5

MARKOV CHAIN MONTE CARLO

In this chapter, we introduce Markov chains and then Markov Chain Monte Carlo (MCMC) methods. These methods are at the heart of Bayesian statistics, generative modelling, statistical physics, and many other fields. We will introduce the Metropolis-Hastings algorithm and then introduce the celebrated Gibbs sampler and, if time permits, some others.

In this chapter, we introduce a new sampling methodology - namely using Markov chains for sampling problems. This is a very powerful and widely used idea in statistics and machine learning. The idea is to set up Markov chains with prescribed stationary distributions. These distributions will be our target distributions.

In this chapter, we will adapt our notation and modify it to suit the new setting. From now on, we denote stationary/invariant distributions of Markov chains (which are also coincide with our target distributions) as $p_\star$. We will introduce discrete space Markov chains next.

5.1 DISCRETE STATE SPACE MARKOV CHAINS

A good setting for an introduction to Markov chains is the discrete space setting. In this setting, we have a finite set of states X where the cardinality of X is finite. We first define the Markov chain in this context.

Definition 5.1 (Markov chain)**.** *A discrete Markov chain is a sequence of random variables $X_0, X_1, \ldots, X_n$ such that*

$$\mathbb{P}(X_{n+1} = x_{n+1} \mid X_0 = x_0, \ldots, X_n = x_n) = \mathbb{P}(X_{n+1} = x_{n+1} \mid X_n = x_n).$$

In other words, a Markov chain is a sequence of random variables such that a given state at time n is conditionally independent of all previous states given the state at time

$n-1$. One can see that this describes many systems in the real world – as evolution of many systems can be summarised with the current state of the system and the evolution law.

An important quantity in the study of Markov chains is the transition matrix (or kernel in the continuous space case). This matrix defines the evolution structure of the chain and determines all of its properties. The transition matrix is defined as follows.

Definition 5.2 (Transition matrix)**.** *The transition matrix of a Markov chain is a matrix M such that*

$$M_{ij} = \mathbb{P}(X_{n+1} = j \mid X_n = i).$$

A usual way to depict Markov chains is the following conditional independence structure which sums the structure of the Markov chain up. We note that we will only consider the case

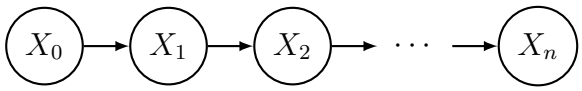


Figure 5.1: A Markov chain with n states.

where the transition matrix is time-homoegeneous, i.e., the transition matrix is the same for all times. We can see then that a Markov chain's behaviour is completely determined by its initial distribution and the transition matrix. We will denote the initial distribution of the chain as p_0 and note that this is a discrete distribution over the state space X (in this case)[1]. The transition matrix M is a matrix of size $d \times d$ where $d = |\mathsf{X}|$.

$$M = \begin{bmatrix} M_{11} & M_{12} & \cdots & M_{1d} \\ M_{21} & M_{22} & \cdots & M_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ M_{d1} & M_{d2} & \cdots & M_{dd} \end{bmatrix}.$$

We note that this matrix is stochastic, i.e. each row sums to 1:

$$\sum_{j=1}^{d} M_{ij} = 1,$$

since $M_{ij} = \mathbb{P}(X_{n+1} = j | X_n = i)$ and

$$\sum_{j=1}^{d} \mathbb{P}(X_{n+1} = j | X_n = i) = 1.$$

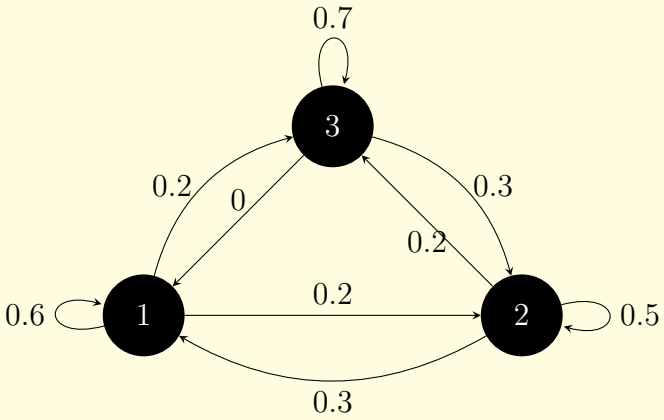
Next we will consider an example.

[1]When we move to continuous spaces, we will use the same notation for densities.

Example 5.1 (Discrete space Markov chain)**.** Consider the transition matrix:

$$M = \begin{bmatrix} 0.6 & 0.2 & 0.2 \\ 0.3 & 0.5 & 0.2 \\ 0 & 0.3 & 0.7 \end{bmatrix}, \qquad \text{where } \mathsf{X} = \{1, 2, 3\}.$$

The state transition diagram of this matrix can be described as follows.



This Markov chain can be simulated using the following idea. Given the above diagram, we can denote its transition matrix as a table:

M	$X_t = 1$	$X_t = 2$	$X_t = 3$
$X_{t-1} = 1$	0.6	0.2	0.2
$X_{t-1} = 2$	0.3	0.5	0.2
$X_{t-1} = 3$	0	0.3	0.7

Given $X_0 = 1$, how to simulate this chain? This boils down to just selecting the correct row from this matrix and then sampling using the discrete distribution given by that row. For example, if we sample from the first row, we get $X_1 = 1$ with probability 0.6, $X_1 = 2$ with probability 0.2 and $X_1 = 3$ with probability 0.2. We can then repeat this process for X_2 and so on. This is a simple way to simulate a Markov chain. The precise sampler is given below.

$$X_t | X_t = x_{t-1} \sim \text{Discrete}(M_{x_{t-1},\cdot}),$$

where the notation $M_{x_{t-1},\cdot}$ denotes the x_{t-1}th row of the transition matrix M (where $x_{t-1} \in \{1, 2, 3\}$ naturally).

We can also compute n-step transition matrix:

$$M^{(n)} = \mathbb{P}(X_n = j | X_0 = i),$$

where $M^{(n)}$ is a matrix of size $d \times d$. For this, see that n-step transition matrix can be

written as:

$$\begin{aligned} M_{ij}^{(n)} &= \mathbb{P}(X_n = j | X_0 = i) \\ &= \sum_k \mathbb{P}(X_n = j, X_1 = k | X_0 = i) \\ &= \sum_k \mathbb{P}(X_n = j | X_1 = k, X_0 = i) \mathbb{P}(X_1 = k | X_0 = i) \\ &= \sum_k \mathbb{P}(X_n = j | X_1 = k) \mathbb{P}(X_1 = k | X_0 = i) \\ &= \sum_k M_{ik} M_{kj}^{(n-1)}. \end{aligned}$$

Therefore, $M^{(n)} = M^n$ which is the nth power of the transition matrix. Note that we can compute in general the conditional distributions of the Markov chain by summing out the variables in the middle. For example, in order to compute $\mathbb{P}(X_{n+2} = x_{n+2} | X_n = x_n)$, we can write

$$\mathbb{P}(X_{n+2} = x_{n+2} | X_n = x_n) = \sum_{x_{n+1}} \mathbb{P}(X_{n+2} = x_{n+2} | X_{n+1} = x_{n+1}) \mathbb{P}(X_{n+1} = x_{n+1} | X_n = x_n).$$

This will lead us to define the Chapman-Kolmogorov equation, which is a generalization of the n-step transition matrix:

$$\begin{aligned} M^{(m+n)} &= \mathbb{P}(X_{m+n} = j | X_0 = i) \\ &= \sum_k \mathbb{P}(X_{m+n} = j | X_n = k) \mathbb{P}(X_n = k | X_0 = i) \\ &= \sum_k M_{ik}^{(m)} M_{kj}^{(n)}. \end{aligned}$$

Therefore, we can write $M^{m+n} = M^m M^n$.

It is also important to define the evolution of the chain. Note that we defined our initial distribution as p_0 and it is important to quantify how this distribution evolves over time. We denote the distribution at time n as p_n and write Then, the density of the chain at time n is given by:

$$\begin{aligned} p_n(i) &= \mathbb{P}(X_n = i) \\ &= \sum_k \mathbb{P}(X_n = i, X_{n-1} = k) \\ &= \sum_k \mathbb{P}(X_n = i | X_{n-1} = k) \mathbb{P}(X_{n-1} = k) \\ &= \sum_k M_{ki} p_{n-1}(k). \end{aligned}$$

This implies that

$$p_n = p_{n-1} M.$$

Therefore,

$$p_n = p_0 M^n.$$

These are important equations, which will have corresponding equations in the continuous case (however, they will be integrals).

Since we have expressed our interest in Markov chains because of their potential utility in sampling, we will now discuss the properties we need to ensure that we can use Markov chains for sampling. In short, we need Markov chains that have (i) invariant distributions, (ii) their convergence to invariant distributions are ensured, (iii) the invariant distribution is unique. We will now discuss the properties we need to ensure these in detail.

5.1.1 IRREDUCIBILITY

The first property we need to ensure is that the Markov chain is irreducible. This means that there is a path from any state to any other state . To be precise, let $x, x' \in \mathsf{X}$ be any two states. We write $x \rightsquigarrow x'$ if there is a path from x to x':

$$\exists n > 0, \text{ s.t. }, \mathbb{P}(X_n = x' | X_0 = x) > 0.$$

If $x \rightsquigarrow x'$ and $x' \rightsquigarrow x$, then we say that x and x' *communicate*. We then define the *communication class* $C \subset \mathsf{X}$ which is a set of states such that $x \in C$ and $x' \in C$ if and only if $x \rightsquigarrow x'$ and $x' \rightsquigarrow x$. A chain is irreducible if X is a single communication class. This simply means that there is a positive probability of moving around to every other state. This makes sense as without ensuring this, we won't be sampling from the full support.

5.1.2 RECURRENCE AND TRANSIENCE

We now define the notion of recurrence and transience. A state $i \in \mathsf{X}$ is *recurrent* if there is a positive probability of returning to i. In order to see define this, consider the return time

$$\tau_i = \inf\{n \geq 1 : X_n = i\}.$$

We say that the state i is recurrent if

$$\mathbb{P}(\tau_i < \infty | X_0 = i) = 1.$$

In other words, the probability of waiting time being finite is 1. If a chain is not recurrent, it is said to be transient. We can also further define the *positive recurrence* which is a slightly stronger (better) condition. We say that i is positively recurrent if

$$\mathbb{E}[\tau_i | X_0 = i] < \infty.$$

This means that the expected waiting time is finite. If a chain is recurrent but not positive recurrent, then it is called null recurrent.

5.1.3 INVARIANT DISTRIBUTIONS

In the discrete time case, a distribution $p_\star$ is called invariant if

$$p_\star = p_\star M.$$

This means that the chain is reach stationarity, i.e., evolving further (via M) does not change the distribution. We have then the following theorem (Yıldırım, 2017).

Theorem 5.1. *If M is irreducible, then M has a unique invariant distribution if and only if it is positive recurrent.*

This is encouraging however for actual convergence of the chain to this distribution, we will need more conditions.

5.1.4 REVERSIBILITY AND DETAILED BALANCE

We define the detailed balance condition as

$$p_\star(i)M_{ij} = p_\star(j)M_{ji}.$$

This trivially implies that $p_\star = p_\star M$, hence the invariance of $p_\star$. We will have a more detailed discussion of this condition in the continuous state space case.

5.1.5 CONVERGENCE TO INVARIANT DISTRIBUTION

Finally, we need the ergodicity condition to ensure that the chain converges to the invariant distribution. For this, we require the chain to be aperiodic, which is defined as follows. A state i is called aperiodic if

$$\{n > 0 : \mathbb{P}(X_{n+1} = i | X_1 = i) > 0\}$$

has no common divisor other than 1. A Markov chain is called aperiodic if all states are aperiodic. An irreducible Markov chain is called ergodic if it is positive recurrent and aperiodic. If $(X_n)_{n\in\mathbb{N}}$ is an ergodic Markov chain with initial p_0 and $p_\star$ as its invariant distribution, then

$$\lim_{n\to\infty} p_n(i) = p_\star(i).$$

Moreover, for $i, j \in \mathsf{X}$

$$\lim_{n\to\infty} \mathbb{P}(X_n = i | X_1 = j) = p_\star(i).$$

In other words, the chain will converge to its invariant distribution from every state.

5.2 CONTINUOUS STATE SPACE MARKOV CHAINS

Our main interest is in the continuous case. However, it is important to understand the definitions above – as we will not go into analogous definitions in the continuous case. The reason for this is that, in continuous cases, the individual states have zero probability (i.e. a point has zero probability) and all the notions above are defined using sets and measure theoretic concepts. We focus on simulation methods within this course, therefore, we will not go into reviewing this material. A couple of very good books for this are Douc et al. (2018) and Douc et al. (2013).

Let X be an uncountable set from now on, e.g., $\mathsf{X} = \mathbb{R}$ or $\mathsf{X} = \mathbb{R}^d$. We denote the initial density as $p_0(x)$ as usual, the transition kernel with $K(x|x')$, the marginal density of the chain at time n as $p_n(x)$.

We can write the Markov property in this case as follows. For any measurable A

$$\mathbb{P}(X_n \in A | X_1 = x_1, \ldots, X_{n-1} = x_{n-1}) = \mathbb{P}(X_n \in A | X_{n-1} = x_{n-1}).$$

This implies that if we write down the joint distribution of $X_{1:n}$, then the following factorisation holds

$$p(x_0, \ldots, x_n) = \prod_{k=0}^{n} p(x_k | x_{k-1}),$$

where $p(x_0|x_{-1}) := p_0(x_0)$. We also assume that the transition kernel has a density which we denote as $K(x_n|x_{n-1})$ at time n. Similarly to the discrete case, we will assume that the density is time-homogeneous (i.e. same for every n). Note that the transition density is a density in its first variable, i.e.,

$$\int_{\mathsf{X}} K(x_n|x_{n-1}) dx_n = 1.$$

It is a function of x_{n-1} otherwise. We give an example of a continuous state-space Markov chain in what follows.

Example 5.2 (Simulation of a Markov process)**.** Consider the following Markov chain with $X_0 = 0$

$$X_n | X_{n-1} = x_{n-1} \sim \mathcal{N}(x_n; a x_{n-1}, 1), \tag{5.1}$$

with $0 < a < 1$. We can simulate this chain by

$$\begin{aligned} X_1 &\sim \mathcal{N}(0, 1) \\ X_2 &\sim \mathcal{N}(aX_1, 1) \\ X_3 &\sim \mathcal{N}(aX_2, 1) \\ &\vdots \\ X_n &\sim \mathcal{N}(aX_{n-1}, 1). \end{aligned}$$

How to do this? We also note that Eq. (5.1) can also be expressed as

$$X_n = aX_{n-1} + \epsilon_n$$

where $\epsilon_n \sim \mathcal{N}(0, 1)$. This is also called AR(1) process. From the last equation, it must be clear how to simulate this as you only need a `for` loop and samples from $\mathcal{N}(0, 1)$.

Similar to the continuous case, we can define the distribution of X_n given a past variable X_{n-k} by integrating out the variables in between . It is important to note that, X_n is independent of past variables **if** (and a big if) $X_{n-1} = x_{n-1}$ is given. Otherwise, we can write down the densities as

$$p(x_n|x_{n-k}) = \int \cdots \int K(x_n|x_{n-1}) K(x_{n-1}|x_{n-2}) \cdots K(x_{n-k+1}|x_{n-k}) dx_{n-1} \cdots dx_{n-k+1}.$$

We define the m-step transition kernel as

$$K^{(m)}(x_{m+n}|x_n) = \int_{\mathsf{X}} K(x_{m+n}|x_{m+n-1}) \cdots K(x_{n+1}|x_n)\, \mathrm{d}x_{m+n-1} \cdots \mathrm{d}x_{n+1}.$$

We now provide the definition of invariance in this context, w.r.t. to the transition kernel.

Definition 5.3 (K-invariance)**.** *A probability measure $p_\star$ is called K-invariant if*

$$p_\star(x) = \int_{\mathsf{X}} K(x|x')\, p_\star(x') \mathrm{d}x'. \tag{5.2}$$

It can be seen that $p_\star$ being invariant means that the kernel operating on $p_\star$ results in the same distribution $p_\star$ (the integral agains the kernel can be seen as a transformation, similar to the matrix product in the discrete case). Finally, we get to the detailed balance condition.

Definition 5.4 (Detailed balance)**.** *A transition kernel K is said to satisfy detailed balance if*

$$K(x'|x)p_\star(x) = K(x|x')p_\star(x'). \tag{5.3}$$

We note that this is a sufficient condition for stationarity of $p_\star$.

Proposition 5.1 (Detailed balance implies stationarity)**.** *If K satisfies detailed balance, then $p_\star$ is the invariant distribution.*

Proof. The proof is a one-liner:

$$\int p_\star(x) K(x'|x) \mathrm{d}x' = \int p_\star(x') K(x|x') \mathrm{d}x',$$

which is just integrating both sides after writing the detailed balance condition. The lhs of this equation is $p_\star(x)$ since $K(x'|x)$ integrates to 1 which leaves us with the definition of K-invariance as given in (5.2). □

Let us see an example of a continuous space Markov chain (or rather go back to AR(1) example).

Example 5.3. Consider again the Markov chain with the following transition kernel

$$K(x_n|x_{n-1}) = \mathcal{N}(x_n; ax_{n-1}, 1).$$

We can also describe the evolution this chain as a recursion, as mentioned before

$$X_n = aX_{n-1} + \epsilon_n$$

where $\epsilon_n \sim \mathcal{N}(0, 1)$. This is a nice example where the stationarity distribution can be computed analytically. It is easy to check, for example,

$$p_\star(x) = \mathcal{N}(x; 0, \frac{1}{1-a^2}).$$

by checking the detailed balance (see relevant exercise solutions). One can also prove that the m-step transition kernel is given by

$$K^{(m)}(x_{m+n}|x_n) = \mathcal{N}(x_{m+n}; a^m x_n, \frac{1-a^{2m}}{1-a^2}).$$

This implies trivially that

$$p_\star(x) = \lim_{m\to\infty} K^{(m)}(x|x').$$

for any x'. In other words, starting from any x', the chain will reach stationarity. The proofs of these results are left as an exercise (as usual, solutions will be posted).

We have now almost everything we need to move on to discuss Metropolis-Hastings method.

5.3 METROPOLIS-HASTINGS ALGORITHM

We finally have all the ingredients to define the celebrated Metropolis-Hastings algorithm. We will not need more technicalities in defining it.

The Metropolis-Hastings (MH) algorithm is a remarkable method which allows us to define transition kernels (defined implicitly via the algorithm) where the detailed balance is satisfied w.r.t. any $p_\star$ we wish to sample from. I call this *remarkable* because it rids us of the need of designing Markov kernels for specific probability distributions and provides a generic way to design samplers that will target any measure we want. The algorithm relies on the idea of using *local* proposals $q(x'|x)$ and accepting them with a certain acceptance ratio. The acceptance ratio is designed so that the resulting samples $X_1, \ldots, X_n$ from the method form a Markov chain that leaves $p_\star$ invariant. We will provide the algorithm below, as seen from Algorithm 9. Note, as mentioned in the lecture, the last step of the method: When a sample is rejected, we do not sample again – we set $X_n = X_{n-1}$ and continue sampling the next sample. This means that, if the rejection rate is high, there will be a lot of duplicated samples and this is the expected behaviour. Another important note is about the `burnin` period. Any Markov chain started at a random point will take some time to reach stationarity (the whole magic is to be able to make them converge faster). Therefore, we discard the first `burnin` samples and only return the remaining ones. This is a common practice in MCMC methods.

Algorithm 9 Pseudocode for Metropolis Hastings method

1: Input: The number of samples N, and starting point X_0.
2: **for** $i = 1, \ldots, N$ **do**
3: Propose (sample): $X' \sim q(x'|X_{n-1})$
4: Accept the sample X' with probability

$$\alpha(X_{n-1}, X') = \min\left\{1, \frac{p_\star(X')q(X_{n-1}|X')}{p_\star(X_{n-1})q(X'|X_{n-1})}\right\}.$$

5: Otherwise reject the sample and set $X_n = X_{n-1}$.
6: **end for**
7: Discard first `burnin` samples and return the remaining samples.

We define the acceptance ratio as

$$\mathsf{r}(x, x') = \frac{p_\star(x')q(x|x')}{p_\star(x)q(x'|x)}. \tag{5.4}$$

We also note that in the practical algorithm, one does not need to implement the $\min$ operation. For accepting with a certain probability (like in the rejection sampling), we draw $U \sim \text{Unif}(0, 1)$ and check if $U \leq \alpha(X_{n-1}, X')$. However, if the ratio $\mathsf{r}(X_{n-1}, X')$ is greater than 1, this sample is always going to be accepted anyway. The $\min$ operation is important however for theoretical properties of the kernel to hold.

As we mentioned above, the algorithm provides us with an implicit kernel $K(x_n|x_{n-1})$ – if you think about it, it is just a way to get X_n given X_{n-1}. The specific structure of the algorithm, however, ensures that we leave the right kind of distribution invariant – i.e. $p_\star$ – that is our target measure. We elucidate this in the following proposition.

Proposition 5.2 (Metropolis-Hastings satisfies detailed balance)**.** *The Metropolis-Hastings algorithm satisfies detailed balance w.r.t. $p_\star$, i.e.,*

$$p_\star(x)K(x|x') = p_\star(x')K(x'|x),$$

where K is the kernel defined by the MH algorithm.

Proof. We first define the kernel induced by the MH algorithm. This can be seen by inspecting the algorithm:

$$K(x'|x) = \alpha(x, x')q(x'|x) + (1 - a(x))\delta_x(x'),$$

where δ_x is the Dirac delta function and

$$a(x) = \int_{\mathsf{X}} \alpha(x, x')q(x'|x)\mathrm{d}x',$$

is the probability of accepting a sample (hence $1 - a(x)$ is the probability of rejecting a new sample while at point x). See Sec. 2.3.1 of Douc et al. (2018) for a rigorous derivation.

Given this, we write

$$\begin{aligned}
p_\star(x)K(x'|x) &= p_\star(x)q(x'|x)\alpha(x',x) + p_\star(x)(1-a(x))\delta_x(x') \\
&= p_\star(x)q(x'|x)\min\left\{1, \frac{p_\star(x')q(x|x')}{p_\star(x)q(x'|x)}\right\} + p_\star(x)(1-a(x))\delta_x(x') \\
&= \min\left\{p_\star(x)q(x'|x), p_\star(x')q(x|x')\right\} + p_\star(x)(1-a(x))\delta_x(x') \\
&= \min\left\{\frac{p_\star(x)q(x'|x)}{p_\star(x')q(x|x')}, 1\right\} p_\star(x')q(x|x') + p_\star(x')(1-a(x'))\delta_{x'}(x) \\
&= K(x|x')p_\star(x'),
\end{aligned}$$

which shows that the detailed balance holds! □

One can see that the algorithm works just the same with unnormalised densities, i.e., recall

$$p_\star(x) = \frac{\bar{p}_\star(x)}{Z},$$

where Z is the normalisation constant. In this case, the acceptance ratio becomes

$$\mathsf{r}(x,x') = \frac{\bar{p}_\star(x')q(x|x')}{\bar{p}_\star(x)q(x'|x)},$$

without any change as the normalising constants cancel out in (5.4). We will next describe certain classes of proposals to sample from various kinds of distributions and assess their performance.

5.3.1 INDEPENDENT PROPOSALS

An important class of proposals that is used in practice is the independent proposal. Note that in general we denoted our proposal with $q(x'|x)$, in particular, we would sample from $q(x'|x_{n-1})$ implying that in general the proposal uses the current state of the chain. This does not have to be the case and we can as well chose just an independent proposal $q(x')$ to ease computations. The acceptance ratio in this specific case becomes

$$\mathsf{r}(x,x') = \frac{\bar{p}_\star(x')q(x)}{\bar{p}_\star(x)q(x')}.$$

In the algorithm, this means that we compute

$$\alpha(X_{n-1}, X') = \min\left\{1, \frac{\bar{p}_\star(X')q(X_{n-1})}{\bar{p}_\star(X_{n-1})q(X')}\right\}.$$

We will see one example as follows.

Example 5.4 (Independent Gaussian proposal)**.** Consider a Gaussian (artificial) target:

$$p_\star(x) = \mathcal{N}(x;\mu,\sigma^2)$$

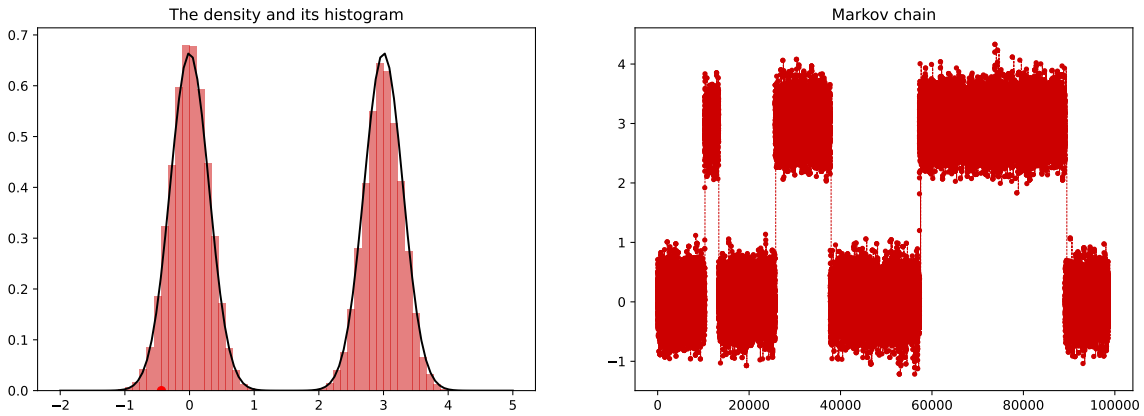


Figure 5.2: Random walk Gaussian proposal for a mixture of two Gaussians.

Assume we want to use MH to sample from it. Choose a proposal

$$q(x) = \mathcal{N}(x; \mu_q, \sigma_q^2).$$

The acceptance ratio can be computed in this case as:

$$\begin{aligned}
\mathsf{r}(x, x') &= \frac{p_\star(x')q(x)}{p_\star(x)q(x')} \\
&= \frac{\mathcal{N}(x'; \mu, \sigma^2)\mathcal{N}(x; \mu_q, \sigma_q^2)}{\mathcal{N}(x; \mu, \sigma^2)\mathcal{N}(x'; \mu_q, \sigma_q^2)} \\
&= \frac{\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x'-\mu)^2}{2\sigma^2}\right) \frac{1}{\sqrt{2\pi\sigma_q^2}} \exp\left(-\frac{(x-\mu_q)^2}{2\sigma_q^2}\right)}{\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \frac{1}{\sqrt{2\pi\sigma_q^2}} \exp\left(-\frac{(x'-\mu_q)^2}{2\sigma_q^2}\right)} \\
&= \frac{\exp\left(-\frac{(x'-\mu)^2}{2\sigma^2}\right) \exp\left(-\frac{(x-\mu_q)^2}{2\sigma_q^2}\right)}{\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \exp\left(-\frac{(x'-\mu_q)^2}{2\sigma_q^2}\right)} \\
&= e^{\left(-\frac{1}{2\sigma^2}\left[(x'-\mu)^2-(x-\mu)^2\right]\right)} e^{\left(-\frac{1}{2\sigma_q^2}\left[(x-\mu_q)^2-(x'-\mu_q)^2\right]\right)}
\end{aligned}$$

5.3.2 Random walk (symmetric) proposals

Another important class of proposals is the random walk proposal. In this case, the proposal does use the current state X_{n-1} to define a proposal $q(x'|x_{n-1})$. These proposals in the random walk (and more generally symmetric) case result in a density that is symmetric, i.e., $q(x'|x) = q(x|x')$. This leads to a considerable simplification in the acceptance ratio calculations. We will see some examples below.

Example 5.5 (Random walk Gaussian proposal)**.** Consider a Gaussian (artificial) target:

$$p_\star(x) = w_1\mathcal{N}(x; \mu_1, \sigma_1^2) + w_2\mathcal{N}(x; \mu_2, \sigma_2^2)$$

Assume we want to use MH to sample from it. Choose a proposal

$$q(x'|x) = \mathcal{N}(x'; x, \sigma_q^2).$$

This proposal is symmetric so we can write

$$\begin{aligned}
\mathsf{r}(x, x') &= \frac{p_\star(x')q(x|x')}{p_\star(x)q(x'|x)} \\
&= \frac{p_\star(x')}{p_\star(x)}, \\
&= \frac{\mathcal{N}(x'; \mu, \sigma^2)}{\mathcal{N}(x; \mu, \sigma^2)} \\
&= e^{\left(-\frac{1}{2\sigma^2}\left[(x'-\mu)^2-(x-\mu)^2\right]\right)},
\end{aligned}$$

which is a considerable simplification. See Fig. 5.2 for a demonstration.

5.3.3 GRADIENT BASED (LANGEVIN) PROPOSALS

One of the powerful proposal alternatives is to chosoe the proposal based on the gradient of the target distribution $p_\star$. Note that we can compute $\nabla \log p_\star(x)$ without necessarily needing the normalising constant, since

$$\nabla \log p_\star(x) = \nabla \log \bar{p}_\star(x) - \underbrace{\nabla \log Z}_{0}$$

Therefore, without doing much more than what we are already doing (using unnormalised density), we can *inform* the proposal by using the gradient of the target distribution:

$$q(x'|x) = \mathcal{N}(x'; x + \gamma \nabla \log p_\star(x), 2\gamma I),$$

This algorithm is widely popular in the fields of statistics and especially in machine learning. This approach is called *Metropolis adjusted Langevin algorithm* (MALA)

5.3.4 BAYESIAN INFERENCE WITH METROPOLIS-HASTINGS

We can finally use the Metropolis-Hastings method for Bayesian inference. In what follows, we will provide some examples for this and visualisations resulting from the sampling procedures.

Recall that, with conditionally independent observations $y_1, \ldots, y_n$, we have the Bayes theorem as

$$p(x|y_{1:n}) = \frac{p(y_{1:n}|x)p(x)}{p(y_{1:n})} = \frac{\prod_{i=1}^n p(y_i|x)p(x)}{p(y_{1:n})}.$$

We write

$$p(x|y_{1:n}) \propto \prod_{i=1}^n p(y_i|x)p(x),$$

and set

$$\bar{p}_\star(x) = \prod_{i=1}^{n} p(y_i|x)p(x),$$

which is the unnormalised posterior density. We can then use the Metropolis-Hastings algorithm to sample from this posterior density. A generic Metropolis-Hastings method for Bayesian inference is described in Algorithm 10.

Algorithm 10 Pseudocode for Metropolis Hastings method for Bayesian inference

1: Input: The number of samples N, and starting point X_0.
2: **for** $i = 1, \ldots, N$ **do**
3: Propose (sample): $X' \sim q(x'|X_{n-1})$
4: Accept the sample $X_n = X'$ with probability

$$\alpha(X_{n-1}, X') = \min\left\{1, \frac{\bar{p}_\star(x')q(x_{n-1}|x')}{\bar{p}_\star(x_{n-1})q(x'|x_{n-1})}\right\}.$$

5: Otherwise reject the sample and set $X_n = X_{n-1}$.
6: **end for**
7: Discard first `burnin` samples and return the remaining samples.

Example 5.6 (Source localisation)**.** This is an example taken from Cemgil (2014) which is another excellent tutorial which shaped much of my thinking – and the same example appears in Yıldırım (2017). Consider the problem of source localisation in the presence of three sensors with three noisy observations. The setup in this example can be seen from the left part of Fig. 5.3. We have three sensors surrounding an object we are trying to locate. The sensors receive noisy observations on $\mathbb{R}^2$. We are trying to locate the object based on these observations. We define our prior rather broadly: $p(x) = \mathcal{N}(x; \mu, \sigma^2 I)$ where $\mu = [0, 0]$ and $\sigma^2 = 20$. We assume that the observations are coming from

$$p(y_i|x, s_i) = \mathcal{N}(y_i; \|x - s_i\|, \sigma_y^2),$$

where s_i is the location of the ith sensor on $\mathbb{R}^2$ for $i = 1, 2, 3$. We assume that the observations are independent and that the noise is independent of the location of the object (of course, for the sake of the example, we simulate our observations from the true model which is not the case in the real world). We are interested in the posterior density of x, i.e., the distribution over the location of the hidden object:

$$p(x|y_1, y_2, y_3, s_1, s_2, s_3) \propto p(y_1, y_2, y_3|x, s_1, s_2, s_3)p(x),$$

and given conditional independence we have

$$p(x|y_1, y_2, y_3, s_1, s_2, s_3) \propto p(x) \prod_{i=1}^{3} p(y_i|x, s_i).$$

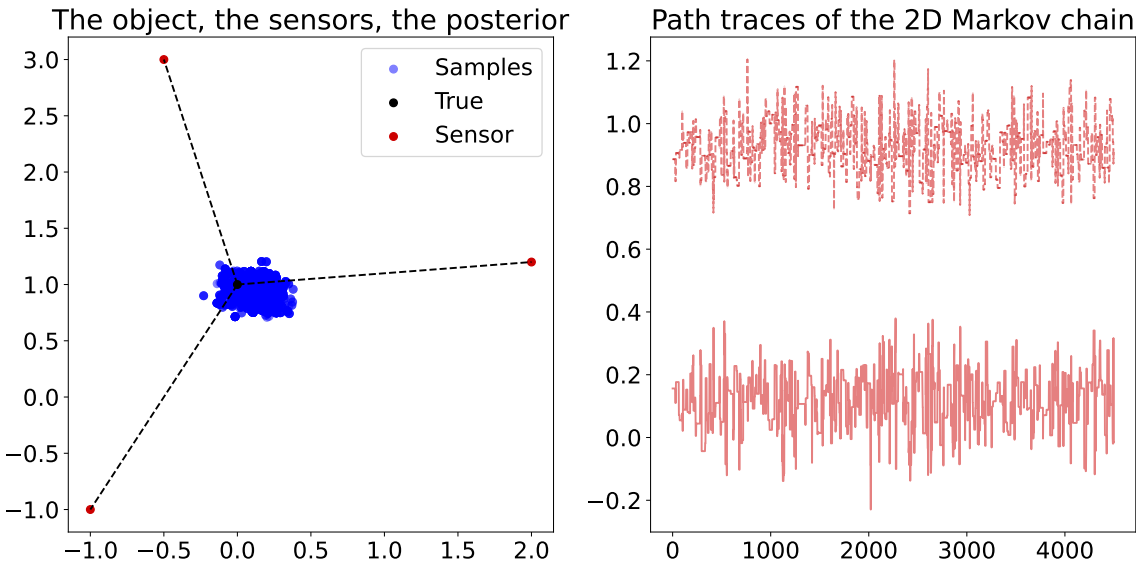


Figure 5.3: Solution of the source localisation problem.

This sort of Bayes update follows from the conditional Bayes rule introduced in Prop. 3.2. In order to design the MH scheme, therefore, we need to just evaluate the likelihood and the prior for MH. We choose a random walk proposal:

$$q(x'|x) = \mathcal{N}(x'; x, \sigma^2 I).$$

This is symmetric so the acceptance ratio is:

$$\mathsf{r}(x, x') = \frac{p(x')p(y_1|x', s_1)p(y_2|x', s_2)p(y_3|x', s_3)}{p(x)p(y_1|x, s_1)p(y_2|x, s_2)p(y_3|x, s_3)}.$$

An example solution to this problem can be seen from Fig. 5.3.

Example 5.7 (Gaussian with unknown mean and variance Example 5.13 in Yıldırım (2017))**.** Assume that we observe

$$Y_1, \ldots, Y_n|z, s \sim \mathcal{N}(y_i; z, s)$$

where we do not know z and s. Assume we have an independent prior on z and s:

$$p(z)p(s) = \mathcal{N}(z; m, \kappa^2)\mathcal{IG}(s; \alpha, \beta).$$

where $\mathcal{IG}(s; \alpha, \beta)$ is the inverse Gamma distribution

$$\mathcal{IG}(s; \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} s^{-\alpha-1} \exp\left(-\frac{\beta}{s}\right).$$

In other words, we have

$$p(z)p(s) = \frac{1}{\sqrt{2\pi\kappa^2}} \exp\left(-\frac{(z-m)^2}{2\kappa^2}\right) \frac{\beta^\alpha}{\Gamma(\alpha)} s^{-\alpha-1} \exp\left(-\frac{\beta}{s}\right).$$

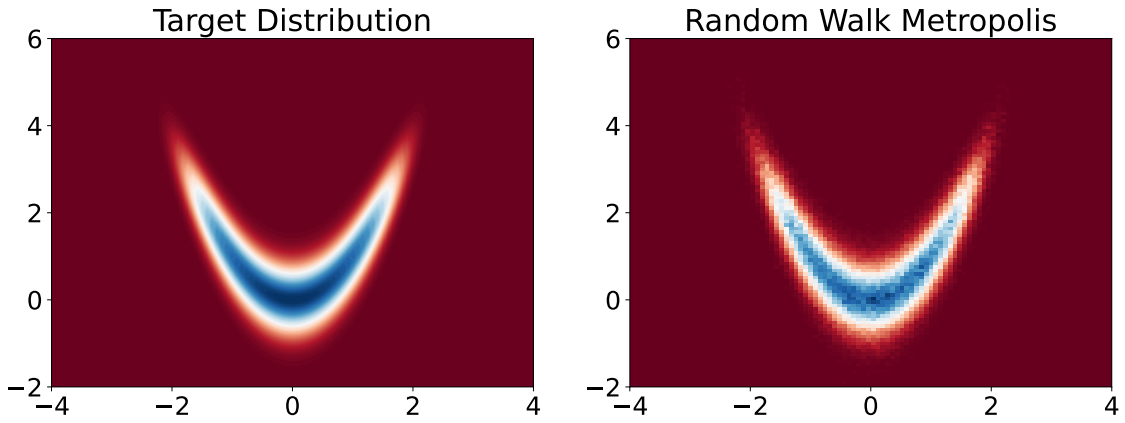


Figure 5.4: Banana density estimation using Random walk metropolis and plotting the histogram.

We are after the posterior distribution

$$\begin{aligned} p(z, s|y_1, \ldots, y_n) &\propto p(y_1, \ldots, y_n|z, s)p(z)p(s), \\ &= \prod_{i=1}^{n} \mathcal{N}(y_i; z, s)\mathcal{N}(z; m, \kappa^2)\mathcal{IG}(s; \alpha, \beta). \end{aligned}$$

Let us call our unnormalised posterior as $\bar{p}_\star(z, s|y_{1:n})$. In order to do this, we need to design proposals over z and s. We choose a random walk proposal for z:

$$q(z'|z) = \mathcal{N}(z'; z, \sigma_q^2).$$

and an independent proposal for s:

$$q(s') = \mathcal{IG}(s'; \alpha, \beta).$$

The joint proposal therefore is

$$q(z', s'|z, s) = \mathcal{N}(z'; z, \sigma_q^2)\mathcal{IG}(s'; \alpha, \beta).$$

When we design the MH algorithm, we see that the acceptance ratio is

$$\begin{aligned} \mathsf{r}(z, s, z', s') &= \frac{\bar{p}(z', s'|y_{1:n})q(z, s|z', s')}{p(z, s|y_{1:n})q(z', s'|z, s)} \\ &= \frac{p(z')p(s')\left[\prod_{k=1}^{n}\mathcal{N}(y_k; z', s')\right]\mathcal{N}(z; z', \sigma_q^2)p(s)}{p(z)p(s)\left[\prod_{k=1}^{n}\mathcal{N}(y_k; z, s)\right]\mathcal{N}(z'; z, \sigma_q^2)p(s')} \\ &= \frac{\mathcal{N}(z'; m, \kappa^2)\left[\prod_{k=1}^{n}\mathcal{N}(y_k; z', s')\right]}{\mathcal{N}(z; m, \kappa^2)\left[\prod_{k=1}^{n}\mathcal{N}(y_k; z, s)\right]} \end{aligned}$$

Example 5.8 (The banana density)**.** Consider the following density:

$$p(x, y) \propto \exp\left(-\frac{x^2}{10} - \frac{y^4}{10} - 2(y - x^2)^2\right).$$

This is only available in unnormalised form and it is an excellent test problem for many algorithms to fail. We have

$$\bar{p}_\star(x, y) = \exp\left(-\frac{x^2}{10} - \frac{y^4}{10} - 2(y - x^2)^2\right).$$

and let us choose the proposal

$$q(x', y'|x, y) = \mathcal{N}(x'; x, \sigma_q^2)\mathcal{N}(y'; y, \sigma_q^2).$$

This is a symmetric proposal so the acceptance ratio is

$$\mathsf{r}(x, y, x', y') = \frac{\bar{p}_\star(x', y')}{\bar{p}_\star(x, y)}.$$

Note that it makes sense to only compute log-acceptance ratio here

$$\log \mathsf{r}(x, y, x', y') = \log \bar{p}_\star(x', y') - \log \bar{p}_\star(x, y),$$

and implement the acceptance rate by drawing $U \sim \text{Unif}(0, 1)$ and accepting if $\log U < \log \mathsf{r}(x, y, x', y')$. The result can be seen from Fig. 5.4.