-1)/2 = n^4/2 + O(n^2) \; .$$

The total number of energy zero configurations is $|\mathcal{S}|$ minus the number of energy $-h$ configurations, which is $n^4/2 + O(n^2) - 2n^2 = n^4/2 + O(n^2)$. Therefore the probability of energy zero is

$$P(\phi(X) = 0) = \frac{\frac{n^4}{2} + O(n^2)}{\left(\frac{n^4}{2} + O(n^2)\right) + 2n^2 e^{h/k_BT}} \approx \frac{1}{1 + \frac{4e^{h/k_BT}}{n^2}} \; .$$

This is significantly less than one only when $\frac{4e^{h/k_BT}}{n^2}$ is significantly different from zero. The larger n is, the smaller T has to be to make a $\phi = -h$ state as likely as a $\phi = 0$ state.

The problem is to understand how the system mixes as a function of the temperature and the *density*, which is $\rho = m/n^2$. We imagine a large system, large n, with a fixed ρ. A mixed state would be occupied sites distributed more or less uniformly in the lattice. A *separated* state would be most of the occupied sites in one high density region while most of the rest of the sites are not occupied. Separated states have lower energy and are preferred at low temperature. We will use Monte Carlo to investigate this quantitatively.

3.2 Example, Chemical potential

Chemical potential, denoted by μ, is the energy needed to add one atom or molecule of something. The very simplest probability distribution involving chemical potential just has N copies of a molecule. The energy for n copies is

$\phi(n) = \mu n$. The probability of having n copies is

$$f_n = \frac{1}{Z(\mu)} e^{-\mu n} \; . \tag{13}$$

Of course,

$$Z(\mu) = \sum_{n=0}^{\infty} e^{-\mu n} = \frac{1}{1 - e^{-\mu}} \; .$$

The state space is not finite, strictly speaking. You can ignore this, or truncate the system with a maximum allowed n. There is a simple direct sampler for this distribution, which we also ignore for the time being.

3.3 Example, unobserved regime switching

A *hidden Markov model* is a model of a sequence of observations that partially describe the state of a Markov chain. Here is a simple example. At each discrete time j there is a binary variable, $X_j \in \{0,1\}$, that we call the *state*, and a binary variable, Y_j, that we call the *observation*. Three parameters characterize the model, p, q_0, and q_1. In one time step, X flips with probability p. That is $P(X_{j+1} \neq X_j) = p$. If $X_j = 0$, then $Y_j = 1$ with probability q_0. If $X_j = 1$, then $Y_j = 1$ with probability q_1.

A general hidden Markov model is like this example. There is a state X_j that is in some state space, $\mathcal{S}$, that is in general more complicated than $\{0,1\}$. The sequence of states forms a stationary Markov chain. For each j there is an observation Y_j that is a random function of X_j. More formally, for each $x \in \mathcal{S}$, there is a probability distribution $g(y|x)$. The observation is a sample of this distribution: $Y_j \sim g(\cdot, X_j)$. The problem is to say what you can about the sequence X_j given the observations $(Y_1, \ldots, Y_T)$.

To get some intuition about our simple binary model, suppose that p is small, that q_0 is close to zero, and q_1 is close to 1. Then spin flips are rare and $Y_j = X_j$ most of the time. The observation sequence will consist of sequences of mostly zeros followed by sequences of mostly ones, and so on. A more interesting case is p small, but $q_0 = .5 - \varepsilon$ and $q_1 = .5 + \varepsilon$. Flips are as rare as before, but harder to identify the spin flips from an observation sequence. It is unlikely to see long sequences of mostly zeros or mostly ones. Instead, there will be periods when ones are slightly more common than zeros, or the other way around. It will be hard to say whether a sequence with slightly more ones than zeros is do to $X_j = 1$ or just random chance.

A major advance in statistics is to take the simple Bayesian point of view that you answer study this question simply by generating samples of the *posterior* distribution, which is the conditional distribution $f(x \mid Y)$. This is the distribution of X conditional on the data Y. We write this distribution using some convenient notation: The transition matrix for the X process is

$$R(x, x') = \begin{cases} 1 - p & \text{if } x = x' \\ p & \text{if } x \neq x' \end{cases}$$

The observation distribution is $g(y \mid x)$ given by

$$g(y \mid x) = \begin{cases} 1 - q_0 & \text{if } y = 0,\ x = 0 \\ q_0 & \text{if } y = 1,\ x = 0 \\ 1 - q_1 & \text{if } y = 0,\ x = 1 \\ q_1 & \text{if } y = 1,\ x = 1 \end{cases}$$

The probability of a particular sequence $(x, y) = (x_1, \ldots, x_T, y_0, \ldots, y_T)$ is

$$P(X = x,\, Y = y) = F(x, y) = \prod_{j=0}^{T-1} R(x_j, x_{j+1}) \prod_{j=0}^{T} g(y_j \mid x_j) \ .$$

For simplicity, we assume that x_0 is known. Once the value $y = Y$ is known, the conditional probability of X is

$$P(X = x \mid Y) = \frac{1}{Z(Y)} F(x, Y) \ . \tag{14}$$

It is easy to evaluate $f(x, Y)$ for any particular x and Y. The normalization constant

$$Z(Y) = \sum_{x \in \mathcal{S}} F(x, Y)$$

is hard to know. The state space $\mathcal{S}$ consists of all binary sequences of length T. The size of this state space is $|\mathcal{S}| = 2^T$.

3.4 Discrete Markov chain Monte Carlo

Here is a quick review of Markov chain theory. Let $\mathcal{S}$ be a finite state space of size n. Let R be an $n \times n$ transition matrix for the stationary Markov chain X_k. That means that

$$R_{ij} = P(i \to j) = P(X_{t+1} = j \mid X_t = i) \ .$$

Suppose $f_{0,i} = P(X_0 = i)$ is the starting probability distribution. Then the probability of a sequence $x_0, \ldots, x_T$ is

$$\mathrm{P}(X_0 = x_0, X_1 = x_1, \ldots, X_T = x_T) = f_{0,x_0} \prod_{j=0}^{T-1} R_{x_j, x_{j+1}} \ .$$

For example, we find the joint distribution of X_0 and X_2 by summing over x_1

$$\mathrm{P}(X_0 = i, X_2 = k) = \sum_k f_{0,i} P_{ik} P_{kj} = f_{0.i} \left(P^2\right)_{ij} \ .$$

The probability distribution of X_t is the row vector f_t with components $f_{t,j} = P(X_t = j)$. These numbers satisfy

$$f_{t+1,j} = P(X_{t+1} = j) = \sum_{k \in \mathcal{S}} P(X_{t+1} = j \mid X_t = k) P(X_t = k) = \sum_k f_{t,k} R_{kj} \ .$$

In matrix notation, this is simply

$$f_{t+1} = f_t R \; . \tag{15}$$

If you think of f_t as a column vector, the same equation would be $f_{t+1} = R^* f_t$. (We write R^* for the transpose of R so that R^t can be R to the power t.)

A distribution is *invariant* if $f = fR$. In MCMC, we are given f and we need to create a non-degenerate R, that can be implemented, so that f is an invariant distribution of R. A non-degenerate R determines the invariant probability distribution f uniquely. But there are many transition matrices that preserve a given f. Finding a good R for a given f, that is one if the central research problems in modern Monte Carlo.

In a Markov chain generated by R, the probability to go from $X_0 = i$ to $X_t = j$, which is an $i \to j$ transition in t steps, is $\left(P^t\right)_{ij}$. The Markov chain is *irreducible* every transition is eventually possible. That means, for each pair (i, j), there is a $t > 0$ so that $\left(P^t\right)_{ij} > 0$. A Markov chain on a finite state space is *aperiodic* if there is a $t > 0$ so that $\left(P^t\right)_{ij} > 0$ for all (i, j). It is "elementary" (takes under an hour) to see that if an irreducible Markov chain on a finite state space is not aperiodic, then it is indeed *periodic* in that there is a state i and a period, r so that $\left(P^t\right)_{ii} = 0$ if t is not a multiple of r.

The basic theory of finite state space Markov chains includes theorems that if R is non-degenerate, which means irreducible and aperiodic, then f (the invariant probability distribution) is unique, $f_t \to f$ as $t \to \infty$, and the sample means (1) converge to the right answer. The conclusion is this. You can sample f well enough to compute expectations by running a non-degenerate Markov chain that has f as its invariant distribution. An MCMC sampler of f is such a Markov chain.

3.5 Balance and detailed balance

The fact that R preserves f may be understood as a balance condition. If $X_t \sim f$, then the probability that $X_{t+1} = i$ is the probability that $X_t = i$, minus the probability of a transition out of the state i, and plus the probability of a transition into state i. If f is an invariant distribution, then these in and out probabilities balance for each state, i. The probability of an out transition is

$$\begin{aligned} \mathrm{P}(\text{observe } i \to \text{ not } i) &= \mathrm{P}(X_t = i)\,\mathrm{P}(i \to \text{ not } i) \\ &= f_i \sum_{j \neq i} P_{ij} \; . \end{aligned}$$

The probability of a transition into i is

$$\mathrm{P}(\text{observe not } i \to i) = \sum_{j \neq i} f_j P_{ji} \; .$$

The balance condition comes from setting these equal to each other:

$$f_i \sum_{j \neq i} P_{ij} = \sum_{j \neq i} f_j P_{ji} \ . \tag{16}$$

We put this in a more familiar form by adding the $i \to i$ "transition" term, which is $f_i P_{ii}$, to both sides. On the left we have

$$f_i \sum_j P_{ij} = f_i \ .$$

On the right we have

$$\sum_j f_j P_{ji} = (fP)_i \ .$$

Setting these equal gives the familiar $f = fP$. We conclude that the balance condition is equivalent to the condition that f is an invariant distribution of P.

The *detailed balance* condition is a refinement of the above balance condition. An R that satisfies detailed balance for f obviously (as we will see) satisfies overall balance. But it is possible to have overall balance without detailed balance. The value of detailed balance is that it is easy to check because it does not involve a sum over all states. Overall balance does involve such sums. If we understood enough about f do to such sums, we would not need to do MCMC.

Detailed balance is a balance condition for each pair of states.

$$\begin{aligned} \mathrm{P}(\text{observe } i \to j) &= \mathrm{P}(\text{observe } j \to i) \\ f_i P_{ij} &= f_j P_{ji} \quad \text{for all pairs } (i,j). \end{aligned} \tag{17}$$

Summing this over j (obviously) the balance condition (refeq:b). When we design MCMC algorithms, we have f and we are looking for P. The detailed balance condition (19) is an equation that relates just two numbers P_{ij} and P_{ji}. Each transition probability occurs in just one of these equations. It is easy to find numbers P_{ij} because the equations (19) are all independent of each other (almost).

3.5.1 Detailed balance

Detailed balance is a simple practical way to construct MCMC samplers. Statistical physics has a *principle* of detailed balance, which is the statement that certain systems should satisfy detailed balance. MCMC has no principle of detailed balance. Rather, detailed balance is a trick that allows you to find matrices R that preserve f. There are many correct MCMC samplers that do not satisfy detailed balance.

The ordinary *balance* condition is a way to say that R preserves f. You look at a particular state $x \in \mathcal{S}$ and ask that the probability of observing a transition out of x is the same as the probability of observing a transition into x. The two probabilities should balance. If $X \sim f$, then the probability of observing a transition out of x is

$$\sum_{y \neq x} P(\text{observe } x \to y) = \sum_{y \neq x} f(x) R_{xy} \ .$$

The probability to observe a transition into x is

$$\sum_{y \neq x} P(\text{observe } y \to x) = \sum_{y \neq x} f(y) R_{yx} \ .$$

The balance condition is that these are equal, which is

$$\sum_{y \neq x} f(x) R_{xy} = \sum_{y \neq x} f(y) R_{yx} \ . \tag{18}$$

This is equivalent to the steady state condition $f = fR$, which you can see by adding $f(x)R_{xx}$ to both sides. On the left, we have

$$f(x) \sum_{y \in \mathcal{S}} R_{xy} = f(x) \ .$$

On the right we have

$$\sum_{y \in \mathcal{S}} f(y) R_{yx} \ ,$$

which is the the x component of fR.

Detailed balance is the balance condition for each $x \neq y$ pair, $P(\text{observe} x \to y) = P(\text{observe } y \to x)$:

$$f(x) R_{xy} = f(y) R_{yx} \ . \tag{19}$$

It is clear that if R satisfies detailed balance for f, then it satisfies ordinary balance. If you sum (19) over $y \neq x$, you get (18). There are many ways to satisfy ordinary balance without detailed balance. This being said, most MCMC strategies use detailed balance in some way.

3.5.2 Metropolis, Metropolis Hastings

The *Metropolis* method achieves detailed balance through rejection. As with rejection sampling, is uses a *proposal* distribution, Q, and an acceptance probability, A. If you are in state x, you propose to move to a random $Y \in \mathcal{S}$ with

$$Q_{xy} = P(\text{propose } Y = y \text{ from } x) \ .$$

Of course, the Q matrix must be a probability distribution as a function of y for each x, which means $Q_{xy} \geq 0$, and $\sum_{y \in \mathcal{S}} Q_{xy} = 1$ for every x. The *acceptance* probability of a proposed move is A_{xy}. If the proposed Y is accepted, then it becomes the new state. If the proposed Y is rejected, the new state is the same as the old state.

We express R in terms of Q and A. In order to make a transition from x to y, the state y must be proposed and then accepted. This means that if $y \neq x$, then

$$R_{xy} = P(x \to y) = P(\text{propose } y \text{ from } x) P(\text{accept } y) = Q_{xy} A_{xy} \; . \tag{20}$$

There are two ways to have an $x \to x$ transition. One is to propose $x \to x$. The probability of this is Q_{xx}. Many common Metropolis samplers never propose $x \to x$. The other way to get $x \to x$ is to reject the proposed move. Altogether, the probability of getting $x \to x$ is

$$R_{xx} = Q_{xx} + \sum_{y \neq x} Q_{xy}(1 - A_{xy}) \; .$$

This formula is consistent with the convention $A_{xx} = 1$. There is no point in rejecting a proposed $x \to x$ proposal, because that gives $x \to x$ just as accepting does.

Once Q and f are given, it is possible to choose A so that the resulting R satisfied detailed balance. The detailed balance condition (19) for the Metropolis formula (20) is

$$f(x) Q_{xy} A_{xy} = f(y) Q_{yx} A_{yx} \; . \tag{21}$$

This determines the ration A_{xy}/A_{yx}. The Metropolis method chooses the largest acceptance probabilities consistent with (21) and the constraint $A_{xy} \leq 1$ for all x, y. You can get the explicit formula by writing (21) in the form

$$A_{xy} = \frac{f(y) Q_{yx}}{f(x) Q_{xy}} \, A_{yx} \; .$$

If $A_{yx} = 1$, this gives

$$A_{xy} = \frac{f(y) Q_{yx}}{f(x) Q_{xy}} \; .$$

If this number is ≤ 1, we should use it, and take $A_{yx} = 1$. Those would be the largest possible acceptance probabilities. Otherwise, we take $A_{xy} = 1$. This reasoning leads to

$$A_{xy} = \min\left(1, \; \frac{f(y) Q_{yx}}{f(x) Q_{xy}} \right) \; . \tag{22}$$

The detailed balance formula (21) is a relationship between A_{xy} and A_{yx}. The Metropolis choice (22) makes at least one of them equal to one. It gives both of them the largest possible values consistent with detailed balance. It is common to use symmetric proposal distributions. If $Q_{xy} = Q_{yx}$, then (22) simplifies to

$$A_{xy} = \min\left(1, \; \frac{f(y)}{f(x)} \right) \; . \tag{23}$$

The original paper of Metropolis, Rosenbluth, Rosenbluth, Teller, and Teller had a symmetric proposal and used (23). For that reason, (23) is the *Metropolis*, or the MR^2T^2 acceptance probability. The generalization to non-symmetric proposals is due to Hastings, so (22) is often called *Metropolis Hastings.*

Here is a Metropolis sampler for the chemical potential problem (13). The proposal is $n \to n \pm 1$ with equal probability. The proposal distribution is symmetric, so we use (23). If $x = n$ and $y = n+1$, we get $f(n+1)/f(n) = e^{-\mu} < 1$. Similarly, $f(n-1)/f(n) = e^{\mu} > 1$. Therefore

$$\begin{aligned} A_{n,n-1} &= \min\left(1,\;\; e^{\mu}\;\right) = 1 \\ A_{n,n+1} &= \min\left(1, e^{-\mu}\right) = e^{-\mu} \;. \end{aligned}$$

Of course, a proposal $0 \to -1$ must be rejected. Recall that the state $n+1$ is less likely than the state n, because it takes an extra $e^{-\mu}$ of energy to add another copy. A proposal to move from a state to a less likely state is sometimes rejected, which is how the MCMC makes it less likely to occupy $n+1$ than n. A proposal, $n \to n-1$, to move to a more likely state is always accepted. See Exercise 1.

Here is an outline of how you might code the MCMC sampler.

```
int nSamp( n, mu) {   //  Take one MCMC step for the
                      //  chemical potential problem
    int np;           //  n-prime is the proposed new value
    double A;         //  The acceptance probability
    if ( rand() < .5 ) {
        np = n - 1;
        A  = 1.;
      }
    else          {
        np = n + 1;
        a  = exp( - mu);
      }
    if ( np < 0 ) return n          // Reject a proposal to -1
    if ( rand() < A ) return np;    // Accept np with probability A
    else              return n;     // If reject, keep n
  }
```

Here is a Metropolis MCMC sampler for the mixing/separation problem from Subsection 3.1. You choose a bond at random with all bonds equally likely to be chosen. Then you propose to exchange the sites at the ends of this bond. This proposal also is symmetric. You compute $\Delta\phi$, which is a local calculation involving six sites. If $\Delta\phi \leq 0$, the proposed new configuration is more likely than the current configuration, you accept the new one. If $\Delta\phi > 0$, you accept with probability $e^{-\Delta\phi/k_BT}$. If the proposed exchange is rejected, the configuration does not change.

Here is an MCMC Metropolis sampler for the binary hidden Markov model. The random object is a path $X = (X_1, \ldots, X_T)$. The proposed *move* is to choose $t \in \{1, \ldots, T\}$ at random and flip the value of X_t, $0 \leftrightarrow 1$. Exercise 2 is the computation of the acceptance probability. This MCMC algorithm is *correct* in the sense that it samples the correct distribution. But it is a bad method for small p because it is likely to be inefficient. Most moves are rejected because flipping X_t probably makes the proposed X path much less likely than X itself. See Exercise 2.

3.5.3 Non-degeneracy conditions

A Markov chain on a finite state space is *ergodic* if it has the following property. There is an $\varepsilon > 0$ and a T so that $R_{ij}^T \geq \varepsilon$ for all $i \in \mathcal{S}$ and $j \in \mathcal{S}$. If a Markov chain is not ergodic, it is *reducible* or *cyclic* or both. Reducible means that there is an $i \in \mathcal{S}$ and $j \in \mathcal{S}$ so that $R_{ij}^t = 0$ for all t. This means, if you start in state i, it is impossible ever to reach state j in any number of steps. An *irreducible* Markov chain has the property that for every (i, j), there is a t with $R_{ij}^t > 0$. To be ergodic, you have to be able to take the same t value for every (i, j). Not every irreducible chain has this property.

State i in a Markov chain is *acyclic* if there is a $T > 0$ so that $R_{ii}^t > 0$ for all $t \geq t$. For example, suppose $\mathcal{S} = \{0, 1, 2, 3\}$ and a step is $X \to X \pm 1 (\text{mod } 4)$. The probabilities do not matter as long as they are all non-zero. If $X_0 = 0$, then $X_t = 0$ implies that t is even, so $R_{00}^t = 0$ whenever t is odd. This example is typical. It is possible to show (think about this for an hour or so) that if state i is cyclic then there is a $d > 1$ so that $R_{ii}^t > 0$ only if t is a multiple of d. This d is the "cycle" length related to the term *cyclic*. A Markov chain with no cyclic states is acyclic.

It is usually easy to decide whether a specific Markov chain is irreducible and acyclic. All the MCMC chains described above are clearly irreducible. MCMC samplers that have a rejection step usually have $R_{ii}^t > 0$ for all t: just get a string of t rejections. There may be extreme states that have no rejections, the states with maximum energy and least probability. For those you can use the observation that if a chain is irreducible and there is any acyclic state, then every state is acyclic.

It is possible to make a mistake and suggest an MCMC sampler that is not irreducible. An example is part (c) of Exercise 2. A reducible MCMC sampler that was used for scientific computing (and therefore gave the wrong answer in principle) is in Exercise 3.

It remains to prove that if a Markov chain is irreducible and acyclic, then it satisfies (**??**). This involves several theorems. One is the *Perron Frobenius* theorem for an acyclic and irreducible Markov chain, which states that

- This is a row vector with all non-negative entries that satisfies $f = fR$. This can be normalized to be a probability distribution, i.e., $\sum f_j = 1$.
- The eigenvalue $\lambda = 1$ of R is simple. If $g = gR$, then $g = cf$ for some constant c. There is no Jordan structure for $\lambda = 1$.
- If λ is an eigenvalue of R, then either $\lambda = 1$ or $|\lambda| < 1$.
- If $X_t \sim f_t$, then $f_t \to f$ as $t \to \infty$

The Markov chains of MCMC are designed to have a given f as invariant distributions. Perron Frobenius states that the probability distribution of X_t converges to f regardless of the distribution of X_0. If R preserves f, then R samples f.

The other theorem is the *ergodic theorem* for Markov chains. This states that the right side of (**??**) converges to the left side almost surely as $T \to \infty$.

These are not hard to prove, but the proofs do not fit into Week 2.

4 Continuous probability

MCMC applies to sampling probability densities. Suppose $f(x)$ is the probability density for $X \in \mathbb{R}^n$. An MCMC sampler of f is a non-degenerate Markov chain that preserves f. The ideas are similar to the discrete case, but you have to be more careful reasoning about random transformations R and probability densities,

A *transition probability density* is a function $R(x, y)$ so that $X_{t+1} \sim R(x, \cdot)$ if $X_t = x$. A mathematician's statement might be

$$P(X_{t+1} \in A \mid X_t = x) = \int_A R(x, y)\, dy \;.$$

If $f_t(x)$ is the probability density of X_t, then the analogue of $f_{t+1} = f_t R$ is

$$f_{t+1}(y) = \int_{\mathbb{R}^n} f_t(x) R(x, y)\, dy \;.$$

An invariant probability density has $f = fR$, which is

$$f(y) = \int f(x) R(x, y)\, dy \;, \tag{24}$$

for every x.

Most MCMC methods used in practice do not have transition densities in the strict sense. They usually have $\delta-$function components of some sort.

4.1 Detailed balance, global Metropolis

The detailed balance condition is

$$f(x) R(x, y) = f(y) R(y, x) \;. \tag{25}$$

Detailed balance implies balance, as it does in the discrete case. If you integrate (25) over y, you find (24). You can create transition densities using the proposal/rejection idea. Suppose $Q(x, y)$ is a proposal probability density and $A(x, y)$ is an acceptance probability function. A Metropolis strategy would be to propose $X_t \to Y \sim Q(x, \cdot)$ and then accept Y with probability $A(X_t, Y)$. If

Y is accepted then $X_{t+1} = Y$. Otherwise $X_{t+1} = X_t$. The resulting probability density is $R(x,y) = Q(x,y)A(x,y)$. The detailed balance condition (25) is satisfied if A is given by the Metropolis Hastings rule

$$A(x,y) = \max\left(1, \, \frac{f(y)Q(y,x)}{f(x)Q(x,y)}\right) \; . \tag{26}$$

The formula and the derivation are the same as the discrete case. If Q is symmetric, then proposals to higher probability states always get accepted.

The statements are not quite true, but the algorithm is OK. The transition distribution for the Metropolis method clearly has a $\delta-$function component that corresponds to rejected proposals. A more correct formula is

$$R(x,y) = R_A(x,y) + (1 - Z(x))\delta(x-y) \; ,$$

where $Z(x)$ is the acceptance probability for a proposal from x:

$$Z(x) = \int Q(x,y)A(x,y)\,dy \; .$$

This presumes that there is no $\delta-$function component in Q. It is reasonable to think that you would not propose to stay at x. The part of R that does have a density is

$$R_A(x,y) = Q(x,y)A(x,y) \; .$$

This is analogous to the discrete formula (20), but we are talking about probabilities there and probability densities here.

It is common to use simple proposal distributions that correspond to a random step from x of a certain *proposal step size*, r. For example, one could propose Y to be a Gaussian centered at x with covariance $e^2 I$. That means

$$Q(x,y) = Ce^{-\sum_{k=1}^{n}(x_k - y_k)^2/(2r^2)} \; .$$

Another possibility is $Y - x$ uniformly distributed in a ball in $\mathbb{R}^n$ of radius r. An exercise last week explained how to sample that distribution.

The proposal step size is a parameter in the MCMC algorithm. *Tuning* (more recently, *training*) the algorithm means finding a good r for a specific problem. If r is too small, then Y is close to x, $f(Y)$ is close to $f(x)$, and the proposal is likely to be accepted. But it takes many such small steps to get a Y that is significantly different from x. The algorithm is slow. If r is too large, then most proposals are rejected because the proposed Y is too far from an x that is likely in the probability distribution f. This also given an inefficient algorithm. There is a “rule of thumb” in the Monte Carlo community that the optimal r is the one that gives overall acceptance probability .5, or maybe .4 or .6 of something. Anyway, it should not be too close to zero (too many rejections) or too close to one (steps smaller than necessary).

4.2 Partial resampling

Resampling means replacing $X \sim f$ with a different $X' \sim f$. In that sense, any MCMC algorithm is resampling. *Partial* resampling means resampling only part of X. Suppose $X = (U, V)$, where U is one set of variables and V is the complementary set. An example is $U = (X_1, \ldots, X_k)$, and $V = (X_{k+1}, \ldots, X_n)$. We have $U \in \mathbb{R}^k$ and $V \in \mathbb{R}^{n-k}$, so $X = (U, V) \in \mathbb{R}^n$. *Partial resampling* means, in this case, resampling just U without changing V. The conditional probability density of U given V is

$$f(u \mid V) = \frac{1}{Z(V)} f(u, V) \; .$$

An MCMC algorithm for resampling U would have a probability density function $R(u, u' \mid V)$. This preserves the conditional distribution if

$$f(u' \mid V) = \int f(u \mid V) R(u, u' \mid V) \, du \; . \tag{27}$$

Partial resampling is based on the following principle: If $X = (U, V) \sim f$, and $U' \sim R(U, \cdot \mid V)$, and R satisfies (27), then $X' = (U', V) \sim f$. We prove this principle by showing that $X' \sim f$. This is simple, once you get the definitions right. The hypotheses are that $X \sim f$ and $U' \sim R(U, \cdot \mid V)$. Let[1]

$$f(v) = \int_{\mathbb{R}^k} f(u, v) \, du$$

be the marginal density of V. Then then density of $X = (U, V)$ is $f(u, v) = f(u \mid v) f(v)$. Resampling U gives U' with density $f(u' \mid V)$ (because of (27)), so the pair $X' = (U', V)$ has density $f(u' \mid v) f(v) = f(u', v)$.

Single variable resampling is partial resampling where U is one component of X and V is the rest. That is $U = X_j$, and $V = (X_1, \ldots, X_{j-1}, X_{j+1}, \ldots, X_n)$. You need a single variable resampler $R(x_j, x_j' \mid V)$ to do this. You could use a direct sampler that produces an independent sample $X' \sim f(x_j|V)$. This is the single variable *heat bath* algorithm, also called the *Gibbs sampler*. If you understand the one variable conditional distributions well enough, you should be able to make an efficient rejection sampler, for example.

Otherwise, you could apply Metropolis to the X_j variable with all the other components held fixed. This is *single variable Metropolis*. As with the global Metropolis, you probably have to tune the proposal distribution so that the acceptance probability is not too close to zero or one.

4.3 Hybrid samplers

Suppose you have two MCMC samplers, R_1 and R_2. You can put them together into a single sampler by first doing R_1 then doing R_2:

$$X \overset{R_1}{\to} X' \overset{R_2}{\to} X'' \; .$$

[1] People in machine learning often use f for any probability density. The probability density of X is $f(x)$. The density of V is $f(v)$, etc. It is possible that adding a subscript, as $f_V(v)$ is clearer, but only a little.

it is "obvious" that the composite sampler preserves f. If $X \sim f$, the correctness of R_1 implies that $X' \sim f$. The correctness of R_2, and $X' \sim f$, implies that $X'' \sim f$. It is possible to combine any number of correct samplers in this way, clearly.

One application of this idea is to cycle through the components of X using a single variable resampler for each. Suppose $R_j(x_j \mid V_j)$ resamples X_j with the complementary variables V_j fixed. Using these in any order, say $R_1, R_2, \ldots, R_n$, gives a resampler that changes all the components of X. A single variable resamplers on its own cannot give an irreducible Markov chain. If you resample X_1 over and over, you never change any of the other variables. But resampling each variable in turn probably gives a non-degenerate (irreducible and acyclic) MCMC sampler.

Is the single variable heat bath method better than, say, a well tuned global Metropolis? Sometimes it is. Sometimes it is not. It depends on details of the problem.

5 Examples and exercises

1. Write a formula for the transition probabilities $R_{n,n-1}$, R_{nn}, and $R_{n,n+1}$ for the simple Metropolis sampler for the chemical potential problem. Verify by explicit calculation that if $n > 0$, then $f_{n-1}R_{n-1,n} + f_n R_{nn} + f_{n+1}R_{n+1,n} = f_n$.

2. In the binary hidden Markov model,

   (a) Find the formula for the acceptance probability if the proposal is flipping the value of X_t. This depends on X_{t-1}, X_{t+1}, X_t, and Y_t. (Be careful not to confuse the time variable in the X path with the time variable in the MCMC sampler. I made that confusion easier by using the same t and T variables for both purposes. Feel free to change notation.)

   (b) Consider the special case where $q_0 = q_1 = .5$, so the data do not effect the probabilities. How does the probability of accepting the proposal $(\cdots, 0, 0, 0, \cdots) \to (\cdots, 0, 1, 0, \cdots)$ depend on p in that case?

   (c) Show that the *exchange* move: $X_t, X_{t+1} \leftrightarrow X_{t+1}, X_t$ (with all other X_k unchanged) is much more likely to be accepted for small p and q_0 and q_1 near .5.

3. A *walk* on a 2D lattice is a sequence of L sites (i_k, j_k) for $k = 1, \ldots, L$, so that (i_{k+1}, j_{k+1}) is a neighbor of (i_k, j_k). The walk is *self avoiding* if $(i_k, j_k) \neq (i_m, j_m)$ if $k \neq m$. This is a qualitative model of a polymer, which is a long string of monomers that cannot overlap. In simple self avoiding walk, every allowed path has the same probability. The *snake move* is an MCMC sampler for SAW (self avoiding walk) that works as follows. You propose to delete the "tail" of the walk and add one step to

the “head”. This means: delete (i_1, j_1), renumber the remaining $L-1$ steps, and choose at random a neighbor for the end (i_{L-1}, j_{L-1}).

(a) Show that this satisfies detailed balance, possibly after adding a rejection step.

(b) Show that this is reducible. In fact, there are states in which the “head” is buried so that it cannot be extended at all.

(c) It was proposed to allow the snake to “reverse” and move from head to tail. Show that this also has states that cannot move.

4. A hybrid sampler $R = R_2 R_1$ may not satisfy detailed balance even if R_1 and R_2 do satisfy detailed balance.

5. Consider the hidden Markov model in the case p is small and $q_0 \approx .5$ and $q_1 \approx .5$. Exercise 2 shows that resampling X_j is unlikely to change X_j. However, we could resample the pair (X_j, X_{j+1}). If, for example, $(X_{j-1}, X_j, X_{j+1}, X_{j+2}) = (0, 0, 1, 1)$, it is not so unlikely to get $(X'_j, X'_{j+1}) = (1, 1)$ or $(0, 0)$. A $0 \to 1$ transition between j and $j+1$ could move to the right or the left by one.

6. Single variable resampling for the Gaussian distribution

$$f(x) = C \exp\left(-\frac{1}{2} \sum_{ij} x_i x_j h_{ij} \right) .$$