Chapter 9

Convergence and error estimation for MCMC

References:

Robert, Casella - chapter 12

chapter 8 of "handbook" - primarily on statitical analysis

Fishman chap 6

9.1 Introduction - sources of errors

When we considered direct Monte Carlo simulations, the estimator for the mean we were computing was a sum of independent random variables since the samples were independent. So we could compute the variance of our estimator by using the fact that the variance of a sum of independent random variables is the sum of their variances. In MCMC the samples are not independent, and so things are not so simple. We need to figure out how to put error bars on our estimate, i.e., estimate the variance of our estimator.

There is a completely different source of error in MCMC that has no analog in direct MC. If we start the Markov chain in a state which is very atypical for the stationary distribution, then we need to run the chain for some amount of time before it will move to the typical states for the stationary distribution. This preliminary part of the simulation run goes under a variety of names: burn-in, initialization, thermalization. We should not use the samples generated during this burn-in period in our estimate of the mean we want to compute. The

preceeding was quite vague. What is meant by a state being typical or atypical for the stationary distribution?

There is another potential source of error. We should be sure that our Markov chain is irreducible, but even if it is, it may take a very long time to explore some parts of the state space. This problem is sometimes called "missing mass."

We illustrate these sources of errors with some very simple examples. One way to visualize the convergence of our MCMC is a plot of the evolution of the chain, i.e, X_n vs n.

Example: We return to an example we considered when we looked at Metropolis-Hasting. We want to generate samples of a standard normal. Given $X_n = x$, the proposal distribution is the uniform distribution on $[x - \epsilon, x + \epsilon]$, where ϵ is a parameter. So

$$q(x, y) = \begin{cases} \frac{1}{2\epsilon} & \text{if } |x - y| \leq \epsilon, \\ 0, & \text{if } |x - y| > \epsilon, \end{cases} \tag{9.1}$$

(When we looked at this example before, we just considered $\epsilon = 1$.) We have

$$\alpha(x, y) = \min\{\frac{\pi(y)}{\pi(x)}, 1\} = \min\{\exp(-\frac{1}{2}y^2 + \frac{1}{2}x^2), 1\} \tag{9.2}$$

$$= \begin{cases} \exp(-\frac{1}{2}y^2 + \frac{1}{2}x^2) & \text{if } |x| < |y|, \\ 1 & \text{if } |x| \geq |y| \end{cases} \tag{9.3}$$

First consider what the evolution plot for direct MC would look like. So we just generate sample of X_n from the normal distribution. The evolution plot is shown in figure 9.1. Note that there is not really any evolution here. X_{n+1} has nothing to do with X_n.

The next evolution plot (figure 9.2) is for the Metropolis-Hasting algorithm with $X_0 = 0.$ and $\epsilon = 1.0$. The algorithm works well with this initial condition and proposal distribution. The samples are correlated, but the Markov chain mixes well.

The next evolution plot (figure 9.3) is for the Metropolis-Hasting algorithm with $X_0 = 0.$ and $\epsilon = 0.1$. The algorithm does not mix as well with this proposal distribution. The state moves rather slowly through the state space and there is very strong correlation between the samples over long times.

The next evolution plot (figure 9.4) is for the Metropolis-Hasting algorithm with $X_0 = 0.$ and $\epsilon = 100.$. This algorithm does very poorly. Almost all of the proposed jumps take the chain to a state with very low probability and so are rejected. So the chain stays stuck in the state it is in for many time steps. This is seen in the flat regions in the plot.

The next evolution plot (figure 9.5) is for the Metropolis-Hasting algorithm with $X_0 = 10.$ and $\epsilon = 0.1$. Note that this initial value is far outside the range of typical values of X. This

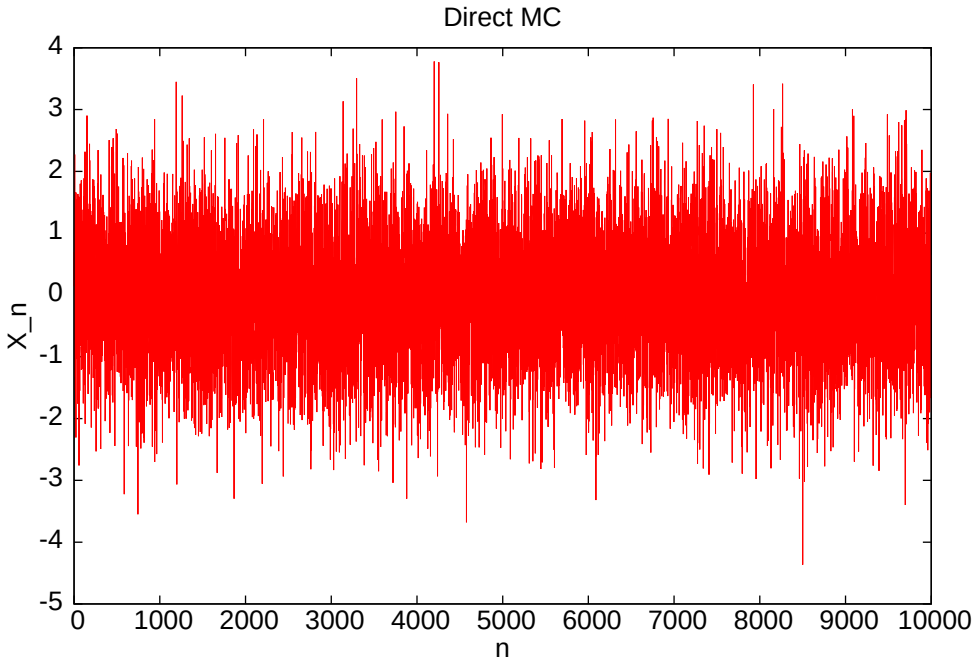

Figure 9.1: Direct Monte Carlo for the normal distribution.

algorithm has a significant burn-in period. It takes a couple thousand time steps before the chain is in a typical state. We need to discard the samples from this burn-in phase.

Now we change the distribution. We consider a mixture or two normal distributions. They both have variance 1; one is centered at 3 and one at -3. So the density is given by

$$f(x) \propto \exp(-\frac{1}{2}(x-3)^2) + \exp(-\frac{1}{2}(x+3)^2) \tag{9.4}$$

We use the same proposal distribution as before. The evolution plot for the Metropolis-Hasting algorithm for this distribution with $X_0 = 0.$ and $\epsilon = 1.0$ is shown in figure 9.6.

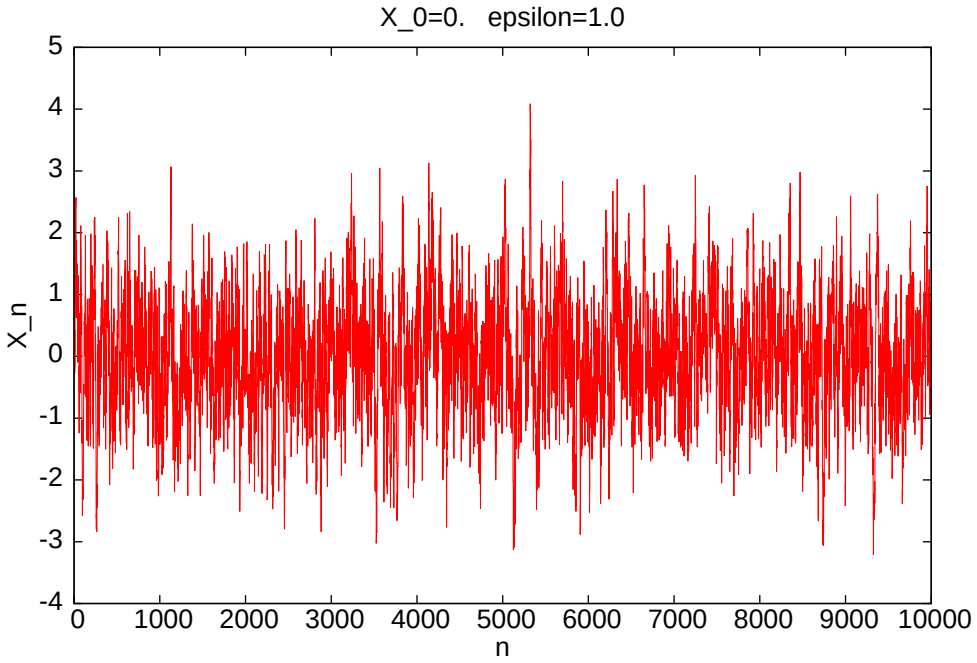

Figure 9.2: Metropolis Hasting for the normal distribution with $X_0 = 0$ and $\epsilon = 1.0$.

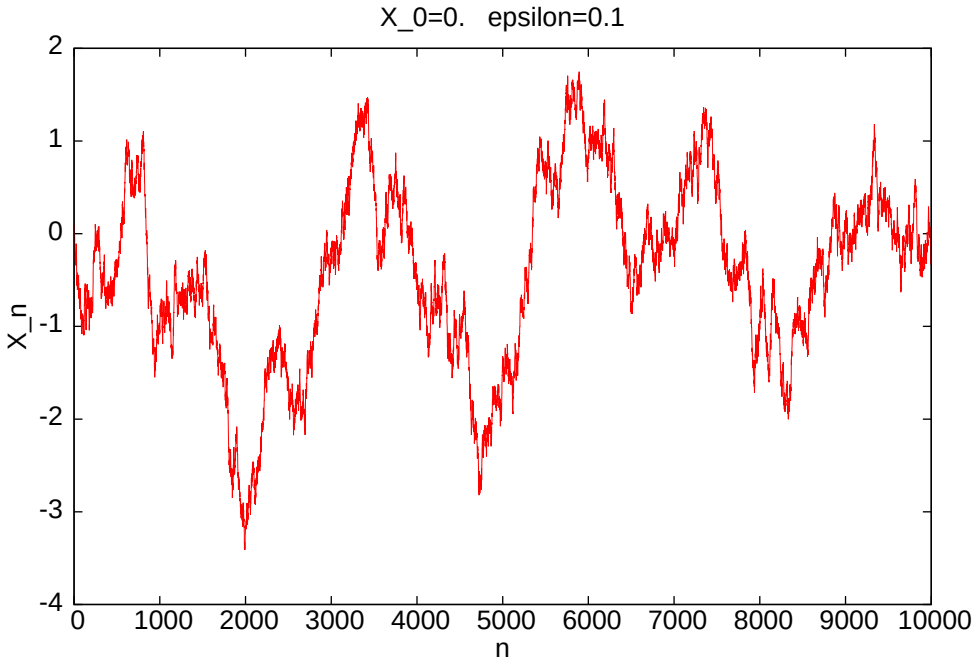

Figure 9.3: Metropolis Hasting for the normal distribution with $X_0 = 0$ and $\epsilon = 0.1$.

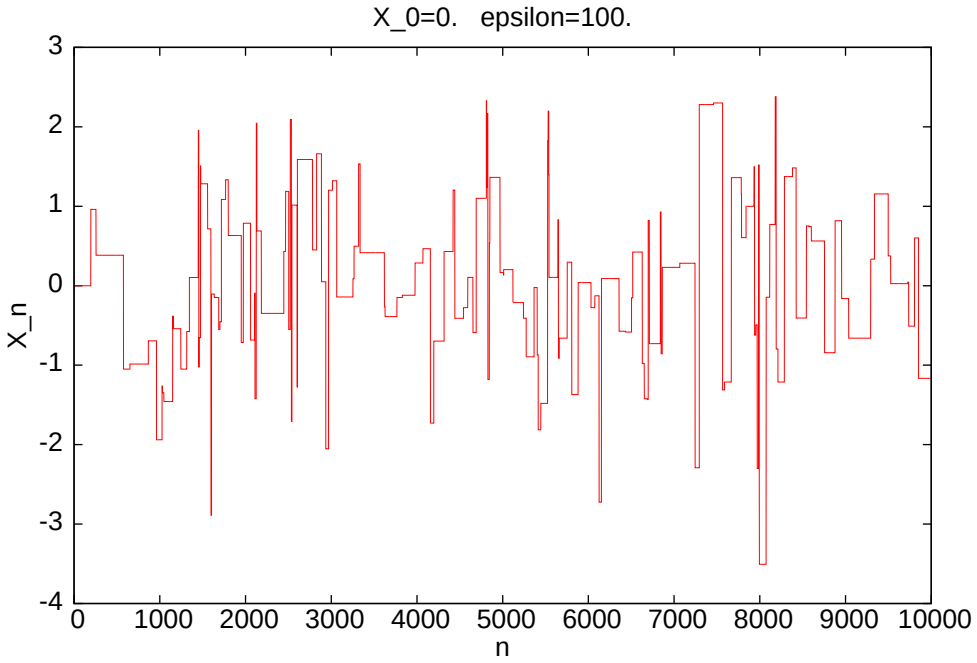

Figure 9.4: Metropolis Hasting for the normal distribution with $X_0 = 0$ and $\epsilon = 100$..

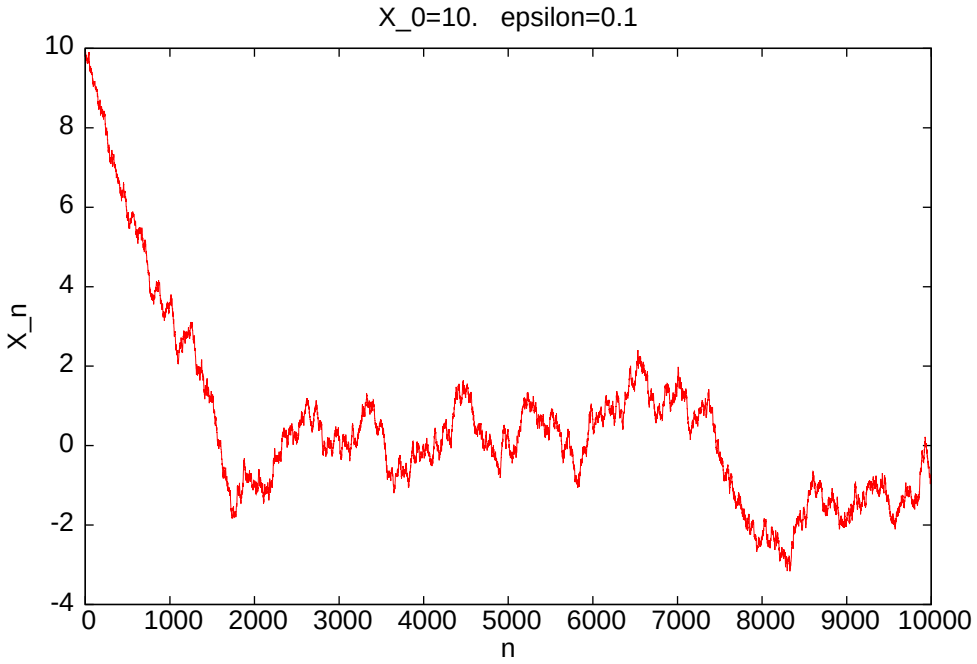

Figure 9.5:

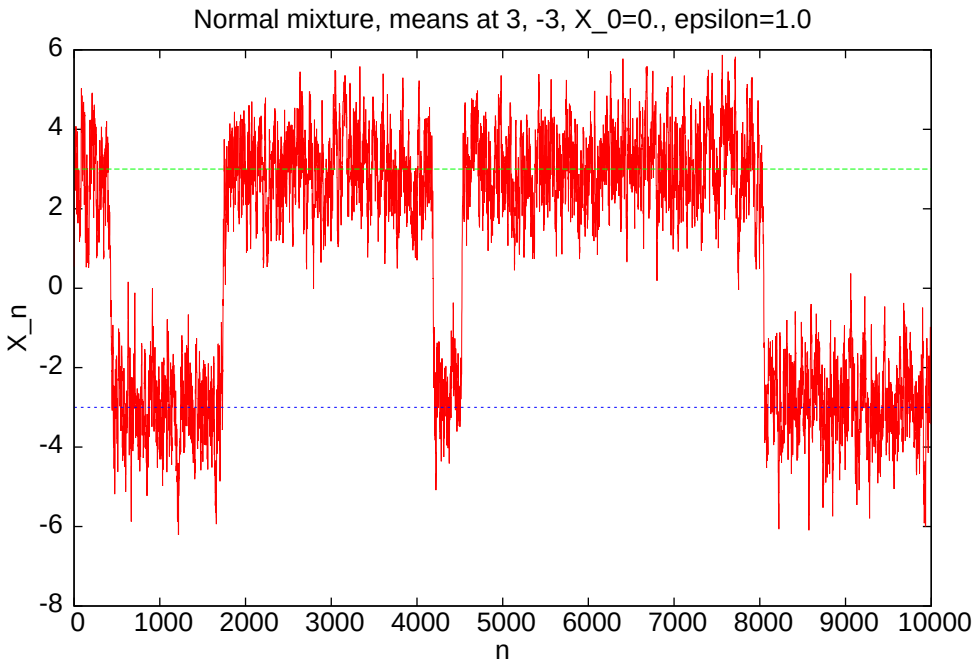
Normal mixture, means at 3, -3, X_0=0., epsilon=1.0
X_n
n
6
4
2
0
-2
-4
-6
-8
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000

Figure 9.6:

9.2 The variance for correlated samples

We recall a few probability facts. The covarance of random variable X and Y is

$$cov(X,Y) = E[XY] - E[X]E[Y] = E[(X-\mu_X)(Y-\mu_Y)] \tag{9.5}$$

This is a bi-linear form. Also note that $cov(X,X) = var(X)$. Thus for any random variables $Y_1, \cdots, Y_N$,

$$var(\sum_{i=1}^{N} Y_i) = \frac{1}{N^2} \sum_{i,j=1}^{N} cov(X_i, X_j) \tag{9.6}$$

A stochastic process X_n is said to be stationary if for all positive integers m and t, the joint distribution of $(X_{1+t}, X_{2+t}, \cdots, X_{m+t})$ is independent of t. In particular, $cov(X_i, X_j)$ will only depends on $|i-j|$. It is not hard to show that if we start a Markov chain in the stationary distribution, then we will get a stationary process. If the initial distribution is not the stationary distribution, then the chain will not be a stationary process. However, if the chain is irreducible, has a stationary distribution and is aperiodic, then the distribution of X_n will converges to that of the stationary distribution. So if we only look at the chain at long times it will be approximately a stationary process.

As before let

$$\hat{\mu} = \frac{1}{N} \sum_{n=1}^{N} f(X_n) \tag{9.7}$$

be our estimator for the mean of $f(X)$. Its variance is

$$Var(\hat{\mu}) = \frac{1}{N^2} \sum_{k,n=1}^{N} cov(f(X_k), f(X_n)) \tag{9.8}$$

In the following we do a bit of hand-waving and make some unjustifiable assumptions. If the initial state X_0 is chosen according to the stationary distribution (which is typically impossible in practice) then these assumptions are justified. But there are certainly situations where they are not. Let $\sigma^2(f)$ be the variance of $f(X)$ in the stationary distribution. If N is large, then for most terms in the sum k and n are large. So the distribution of X_k and X_n should be close to the stationary distribution. So the variance of $f(X_k)$ should be close to $\sigma(f)^2$. So

$$\begin{aligned} cov(f(X_k), f(X_n)) &= [var(f(X_k))var(f(X_n))]^{1/2} cor(f(X_k), f(X_n)) & (9.9) \\ &\approx \sigma^2(f)\, cor(f(X_k), f(X_n)) & (9.10) \end{aligned}$$

where $cor(Y,Z)$ is the correlation coefficient for Y and Z. Thus

$$Var(\hat{\mu}) = \frac{\sigma(f)^2}{N^2} \sum_{k=1}^{N} \sum_{n=1}^{N} cor(f(X_k), f(X_n)) \tag{9.11}$$

Fix a k and think of $cor(f(X_k), f(X_n))$ as a function of n. This function will usually decay as n moves away from k. So for most values of k we can make the approximation

$$\sum_{n=1}^{N} cor(f(X_k), f(X_n)) \approx \sum_{n=-\infty}^{\infty} cor(f(X_k), f(X_{k+n})) = 1 + 2\sum_{n=1}^{\infty} cor(f(X_k), f(X_{k+n})) \quad (9.12)$$

Finally we assume that this quantity is is essentially independent of k. This quantity is called the autocorrelation time of $f(X)$. We denote it by $\tau(f)$. So

$$\tau(f) = 1 + 2\sum_{n=1}^{\infty} cor(f(X_0), f(X_n)) \quad (9.13)$$

Explain why it is a "time" by looking at correlation that decays exponentially with a time scale. We now have

$$Var(\hat{\mu}) \approx \frac{\sigma(f)^2}{N^2} \sum_{k=1}^{N} \tau(f) = \frac{\sigma(f)^2 \tau(f)}{N} \quad (9.14)$$

If the samples were independent the variance would be $\sigma(f)^2/N$. So our result says that this variance for independent samples is increased by a factor of $\tau(f)$. Another way to interpret this result is that the "effective" number of samples is $N/\tau(f)$.

To use this result we need to compute $\tau(f)$. We can use the simulation run to do this. We approximate the infinite sum in the expression for $\tau(f)$ by truncating it at M. So we need to estimate

$$1 + 2\sum_{n=1}^{\infty} cor(f(X_k), f(X_{k+n})) \quad (9.15)$$

$$\hat{\tau(f)} = 1 + 2\frac{1}{N-M} \sum_{k=1}^{N-M} \sum_{n=1}^{M} cor(f(X_k), f(X_{k+n})) \quad (9.16)$$

Of course we do not typically have any a priori idea of how big M should be. So we need to first try to get an idea of how fast the correlation function decays.

MORE Help Help Help

Once we have an estimate for the variance of our estimator of $E[f(X)]$, we can find a confidence interval, i.e. error bars, in the usual way.

9.3 Variance via batched means

Esimating $\tau(f)$ can be tricky. In this section we give a quick and dirty method for putting error bars on our estimator that does not require computing the correlation time $\tau(f)$. Let N

be the number of MC time steps that we have run the simulation for. We pick an integer l and put the samples in batches of length l. Let $b = N/l$. So b is the number of batches. So the first batch is $X_1, \cdots, X_l$, the second batch is $X_{l+1}, \cdots, X_{2l}$, and so on. If l is large compared to the autocorrelation time of $f(X)$, then if we pick X_i and X_j from two different batches then for most choices of i and j, $f(X_i)$ and $f(X_j)$ will be almost independent. So if we form estimators for the batches, $j = 1, 2, \cdots, b$,

$$\hat{\mu}_j = \frac{1}{l} \sum_{i=1}^{b} f(X_{(j-1)l+i}) \tag{9.17}$$

then the $\hat{\mu}_1, \cdots, \hat{\mu}_b$ will be almost independent. Note that

$$\hat{\mu} = \frac{1}{b} \sum_{i=1}^{b} \hat{\mu}_i \tag{9.18}$$

So

$$var(\hat{\mu}) \approx \frac{1}{b^2} \sum_{i=1}^{b} var(\hat{\mu}_i) \tag{9.19}$$

The batches should have essentially the same distribution, so $var(\hat{\mu}_i)$ should be essentially independent of i. We estimate this common variance using the sample variance of $\hat{\mu}_i$, $i = 1, 2, \cdots, b$. Denote it by s_l^2. Note that this batch variance depends very much on the choice of l. Then our estimator for the variance of $\hat{\mu}$ is

$$var(\hat{\mu}) \approx \frac{s_l^2}{b} \tag{9.20}$$

How do we choose the batch size? The number of batches b is N/l. We need b to be reasonably large (say 100, certainly at least 10) since we estimate the variance of the batches using a sample of size b. So l should be at most 1/10 of N. We also need l to be large compared to the autocorrelation time. If N is large enough, there will be a range of l where these two criteria are both met. So for l in this range the estimates we get for the variance of $\hat{\mu}$ will be essentially the same. So in practice we compute the above estimate of $var(\hat{\mu})$ for a range of l and look for a range over which it is constant. If there is no such range, then we shouldn't use batched means. If this happens we should be suspicious of the MC simulation itself since this indicates the number of MC time steps is not a large multiple of the autocorrelation time.

Stop - Wed, 4/6

We return to our hand-waving argument that the batch means should be almost independent if the batch size is large compared to the autocorrelation time and make this argument more quantitative.

The variance of $\hat{\mu}$ is given exactly by

$$var(\hat{\mu}) = \frac{1}{N^2} \sum_{i,j=1}^{N} cov(f(X_i), f(X_j)) \tag{9.21}$$

Using batched means we approximate this by

$$\frac{1}{b^2} \sum_{i=1}^{b} var(\hat{\mu}_i) \tag{9.22}$$

This is equal to

$$\frac{1}{N^2} \sum_{(i,j) \in S} cov(f(X_i), f(X_j)) \tag{9.23}$$

where S is the set of pairs (i, j) such that i and j are in the same batch. So the difference between the true variance of $\hat{\mu}$ and what we get using batched means is

$$\frac{1}{N^2} \sum_{(i,j) \notin S} cov(f(X_i), f(X_j)) \tag{9.24}$$

Keep in mind that the variance of $\hat{\mu}$ is of order $1/N$. So showing this error is small means showing it is small compared to $1/N$. We fix the number b of batches, let $N = bl$ and let $l \to \infty$. We will show that in this limit the error times N goes to zero. So we consider

$$\frac{1}{N} \sum_{(i,j) \notin S} cov(f(X_i), f(X_j)) \tag{9.25}$$

Define $C()$ by $C(|i - j|) = cov(f(X_i), f(X_j))$. Then the above can be bounded by

$$\frac{2b}{N} \sum_{k=1}^{l} \sum_{i=1}^{\infty} |C(i + k)| \tag{9.26}$$

Let

$$T(k) = \sum_{i=1}^{\infty} |C(i + k)| = \sum_{i=k+1}^{\infty} |C(i)| \tag{9.27}$$

We assume that $C(i)$ is absolutely summable. So $T(k)$ goes to zero as $k \to \infty$. Using the fact that $l = N/b$, we have that the above equals

$$\frac{2}{l} \sum_{k=1}^{l} T(k) \tag{9.28}$$

which goes to zero by the analysis fact that if $\lim_{k \to \infty} a_k = 0$ then $\lim_{n \to \infty} \frac{1}{n} \sum_{k=1}^{n} a_k = 0$.

9.4 Subsampling

Until now we have always estimated the mean of $f(X)$ with the estimator

$$\hat{\mu} = \frac{1}{N} \sum_{i=1}^{N} f(X_i) \tag{9.29}$$

Subsampling (sometimes called lagging) means that we estimate the mean with

$$\hat{\mu}_{sub} = \frac{1}{N/l} \sum_{i=1}^{N/l} f(X_{il}) \tag{9.30}$$

where l is the subsampling time. In other words we only evaluate f every l time steps.

One reason to do this is that if l is relatively large compared to the autocorrelation time, then the X_{il} will be essentially independent and we can estimate the variance of our estimator just as we did for direct Monte Carlo where the samples where independent. So the variance of the subsampling estimator will be approximately $\sigma^2/(N/l)$, where σ^2 is the variance of $f(X)$.

Intuitively one might expect if we compare two estimators, one with subsampling and one without, for the same number of MC steps, then the estimator without subsampling will do better. More precisely, we expect the variance of $\hat{\mu}$ to be smaller than the variance of $\hat{\mu}_{sub}$ if we use the same N in both. The following proposition makes this rigorous.

Proposition 1 *Define $\hat{\mu}$ as above and define*

$$\hat{\mu}_j = \frac{1}{N/l} \sum_{i=0}^{N/l-1} f(X_{il+j}) \tag{9.31}$$

Letting $\sigma(?)$ denote the standard deviation of ?, we have

$$\sigma(\hat{\mu}) \leq \frac{1}{l} \sum_{j=1}^{l} \sigma(\hat{\mu}_j) \tag{9.32}$$

Typically the variances of the $\hat{\mu_j}$ will be essentially the same and so morally the proposition says that the variance of $\hat{\mu}$ is no larger than the variance of $\hat{\mu}_{sub}$.

Proof: The crucial fact used in the proof is that for any RV's X, Y,

$$|cov(X,Y)| \leq [var(X)\, var(Y)]^{1/2} \tag{9.33}$$

which follows from the Cauchy Schwarz inequality. Note that

$$\hat{\mu} = \frac{1}{l} \sum_{j=1}^{l} \hat{\mu}_j \tag{9.34}$$

So

$$\begin{aligned} var(\hat{\mu}) &= \frac{1}{l^2} \sum_{i=1}^{l} \sum_{j=1}^{l} cov(\hat{\mu}_i, \hat{\mu}_j) && (9.35) \\ &\leq \frac{1}{l^2} \sum_{i=1}^{l} \sum_{j=1}^{l} [var(\hat{\mu}_i) var(\hat{\mu}_j)]^{1/2} && (9.36) \\ &= \frac{1}{l^2} \sum_{i=1}^{l} \sum_{j=1}^{l} \sigma(\hat{\mu}_i) \sigma(\hat{\mu}_j) && (9.37) \\ &= \left[\frac{1}{l} \sum_{i=1}^{l} \sigma(\hat{\mu}_i) \right]^2 && (9.38) \end{aligned}$$

Taking square roots the result follows. **QED**.

The proposition does not imply that it is never beneficial to subsample. Computing f for every time step in the Markov chain takes more CPU time than subsampling with the same number of time steps. So there is a subtle trade-off between the increase in speed as we subsample less frequently and the increase in the variance. Choosing l so large that it is much larger than the autocorrelation time is not necessarily the optimal thing to do. Ideally we would like to choose l to minimize the error we get with a fixed amount of CPU time. Finding this optimal choice of l is not trivial.

Factor of two hueristic: Actually finding the optimal amount choice of l is non-trivial. We give a crude method that at worst will require twice as much CPU time as the optimal choice of l will. Let τ_f be time to compute $f(X_i)$ and let τ_X be the time to perform one time step, i.e., compute X_{i+1} given X_i. We take l to be τ_f/τ_X (rounded to an integer). Note that this will make the time spent on computing the X_i equal to the time spent evaluating f on the X_i. Now we argue that this is worse than by the optimal choice by at most a factor of two in CPU time.

Let l_{opt} be the optimal choice of l. First consider the case that $l < l_{opt}$, i.e., we are evaluating f more often than in the optimal case. Now compare a simulation with N time steps with our crude choice of l with a simulation with N time steps with the optimal choice l_{opt}. They have the same number of MC steps, but the simulation using l is sampled more often and so is at least as accurate as the simulation using l_{opt}. The simulation using l takes more CPU time, but at most the extra time is the time spent evaluating f and this is half of the total CPU time. So the simulation using l takes at most twice as long as the one using l_{opt}.

Now consider the case that $l > l_{opt}$. Now we compare two simulations that each evaluate f a total of M times. Since the evaluations using l are more widely spaced in time, they will be less correlated than those for the simulation using l_{opt}. So the simulation using l will be at least as accurate as the simulation using l_{opt}. The two simulations evaluate f the same number of times. The simulation using l requires more MC steps. But the total time it spends computing the X_i is equal to the total time it spends evaluating f. So the total CPU is twice the time spent evaluating f. But the time spent evaluating f is the same for the two simulations, and so is less that the total time the simulation using l uses.

9.5 Burn-in or initialization

In this section we consider the error resulting from the fact that we start the chain in an arbitrary state X_0 which may be atypical is some sense. We need to run the chain for some number T of time steps and discard those times steps, i.e., we estimate the mean using

$$\frac{1}{N-T} \sum_{i=T+1}^{N} f(X_i) \tag{9.39}$$

This goes under a variety of names: initilization, burn-in, convergence to stationarity, stationarization, thermalization.

We start with a trivial, but sometimes very relevant comment. Suppose we have a direct MC algorithm that can generate a sample from the distribution we are trying to simulate, but it is really slow. As long as it is possible to generate one sample in a not unreasonable amount of time, we can use this algorithm to generate the inital state X_0. Even if the algorithm that directly samples from π takes a thousand times as much time as one step for the MCMC algorithm, it may still be useful as a way to initialize the MCMC algorithm. When we can do this we eliminate the burn-in or initialization issue altogether.

In the example of burn-in at the start of this chapter, $f(X)$ was just X. When we started the chain in $X_0 = 10$, this can be thought of as starting the chain in a state where the value of $f(X)$ is atypical. However, we should emphasize that starting in a state with a typical value of $f(X)$ does not mean we will not need a burn-in period. There can be atypical X for which $f(X)$ is typical as the following example shows.

Example: We want to simulate a nearest neighbor, symmetric random walk in one dimension with n steps. This is easily done by direct Monte Carlo. Instead we consider the following MCMC. This is a 1d version of the pivot algorithm. This 1d RW can be thought of as a sequence of n steps which we will denote by $+1$ and -1. So a state $(s_1, s_2, \cdots, s_n)$ is a string of n $+1$'s and -1's. Let $f(s_1, \cdots, s_n) = \sum_{i=1}^{n} s_i$, i.e., the terminal point of the random

walk. The probability measure we want is the uniform measure - each state has probability $1/2^n$. The pivot algorithmn is as follows. Pick an integer j from $0, 1, 2, \cdots, n-1$ with the uniform distribution. Leave s_i unchanged for $i \leq j$ and replace s_i by $-s_i$ for $i > j$. Show this statisfies detailed balance.

In the following plots the number of steps in the walk is always $L = 1,000,000$. All simulations are run for 10,000 time steps. The initial state and the RV we look at varies. Note that all state have probability 2^{-n}, so no state is atypical in the sense of having unusually small probability.

In the first plot, figure 9.7, we start with the state that is $n/2$ +'s, followed by $n/2$ −'s. The RV plotted is the distance of the endpoint to the origin. The signed distance to the endpoint is approximately normal with mean zero and variance L. So the range of this RV is roughly $[0, 2\sqrt{L}]$. In the initial state the RV is 0. This is a typical value for this RV. There is a significant burn-in period.

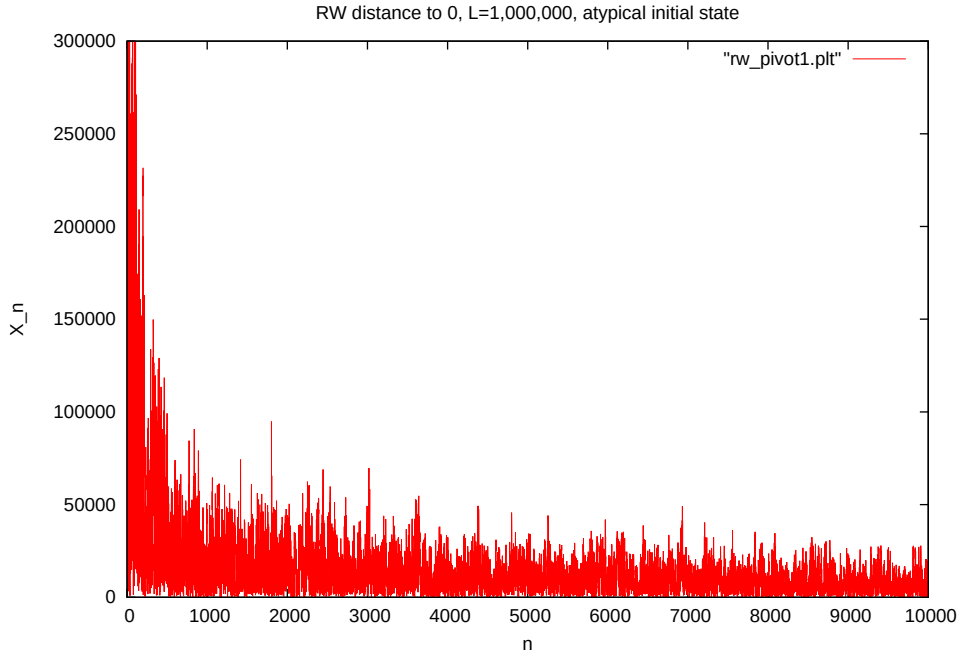

Figure 9.7: MCMC simulation of random walk with $L = 1,000,000$ steps. Random variable plotted is distance from endpoint of walk to 0. Initial state is a walk that goes right for $500,000$ steps, then left for $500,000$ steps. MC is run for 10,000 time steps. There is a significant burn-in period.

In the second plot, figure 9.8, the initial state is a random walk generated by just running a direct MC. The RV is still the distance from the endpoint to the origin. Note the difference in vertical scale between this figure and the preceding figure. This MCMC seems to be working

well.

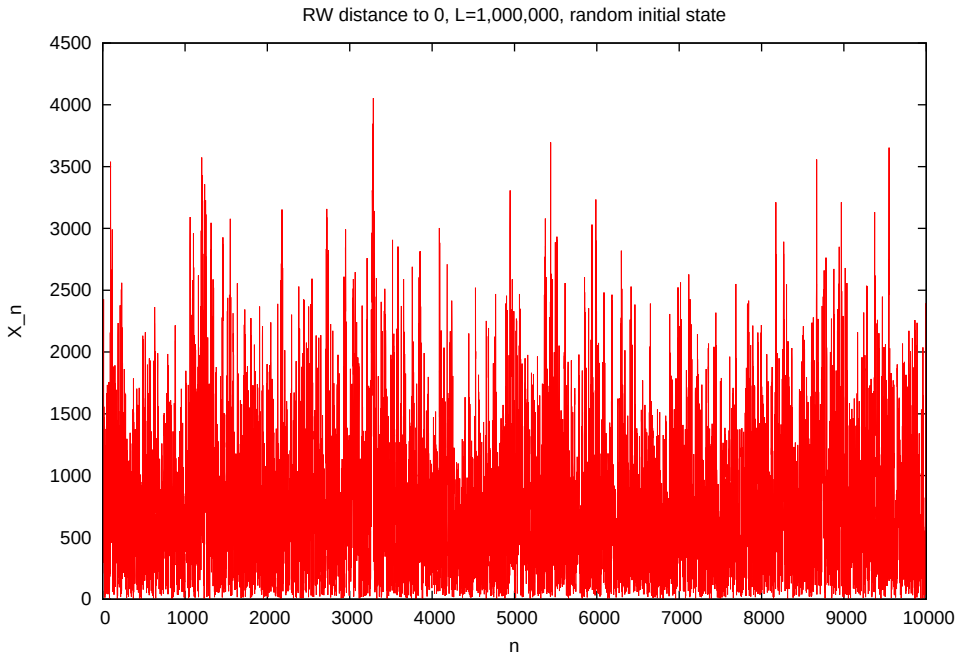

Figure 9.8: MCMC simulation of random walk with $L = 1,000,000$ steps. Random variable plotted is distance from endpoint of walk to 0. Initial state is a random walk generated by direct sampling. MC is run for 10,000 time steps.

In the third plot, figure 9.9, the random variable is the distance the walk travels over the time interval $[L/2, L/2 + 10,000]$. The initial state is again a random walk generated by just running a direct MC. For this RV the autocorrelation time is large.

There are two lessons to be learned from this example. One is the subtlety of initilization. The other is that there can be very different time scales in our MCMC. The autocorrelation time of the RV that is the distance to the origin of the endpoint appears to be not very large. By contrast, the autocorrelation time for the other RV is rather large. We might expect that the exponential decay time for $cov(f(X_i), f(X_j))$ for the first RV will be not too large while this time for the second RV will be large. However, it is quitely like that for the first RV there is some small coupling to this "slow mode." So even for the first RV the true exponential decay time for the covariance may actually be quite large. Note that there are even slower modes in this system. For example, consider the RV that is just the distance travelled over the time interal $[L/2, L/2 + 2]$.

A lot of theoretical discussions of burn-in go as follows. Suppose we start the chain in some

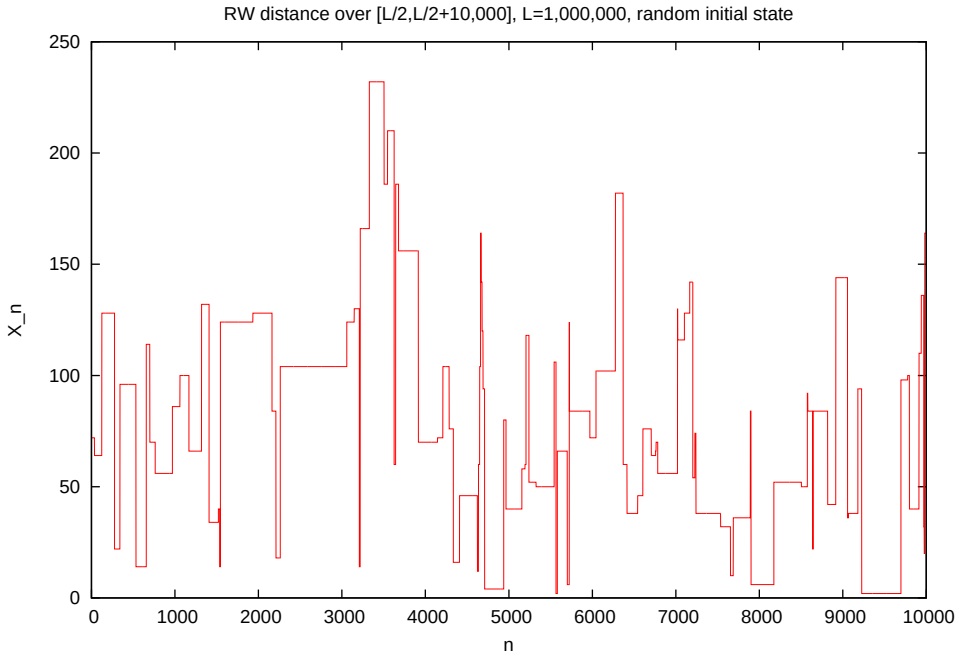

Figure 9.9: MCMC simulation of random walk with $L = 1,000,000$ steps. Random variable plotted is distance walk travels over the time interval $[L/2, L/2 + 10,000]$. Initial state is a random walk generated by direct sampling.

distribution π_0 and let π_n be the distribution of X_n. Suppose we have a bound of the form

$$||\pi_n - \pi||_{TV} \leq C \exp(-n/\tau) \tag{9.40}$$

Then we can choose the number of samples T to discard by solving for T in $C \exp(-T/\tau) \leq \epsilon$ where ϵ is some small number. The problem is than we rarely have a bound of the above form, and in the rare cases when we do, the τ may be far from optimal. So we need a practical test for convergence to stationarity.

Stop - Wed, 4/6

If the distribution of the initial state X_0 is the stationary (so the chain is stationary), then the two samples

$$f(X_1), f(X_2), \cdots, f(X_N), \quad \textit{and} \quad f(X_{N+1}), f(X_{N+2}), \cdots, f(X_{2N}) \tag{9.41}$$

have the same distribution.

To use this observation to test for converges to stationarity, we let T be the number of samples we will discard for burn-in purposes and compare the two samples

$$f(X_{T+1}), f(X_{T+2}), \cdots, f(X_{T+N}), \quad \textit{and} \quad f(X_{T+N+1}), f(X_{T+N+2}), \cdots, f(X_{T+2N}) \tag{9.42}$$

A crude graphical test is just to plot histograms of these two samples and see if they are obviously different.

For a quantitative test we can use the Kolmogorov-Smirnov statistic. This is a slightly different version of KS from the one we saw before when we tested if a sample came from a specific distribution with a given CDF F. We first forget about the Markov chain and consider a simpler situation. Let $X_1, X_2, \cdots X_{2N}$ be i.i.d. We think of this as two samples: $X_1, X_2, \cdots X_N$ and $X_{N+1}, X_{N+2}, \cdots X_{2N}$. We form their empirical CDF's:

$$F_1(x) = \frac{1}{N} \sum_{i=1}^{N} 1_{X_i \leq x}, \tag{9.43}$$

$$F_2(x) = \frac{1}{N} \sum_{i=N+1}^{2N} 1_{X_i \leq x} \tag{9.44}$$

Then we let

$$K = \sup_x |F_1(x) - F_2(x)| \tag{9.45}$$

Note that K is a random variable (a statistic). As $N \to \infty$, the distribution of $\sqrt{N}K$ converges. The limiting distribution has CDF

$$R(x) = 1 - \sum_{k=1}^{\infty} (-1)^{k-1} \exp(-2k^2 x^2) \tag{9.46}$$

This sum converges quickly and so $R(x)$ is easily computed numerically. It is useful to know the 95% cutoff. We have $R(1.36) = 0.95$. So for large N, $P(\sqrt{N}K \leq 1.36) \approx 0.95$.

Now return to MCMC. We cannot directly use the above since the samples of our Markov chain are correlated. So we must subsample. Let l be large compared to the autocorrelation time for f, i.e., $\tau(f)$. Then we compare the two samples

$$f(X_{T+l}), f(X_{T+2l}), \cdots, f(X_{T+Nl}), \quad \textit{and} \quad f(X_{T+Nl+l}), f(X_{T+Nl+2l}, \cdots, f(X_{T+2Nl}) \tag{9.47}$$

Finally, we end this section by noting that it never hurts (much) to throw away the first 10% of your simulation run.

9.6 Autocorrelation times and related times

This section is based in large part on the article "The Pivot Algorithm: A Highly Efficient Monte Carlo Method for the Self-Avoiding Walk" by Madras and Sokal, Journal of Statistical Physics, **50**, 109-186 (1988), sections 2.2.

We assume the Markov chain is discrete with a finite state space. Not all of the following statement extend to the case of continuous state space or infinite discrete state space. We also assume our chain is irrecducible and aperiodic.

In this section we study three different times that can be defined for an MCMC and how they are related. We have already seen one autocorrelation time defined by (9.13). In this section we will refer to this as an "integrated autocorrelation time" and write it as $\tau_{int,f}$. We recall the formula for it:

$$\tau_{int,f} = 1 + 2\sum_{n=1}^{\infty} cor(f(X_0), f(X_n)) \tag{9.48}$$

This time is important since it enters our formula for the variance of our estimator $\hat{\mu}$. That variance is given approximatly by $\sigma(f)^2\tau_{int,f}/N$. Since the variance for direct Monte Carlo is $\sigma(f)^2/N$, we can interpret this formula as saying that in an MCMC the number of effectively independent samples is $N/\tau(f)$. (The definition of $\tau_{int,f}$ in Madras and Sokal differs by a factor of 2.) In the definition of $\tau_{int,f}$ we assume that X_0 is distributed according to the stationary distribution. So the Markov chain is stationary. (We can imagine that we have run the chain for a sufficiently long burn-in period to achieve this. In that case X_0 is the state after the burn-in period.)

The second time we will consider is closely related. We follow the terminology of Madras and Sokal. For a function f on the state space we define the unnormalized autocorrelation function of f to be

$$C_f(t) = E[f(X_s)f(X_{s+t})] - \mu_f^2, \quad where \tag{9.49}$$

$$\mu_f = E[f(X_s)] \tag{9.50}$$

Note that $C_f(0)$ is the variance of $f(X_t)$. We define the normalized autocorrelation function to be

$$\rho_f(t) = \frac{C_f(t)}{C_f(0)} \tag{9.51}$$

With our assumptions on the Markov chain $\rho_f(t)$ will decay exponentially with t. We define the *exponential autocorrelation time* for f by

$$\tau_{exp,f} = \limsup_{t\to\infty} \frac{t}{-log|\rho_f(t)|} \tag{9.52}$$

We define an exponential autocorrelation time for the entire chain by

$$\tau_{exp} = \sup_f \tau_{exp,f} \tag{9.53}$$

With our assumptions on the Markov chain, τ_f will be finite, but it can be infinite when the state space is finite.

The two times we have considered so far are properties of the stationary Markov chain. The third time characterizes how long it takes the Markov chain to converge to the stationary distribution starting from an arbitrary initial condition. With our assumptions on the chain this convergence will be exponentially fast. We define τ_{conv} to be the slowest convergence rate we see when we consider all initial distributions. More precisely we define

$$\tau_{conv} = \sup_{\pi_0} \limsup_{t \to \infty} \frac{t}{-\log(||\pi_t - \pi||_{TV})} \tag{9.54}$$

Here π_0 is the distribution of X_0, π_t is the distribution of X_t and π is the stationary distribution.

The following two propositions almost say that $\tau_{exp} = \tau_{conv}$.

Proposition 2 *Suppose there are constants c and τ such that for all initial π_0,*

$$||\pi_t - \pi||_{TV} \leq ce^{-t/\tau} \tag{9.55}$$

Then $\tau_{exp} \leq \tau$.

Remark: This almost says $\tau_{exp} \leq \tau_{conv}$. It does not exactly say this since our definition of τ_{conv} does not quite imply the bound (9.55) holds with $\tau = \tau_{conv}$.

Proof: If we apply the hypothesis to the initial condition where π_0 is concentrated on the single state x, we see that

$$||p^t(x, \cdot) - \pi(\cdot)||_1 \leq ce^{-t/\tau} \tag{9.56}$$

Now let f be a function on the state space with zero mean in the stationary distribution. Then using $\sum_y f(y)\pi(y) = 0$, we have

$$\begin{aligned}
|E[f(X_0)f(X_t)]| &= |\sum_{x,y} f(x)\pi(x)p^t(x,y)f(y)| && (9.57)\\
&= |\sum_{x,y} f(x)\pi(x)f(y)[p^t(x,y) - \pi(y)]| && (9.58)\\
&\leq \sum_x |f(x)|\pi(x)||f||_\infty ||p^t(x,\cdot) - \pi(\cdot)||_1 && (9.59)\\
&\leq \sum_x |f(x)|\pi(x)||f||_\infty ce^{-t/\tau} && (9.60)
\end{aligned}$$

The propostion follows. **QED**.

Proposition 3 *Suppose there are constants c and τ such that*

$$Cov(g(X_0), f(X_t)) \leq ce^{-t/\tau}||f||_\infty||g||_\infty \tag{9.61}$$

for all functions f and g on the state space. Then $\tau_{conv} \leq \tau$.

Remark: This almost says $\tau_{conv} \leq \tau_{exp}$. It does not exactly say this since our definition of τ_{exp} does not quite imply the bound (9.61) holds with $\tau = \tau_{exp}$.

Proof: The total variation norm can be computed by

$$||\pi_t - \pi||_{TV} \sup_{f:||f||_\infty \leq 1} \sum_x f(x)[\pi_t(x) - \pi(x)] \tag{9.62}$$

Let $g(x) = \pi_0(x)/\pi(x)$. (Note that $\pi(x) > 0$ for all x.) The expected value of g is the stationary distribution is

$$Eg(X_t) = \sum_x g(x)\pi(x) = \sum_x \pi_0(x) = 1 \tag{9.63}$$

So

$$\begin{aligned} Cov(g(X_0), f(X_t)) &= \sum_{x,y} g(x)f(y)\pi(x)p^t(x,y) - E[g(X_0)]E[f(X_t)] && (9.64)\\ &= \sum_{x,y} \pi_0(x)f(y)p^t(x,y) - E[f(X_t)] && (9.65)\\ &= \sum_y \pi_t(y)f(y) - \sum_x \pi(x)f(x) && (9.66) \end{aligned}$$

Since this is bounded by $ce^{-t/\tau}||f||_\infty||g||_\infty = ce^{-t/\tau}||g||_\infty$, the proposition follows. **QED**.

Recall that the stationary distribution is a left eigenvector of the transition matrix $p(x,y)$, and the constant vector is a right eigenvector. (Both have eigenvalue 1.) In general $p(x,y)$ is not symmetric, so there need not be a complete set of left or right eigenvectors. However, if the chain satisfies detailed balance then there is, as we now show. We rewrite the detailed balance condition as

$$\pi(x)^{1/2}p(x,y)\pi(y)^{-1/2} = \pi(y)^{1/2}p(x,y)\pi(y)^{-1/2} \tag{9.67}$$

So if we define

$$\hat{p}(x,y) = \pi(x)^{1/2}p(x,y)\pi(y)^{-1/2} \tag{9.68}$$

then $\hat{p}$ is symmetric. (p is self-adjoint on $l^2(\pi)$.) If we let S be the linear operator on functions on the state space that is just multiplication by $\pi^{1/2}$, then

$$\hat{p} = SpS^{-1} \tag{9.69}$$

Let $\hat{e}_k$ be a complete set of eigenvectors for it. So $\hat{p}\hat{e}_k = \lambda_k \hat{e}_k$. Note that the $\hat{e}_k$ are orthonormal.

Now let $f(x)$ be a function on the state space. It suffices to consider functions which have zero mean in the stationary distribution. So the covariance is just $E[f(X_0)f(X_t)]$. To compute this expected value we need the joint distribution of X_0, X_t. It is $\pi(x_0)p^t(x_0, x_t)$. So

$$E[f(X_0)f(X_t)] = \sum_{x_0,x_t} \pi(x_0)p^t(x_0,x_t)f(x_0)f(x_t) = \sum_{x_0} f(x_0)\pi(x_0) \sum_{x_t} p^t(x_0,x_t)f(x_t) \tag{9.70}$$

We can write this as

$$(f\pi, p^t f) = (f\pi, (S^{-1}\hat{p}S)^t f) = (f\pi, S^{-1}\hat{p}^t S f) = (S^{-1}\pi f, \hat{p}^t S f) = (f\pi^{1/2}, \hat{p}^t \pi^{1/2} f) \tag{9.71}$$

Using the spectral decompostion of $\hat{p}$ this is

$$\sum_k (f\pi^{1/2}, \hat{e}_k)\lambda_k^t (\hat{e}_k, \pi^{1/2} f) = \sum_k c_k^2 \lambda_k^t \tag{9.72}$$

with $c_k = (f\pi^{1/2}, \hat{e}_k)$.

The right eigenvector of p with eigenvalue 1 is just the constant vector. So $\pi^{1/2}$ is the eigenvector of $\hat{p}$ with eigenvalue 1. So

$$c_1 = \sum_x \pi(x)^{1/2}\pi(x)^{1/2} f(x) = 0 \tag{9.73}$$

since the mean of f in the stationary distribution is zero. Let λ be the maximum of the absolute values of the eigenvalues not equal to 1. So for $k \geq 2$, $|\lambda_k^t| \leq \lambda$. So

$$E[f(X_0)f(X_t)] \leq \lambda^t \sum_{k \geq 2} c_k^2 \tag{9.74}$$

Note that $var(f(X_0)) = \sum_k c_k^2$. So

$$\tau_{int,f} = 1 + 2\sum_{t=1}^{\infty} cor(f(X_0), f(X_t)) \leq 1 + 2\sum_{t=1}^{\infty} \lambda^t = 1 + 2\frac{\lambda}{1-\lambda} = \frac{1+\lambda}{1-\lambda} \leq = \frac{2}{1-\lambda} \tag{9.75}$$

Stop - Mon, 4/11

Proposition 4 *Assume that p is diagonalizable. Let λ be the maximum of the absolute values of the eigenvalues not equal to 1. Then τ_{conv} is given by $\lambda = \exp(-1/\tau_{conv})$.*

Proof: Let S be an invertible matrix that diagonalizes p. So $p = SDS^{-1}$ where D is diagonal with entries λ_k. We order things so that the first eigenvalue is 1. We have $\pi p = \pi$. So $\pi SD = \pi S$. So πS is an eigenvector of D with eigenvalue 1. So it must be $(1, 0, \cdots, 0)$. So $\pi = (1, 0, \cdots, 0)S^{-1}$. This says that the first row of S^{-1} is π. If we let $\vec{1}$ be the column vector $(1, 1, \cdots)^T$, then we know $p\vec{1} = \vec{1}$. So $DS^{-1}\vec{1} = S^{-1}\vec{1}$. This says $S^{-1}\vec{1}$ is the eigenvector of D with eigenvalue 1, so it is $(1, 0, \cdots, 0)^T$. So $\vec{1} = S(1, 0, \cdots, 0)^T$. We now have

$$\begin{aligned} \pi_t &= \pi_0 p^t = \pi_0 (SDS^{-1})^t = \pi_0 S D^t S^{-1} = \pi_0 S diag(\lambda_1^t, \cdots, \lambda^t) S^{-1} && (9.76) \\ &= \pi_0 S diag(1, 0, 0 \cdots, 0) S^{-1} + \pi_0 S diag(0, \lambda_2^t, \cdots, \lambda^t) S^{-1} && (9.77) \end{aligned}$$

Since $\vec{1} = S(1, 0, \cdots, 0)^T$, $Sdiag(1, 0, \cdots)$ is the matrix with 1's in the first column and 0's elsewhere. So $\pi_0 Sdiag(1, 0, 0 \cdots, 0)$ is just $(1, 0, \cdots, 0)^T$ Thus $\pi_0 Sdiag(1, 0, 0 \cdots, 0)S^{-1} = \pi$. The second term in the above goes to zero like λ^t. **QED**

Now consider the bound $\tau_{int,f} \leq \frac{2}{1-\lambda}$ which we derived when the chain satistifed detailed balance. Using the previous proposition, this becomes

$$\tau_{int,f} \leq \frac{2}{1 - \exp(-1/\tau_{conv})} \tag{9.78}$$

If τ_{conv} is large, then the right side is approximately $2\tau_{conv}$.

Madras and Sokal carry out a detailed analysis for the pivot algorithm for a random walk in two dimensions. Then show that τ_{exp} is $O(N)$. For "global" random variables such as the distance from the origin to the end of the walk, $\tau_{f,int}$ is $O(\log(N))$. But for very local observables, such as the angle between adjacent steps on the walk, $\tau_{f,int}$ is $O(N)$.

9.7 Missing mass or bottlenecks

The problem we briefly consider in this section is an MCMC in which the state space can be partitioned into two (or more) subsets $S = S_1 \cup S_2$ such that although the probabilities of getting from S_1 to S_2 and from S_2 to S_1 are not zero, they are very small.

The worst scenario is that we start the chain in a state in one of the subsets, say S_1, and it never leaves that subset. Then our long time sample only samples S_1. Without any apriori knowledge of the distribution we want to simulate, there is really no way to know if we are

failing to sample a representative portion of the distribution. "You only know what you have seen."

The slightly less worse scenario is that the chain does occasionally make transitions between S_1 and S_2. In this case there is no missing mass, but the autocorrelation time will be huge. In particular it will be impossible to accurately estimate the probabilities of S_1 and S_2.

It is worth looking carefully at the argument that your MCMC is irreducible to see if you can see some possible "bottlenecks".

If a direct MC is possible, although slow, we can do a preliminary direct MC simulation to get a rough idea of where the mass is.

Missing mass is not just the problem that you fail completely to visit parts of the state space, there is also the problem that you do eventually visit all of the state space, but it takes a long time to get from some modes to other modes. One way to test for this is to run multiple MCMC simulations with different initial conditions.

In the worst scenario when the chain is stuck in one of the two subsets, it may appear that there is a relatively short autocorrelation time and the burn-in time is reasonable. So our estimate of the autocorrelation time or the burn-in time will not indicate any problem. However, when the chain does occassionally make transitions between modes, we will see a very long autocorrelation time and a very slow convergence to stationary. So our tests for these sources of errors will hopefully alert us to the bottleneck.