Chapter 11
Markov chain Monte Carlo II

The next question question we need to discuss is whether the Markov chain indeed converges to the desired stationary distribution and how fast the convergence is. The detailed answers to these questions are technical and beyond the scope of this book. Nevertheless we attempt to give an intuitive answers here.

Irreducibility. It turns out that even a Markov chain has $\pi(m)$ as its stationary distribution, it may fail to converge to stationarity. That is, the stationary distribution may not be unique or the chain may never get from one state to another. To avoid this issue, the chain is required to be irreducible[1] [21, 24]. In other words, any state has positive probability of eventually being reached from an arbitrary state. For our setting in this chapter where $\mathbb{X} = \mathbb{R}^n$, we need to mild condidtions for the Markov chain from Algorithm 3 to be irreducible: 1) the desired density $\pi(m)$ is finite everywhere; and 2) the proposal $q(\cdot,\cdot)$ is positive and continuous.

Aperiodicity. The fact that even irreducible chain may not converge in distribution is due to periodicity problem, i.e., the chain may oscillate between states and hence is not convergent. Fortunately, for our setting the chain is automatically aperiodic [24].

Theorem 11.1. *If the Markov chain has $\pi(m)$ as its stationary distribution, and it is both irreducible and aperiodic, then for* almost everywhere *$m \in \mathbb{X}$[2], then the Markov chain is eventually distributed as $\pi(m)$, i.e.,*

$$\lim_{N\to\infty} P^N(m,A) = \mu(A), \quad \forall A \in \mathscr{S},$$

where $\mu(A)$ is the distribution function (the law), and $P^N(m,A) \overset{def}{=} \mathbb{P}[m_N \in A | m_0 = m]$ denotes N-step transition law of the Markov chain. Furthermore, the LLN holds:

$$\lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N} g(m_i) = \int_{\mathbb{X}} g(m)\, d\pi(m),$$

[1] In fact, only ϕ-irreducibility, a weaker condition, is needed.

[2] The correct requirement for $\mathbb{X}$ is that it has countably generated σ-algebra [24].

for all g such that $\int_{\mathbb{X}} |g(m)|\, d\pi(m) < \infty$.

Theorem 11.1 however has a little caveat, that is, it is true for almost every $m \in \mathbb{X}$ except for the zero-probability exceptional set. For example, if we are unlucky to start the chain in this exceptional set, the chain does not converge! To overcome this issue, the chain needs to be *Harris recurrent*, which means that $\forall A : \mu(A) > 0$, and for all m, the chain eventually reaches A from m with probability 1. For our setting, if the proposal $q(m,\cdot)$ is absolutely continuous with respect to $\pi(\cdot)$, i.e.,

$$q(m,A) = \int_A d\pi(x), \quad \forall A \in \mathscr{S},$$

then the Markov chain is *Harris recurrent*.

Up to this point, we know that under mild conditions, the Markov chain from the above Metropolis-Hastings algorithm converges, but we haven't discussed the convergence rate. The reason is that it is much more technical involving the notion of *ergodicity*. The readers are referred to [21, 24] for the details.

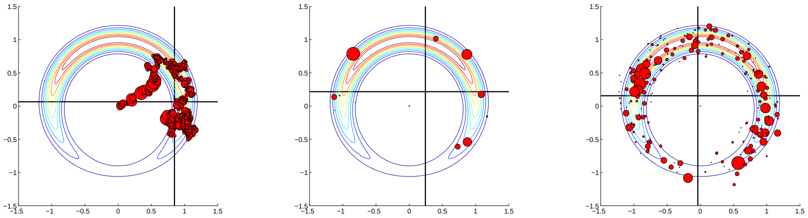
As you can see the Metropolis-Hastings algorithm is simple and elegant, but provides us a reversible transition kernel, which is exactly what we are looking for. The keys behind this are Steps 3 and 4 in Algorithm 3, known as Metropolized steps. At this point, we should be able to implement the algorithm except for one small detail: what should we choose for the proposal density? Let us choose the following Gaussian kernel

$$q(m,p) = \frac{1}{\sqrt{2\pi\gamma^2}} \exp\left(-\frac{1}{2\gamma^2}\|m-p\|^2\right),$$

which is the most popular choice. Metropolis-Hastings algorithm with above isotropic Gaussian proposal is known as *Random Walk Metropolis-Hastings* (RWMH) algorithm. For this particular method, the acceptance probability is very simple, i.e.,

$$\alpha = \min\left\{1, \frac{\pi(p)}{\pi(m)}\right\}.$$

We are now in the position to implement the method. For concreteness, we apply the RWMH algorithm on the horse-shoe shape in Exercise 8.3. We take the origin as the starting point m_0. Let us first be conservative by choosing a small proposal variance $\gamma^2 = 0.02^2$ so that the proposal p is very close to the current state m_k. In order to see how the MCMC chain evolves, we plot each state m_k as a circle (red) centered at m_k with radius proportional to the number of staying-puts. Figure 11.1(a) shows the results for $N = 1000$. We observe that the chain takes about 200 MCMC simulations to enter the high probability density region. This is known as *burn-in* time in MCMC literature, which tells us how long a MCMC chain takes to start exploring the density. In other words, after the burn-in time, a MCMC begins to distribute like the target density. As can be seen, the chain corresponding to small proposal variance γ^2 explores the target density very slowly. If we approximate the average acceptance rate by taking the ratio of the number of accepted proposal over N, it is 0.905 for this case. That is, almost all the proposals p are accepted, but exploring a very small region of high probability density.



(a) $\gamma = 0.02$ (b) $\gamma = 5$ (c) $\gamma = 0.5$

Fig. 11.1 RWMH with different proposal variance γ^2.

Let us now increase the proposal stepsize γ to 5, and we show the corresponding chain in Figure 11.1(b). This time, the chain immediately explores the target density without any burn-in time. However, it does so in an extremely slow manner. Most of the time the proposal p is rejected, resulting in a few big circles in Figure 11.1(b). The average acceptance rate in this case is 0.014, which shows that most of the time we reject proposals p.

The results in Figures 11.1(a) and 11.1(b) are two extreme cases, both of which explore the target density in a very lazy manner since the chain either accepts all the small moves with very high acceptance rate or rejects big moves with very low acceptance rate. This suggests that there must be an *optimal* acceptance rate for the RWMH algorithm. Indeed, one can show that the optimal acceptance rate is 0.234 [23]. For our horse-shoe target, it turns out that the corresponding optimal stepsize is approximately $\gamma = 0.5$. To confirm this, we generate a new chain with this stepsize, again with $N = 1000$, and show the result in Figure 11.1(c). As can be seen, the samples spread out nicely over the horse-shoe.

We have judged the quality and convergence of a MCMC chain by looking at the scatter plot of the samples. Another simple approach is to look at the trace plot of components of m. For example, we show the trace plot of the first component in Figure 11.2 for the above three stepsizes. The rule of thumb is that a Markov chain is considered to be good if its trace plot is close to a white noise one, a "fuzzy worm", in which all the samples are completely uncorrelated. Based on this criteria, we again conclude that $\gamma = 0.5$ is the best compared to the other two extreme cases.

Plot a trace plot for a one dimensional Gaussian white noise to see how it looks like!

Nevertheless, the above two simple criteria are neither rigorous nor possible in high dimensions. This observation immediately reminds us the strong law of large number in computing the mean and its dimension-independent error analysis using the central limit theorem. Since the target is symmetric about the vertical axis, the first component of the mean must be zero. Let us use the strong law of large number to estimate the means for the above three stepsizes and show them as cross signs in Figures 11.1(a), 11.1(b), and 11.1(c). As can be seen and expected, the sample mean for the optimal stepsize is the most accurate, though it is not exactly on the vertical axis since $\overline{m}_1 = -0.038$. This implies that the sample size of $N = 1000$ is small. If we take $N = 10000$, the sample mean gives $\overline{m}_1 = 0.003$, signifying the convergence when N increases.

Take $N = 100000$ for the optimal stepsize case, and again compute the sample mean using `BayesianMCMC.m`*. Is the sample mean better? If not, why?*

However, the application of LLN and CLT is very limited for Markov chains since they don't provide iid samples. Indeed, as in the above Markov chain theory, the states of the chain eventually identically distributed by $\pi(m)$, but they are always correlated instead of independent since any state in the chain depends on the previous one. What we could hope for is that the current state is effectively independent from its kth previous state. In that case, the effective number of iid samples is N/k, and the mean square error, by the central limit theorem, decays as $\sqrt{k/N}$. As the result, if k is large, the decay rate is very slow. How to estimate k is the goal of the *autocorrelation* study, as we now discuss.

We shall compute the autocorrelation for each component of m separately, therefore, without loss of generality, assume that $m \in \mathbb{R}^1$ and that the Markov chain $\{m_j\}_{j=0}^N$ has zero mean. Consider the following discrete convolution quantities

$$c_k = \sum_{j=0}^{N-k} m_{j+k} m_j, \quad ,k = 0, \ldots, N-1,$$

and define the autocorrelation of m with *lag k* as

$$\hat{c}_k = \frac{c_k}{c_0}, \quad k = 0, \ldots, N-1.$$

If $\hat{c}_k$ is zero, then we say that the correlation length of the Markov chain is approximately k, that is, any state m_j is considered to be insignificantly correlated to m_{j-k} (and hence any state before m_{j-k}), and to m_{j+k} (and hence any state after m_{j+k}). In other words, every kth sample point can be considered to be approximately independent. Note that this is simply a heuristic and one should be aware that independece implies un-correlation but not vice versa.

Let us now approximately compute the correlation length for three Markov chains corresponding to $\gamma = 0.02$, $\gamma = 0.5$, and $\gamma = 5$, respectively, with $N = 100000$. We first subtract away the sample mean as

$$z_j = m_j - \frac{1}{N+1} \sum_{i=0}^{N} m_i.$$

Then, we plot the autocorrelation functions $\hat{c}_k$ for each component of the zero mean sample $\{z_j\}_{j=0}^N$ in Figure 11.3. As can be observed, the autocorrelation length for the chain with optimal stepsize $\gamma = 0.5$ is about $k = 100$, while the others are much larger (not shown here). That is, every 100th sample point can be considered to be independent for $\gamma = 0.5$. The case with $\gamma = 0.02$ is the worst, indicating slow move around the target density. The stepsize of $\gamma = 5$ is better, but so big that the chain remains at each state for a long period of time, and hence autocorrelation length is still significant relatively to that of $\gamma = 0.5$.

Extensive MCMC methods including improvements on the standard RWMH algorithm can be found in [**?**]. Let us introduce two simple modifications through the following two exercises.

Exercise 11.1. Consider the following target density

$$\pi(m) \propto \exp\left(-\frac{1}{2\delta^2}\|m\|^2 - \frac{1}{2\sigma^2}\|y - h(m)\|^2\right), \tag{11.1}$$

where

$$h(m) = \begin{bmatrix} m_1^2 - m_2 \\ m_2/5 \end{bmatrix}, \quad y = \begin{bmatrix} -0.2 \\ 0.1 \end{bmatrix}.$$

Take $\delta = 1$ and $\sigma = 0.1$.

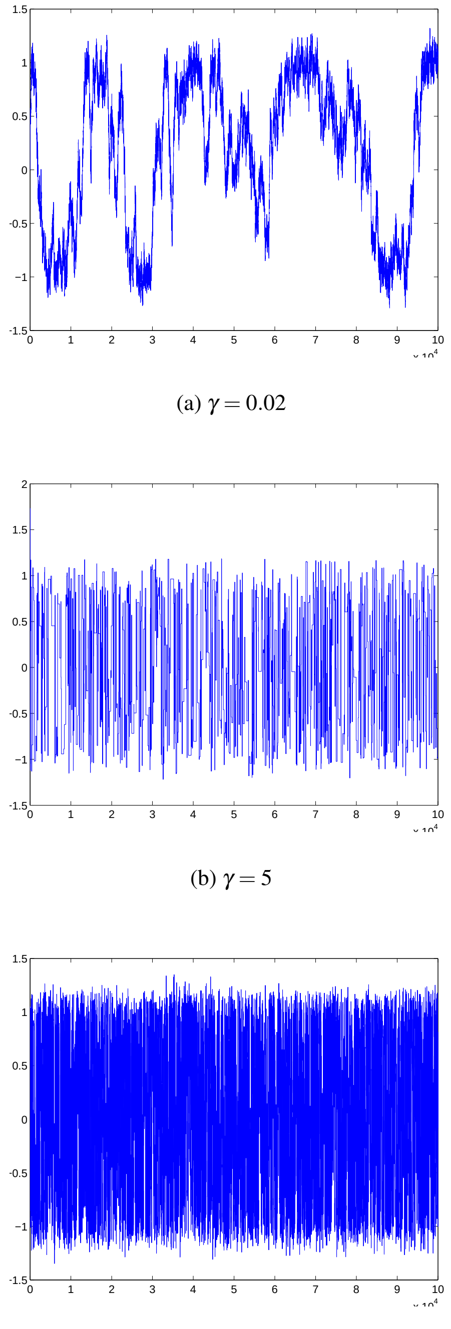
1. Modify `BayesianMCMC.m` to simulate the target density in (11.1) with $N = 5000$.
2. Tune the proposal stepsize γ so that the average acceptance probability is about 0.234. Show the scatter, trace, and autocorrelation plots for the optimal stepsize.

Exercise 11.2. So far the proposal density $q(m, p)$ is isotropic and independent of the target density $\pi(m)$. For anisotropic target density, isotropic proposal is not a good idea, intuitively. The reason is that the proposal is distributed equally in all directions, whereas it is not in the target density. A natural idea is to shape the proposal density to make it locally resemble the target density. A simple idea in this direction is to linearize $h(m)$, and then define the proposal density as

$$q(m_k, p) \propto \exp\left(-\frac{1}{2\delta^2}\|p\|^2 - \frac{1}{2\sigma^2}\|y - h(m_k) - \nabla h(m_k)(p - m_k)\|^2\right),$$

1. Determine $H(m_k)$ such that $q(m_k, p) = \mathcal{N}\left(m_k, H(m_k)^{-1}\right)$, by keeping only the quadratic term in $p - m_k$.
2. Modify `BayesianMCMC.m` to simulate the target density in (11.1) using the proposal density $q(m_k, p) = \mathcal{N}\left(m_k, H(m_k)^{-1}\right)$. Show the scatter, trace, and autocorrelation plots. Is it better than the isotropic proposal density?

Exercise 11.3. Another idea to improve the standard RWMH algorithm is by adaptation. Let's investigate a simple adaptation strategy. Use the resulting sample in Exercise 11.1 to compute the empirical covariance $\hat{\Gamma}$, then use it to construct the proposal density $q(m, p) = \mathcal{N}\left(m, \hat{\Gamma}\right)$. Show the scatter, trace, and autocorrelation plots. Is it better than the isotropic proposal density?

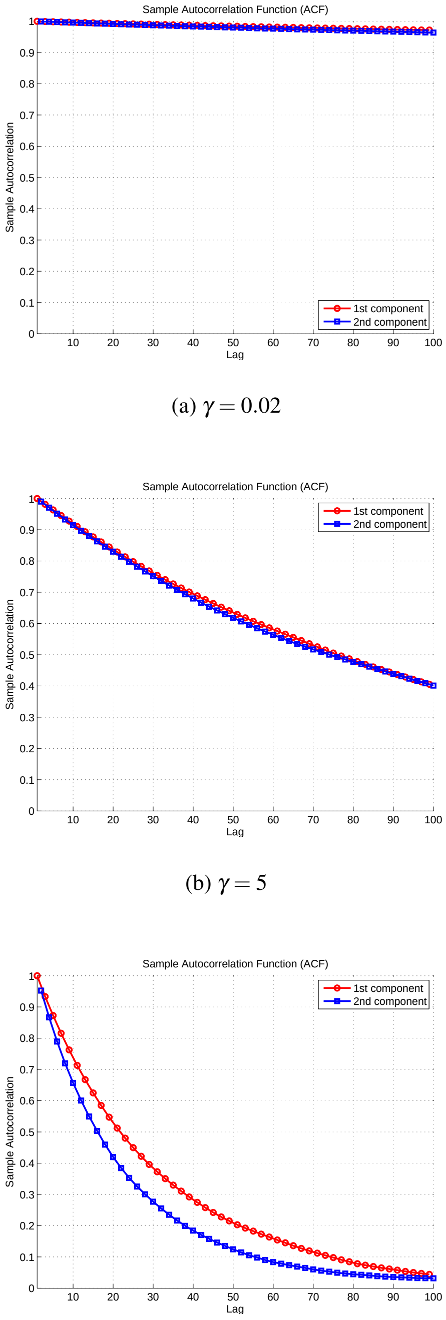


(a) $\gamma = 0.02$

(b) $\gamma = 5$

(c) $\gamma = 0.5$

Fig. 11.2 Trace plots of the first component of m with different γ^2.



(a) $\gamma = 0.02$

(b) $\gamma = 5$

(c) $\gamma = 0.5$

Fig. 11.3 Autocorrelation function plot for both components of m with different γ^2.