Corollary 6.17. *If the Metropolis-Hastings transition probability K targetting p is irreducible on $\mathbb{S} = \operatorname{supp}(p)$, then for any function $f : \mathbb{X} \to \mathbb{R}$ with $\mathbb{E}_p[f(X)]$ finite*

$$I^{(n)}_{p,q,\mathrm{MH}}(f) = \frac{1}{n}\sum_{k=1}^{n} f(X_k) \xrightarrow{n\to\infty} \mathbb{E}_p[f(X)] \quad \textit{(almost surely).}$$

Remark 6.18. Irreducibility is ensured by proper choice of proposal distributions $q(x, y)$. The proposal distributions need to be defined so that every point $y \in \mathbb{S}$ is reachable from any $x \in \mathbb{S}$ in $n = n(x, y)$ steps.

Example 6.19. Let $p(x) = x/Z_p$ for $x \in \mathbb{X} := \{1, \dots, m\}$ with $Z_p = \sum_{x=1}^{m} x$. Let us design a Metropolis-Hastings algorithm targetting p.

Step 1: Choose a proposal distribution $q(x, y)$. It needs to be easy to simulate and to determine an irreducible chain. A simple distribution that 'will do' is drawing $Y_k \sim \mathcal{U}(\mathbb{X})$ independent of X_{k-1}, so

$$q(x, y) = q(y) = 1/m, \quad y \in \mathbb{X}$$

This proposal scheme is irreducible, because for all $x, y \in \mathbb{X}$,

$$\begin{aligned}\mathbb{P}(X_1 = y \mid X_0 = x) &\geq q(x, y) \min\left\{1, \frac{p(y)}{p(x)}\frac{q(y, x)}{q(x, y)}\right\} \\ &= \frac{1}{m} \min\left\{1, \frac{y}{x}\right\} > 0.\end{aligned}$$

That is, we can get from any $x \in \mathbb{S}$ to any $y \in \mathbb{S}$ in one step (we can take $n(x, y) \equiv 1$ in Definition 6.4).

Step 2: write down the algorithm. Start from $X_0 = 1$ (say), and for $k = 1, \dots, n$ do

(a) Simulate $Y_k \sim U\{1, 2, ..., m\}$.

(b) Simulate $U_k \sim \mathcal{U}(0, 1)$ and if

$$U_k \quad \leq \quad \frac{Y_k}{X_{k-1}}$$

set $X_k = Y_k$, otherwise set $X_k = X_{k-1}$.

```
function imh_example(m=30, n=10_000)
    X = zeros(n); X[1] = 1
    for k = 2:n
        x = X[k-1]
        y = ceil(m*rand())  # y ~ U{1,2,...,m}
        if (rand() < y/x)
            X[k] = y
        else
            X[k] = x
        end
    end
    X
end
```

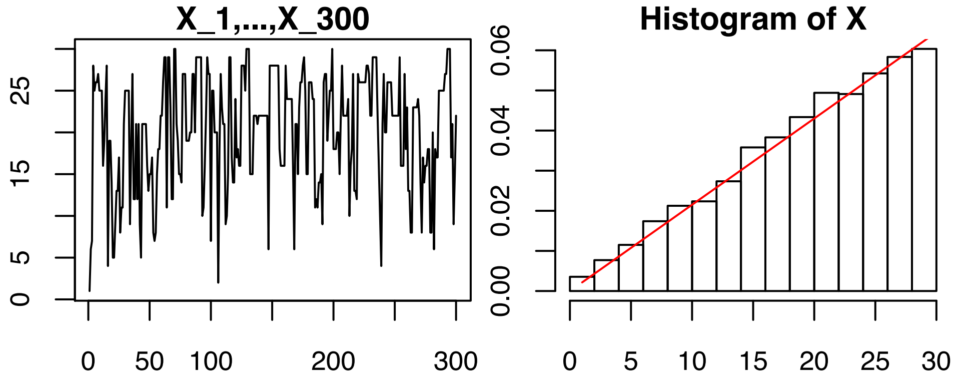

Figure 9: Left: x-axis is Markov chain step counter $k = 1, 2, \ldots, 300$ and y-axis is Markov chain state X_k. Right: histogram of $X_1, X_2, \ldots, X_n$ for $n = 10,000$ along with p.

Example 6.19 is an instance of the following class of Metropolis-Hastings algorithms.

Definition 6.20. Metropolis-Hastings algorithm with $q(x, y) = q(y)$, that is, proposal is independent of current state, is referred to as *independence sampler* or *independent Metropolis-Hastings* (IMH).

The IMH acceptance probability takes the form

$$\alpha(x, y) = \min\left\{1, \frac{p(y)q(x)}{p(x)q(y)}\right\} = \min\left\{1, \frac{w(y)}{w(x)}\right\},$$

where $w(x) = p(x)/q(x)$ for $q(x) > 0$. In order the IMH to be irreducible, we need $q(x) = 0$ implies $p(x) = 0$.

Remark 6.21. (Self-normalised) importance sampling can always be used instead of the IMH.

6.4 The Metropolis-Hastings algorithm on $\mathbb{X} = \mathbb{R}^d$

The Metropolis-Hastings (Algorithm 6.13) generalises directly to continuous setting, that is, $\mathbb{X} = \mathbb{R}^d$:

(i) p is a probability density on $\mathbb{R}^d$.

(ii) $q(x, \cdot\,)$ is a probability density on $\mathbb{R}^d$ for each $x \in \mathbb{R}^d$.

Everything else in Algorithm 6.13 remains unchanged.

Fact 6.22. The MH algorithm defines a Markov chain on $\mathbb{S} := \{x \in \mathbb{R}^d : p(x) > 0\}$. The transition probability K can be written as

$$\mathbb{P}(X_n \in A \mid X_{n-1} = x) =: K(x, A) = \int_A k(x, y)\mathrm{d}y + \rho(x)\mathbf{1}\left(x \in A\right), \qquad (14)$$

where $k(x, y) := q(x, y)\alpha(x, y)$ is a *sub-probability density* for each $x \in \mathbb{X}$ and $\rho(x) = 1 - \int k(x, y)\mathrm{d}y$ is the probability of rejection.

Precise definition of Markov chains on $\mathbb{S} \subset \mathbb{R}^d$ will be out of the scope of the course, but we shall see how the necessary ingredients are defined in this case. The article [17] by Nummelin contains a minimal self-contained proofs about the strong law of large numbers and more.

Definition 6.23. The $\mathbb{R}^d$-valued Markov chain is $(X_k)_{k\geq 1}$ is p-reversible, if $X_0 \sim p$ then $(X_0, X_1) \overset{d}{=} (X_1, X_0)$. That is, $\mathbb{P}(X_0 \in A, X_1 \in B) = \mathbb{P}(X_0 \in B, X_1 \in A)$.

Proposition 6.24. *Markov transition probability defined as in* (14) *is reversible with respect to a p.d.f. p on $\mathbb{X}$ if*

$$p(x)k(x,y) = p(y)k(y,x) \qquad \textit{for all } x, y \in \mathbb{X}. \tag{15}$$

The condition (15), sometimes also called as detailed balance, is essentially equivalent[9] with reversibility with transition probabilities of the form (14). This is identical to the definition of reversibility in the discrete case for $x \neq y$, which turns out to be sufficient. The proof of reversibility of Metropolis-Hastings is identical to the discrete case.

The irreducibility condition in the continuous case is likewise slightly different, as there is zero probability of reaching any single state from other states. Rather, any set of positive p-probability have to be reachable from everywhere.

Definition 6.25 (p-irreducibility)**.** Suppose that p is a p.d.f. on $\mathbb{S}$. The Markov chain $X_0, X_1, \ldots$ if p-irreducible if for any $x \in \mathbb{S}$ and any set $A \subset \mathbb{S}$ such that $\int_A p(y)\mathrm{d}y > 0$, there exists $n = n(x, A) < \infty$ such that

$$\mathbb{P}(X_n \in A \mid X_0 = x) > 0.$$

The proposal densities $q(x, y)$ are chosen to satisfy this condition.

We state the following general consistency theorem without proof[10]

Theorem 6.26. *If the Metropolis-Hastings algorithm is p-irreducible, then for any function f with $\mathbb{E}_p|f(X)| < \infty$, the MH-estimate is (strongly) consistent*

$$I^{(n)}_{p,q,\mathrm{MH}}(f) = \frac{1}{n}\sum_{k=1}^{n} f(X_k) \xrightarrow{n\to\infty} \mathbb{E}_p[f(X)] \qquad \textit{(almost surely).}$$

Note that Theorem 6.26 holds both when $\mathbb{X}$ is discrete or when $\mathbb{X} = \mathbb{R}^d$.

Example 6.27. Suppose want to simulate the standard normal distribution $X \sim N(0, 1)$. The target density is

$$p(x) \propto p_u(x) = \exp(-x^2/2).$$

Step 1: Choose the proposal distribution. We need something simple that can 'take us everywhere' (for irreducibility). Fix a constant $a > 0$ and choose

[9] To be precise, the continuous part $k(x, y)$ in the representation of (14) is unique only up to a set of measure zero. So the statement would be 'there exists a k such that...'.

[10] The proof follows, for example, from Corollary 2 of Tierney [26] along with Theorem 17.0.1 of Meyn and Tweedie [14]

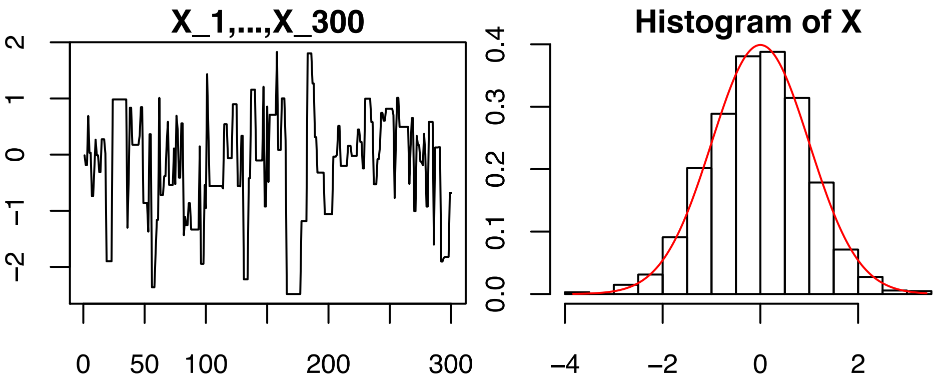

Figure 10: Simulation of Example 6.27: MCMC samples (left) and histogram approximation of the theoretical density (right).

a new point uniformly at random in a window of length $2a$ centred at x. The proposal density is

$$q(x,y) = \frac{1}{2a}\mathbf{1}\left(x-a<y<x+a\right).$$

Notice that $q(x,y) = q(y,x)$; this simplifies the acceptance probability

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

Step 2: Write the MCMC algorithm. Start from $X_0 = 0$ (say), and iterate for $k = 1,\dots,n$:

(a) Simulate $Z_k \sim \mathcal{U}(-a,a)$ and set $Y_k = X_{k-1} + Z_k$.

(b) Simulate $U_k \sim \mathcal{U}(0,1)$ and set

$$X_k = \begin{cases} Y_k, & \text{if } U_k \le \exp\big(r(X_{k-1},Y_k)\big), \\ X_{k-1}, & \text{otherwise.} \end{cases}$$

where $r(x,y) = \log p_u(y) - \log p_u(x) = -y^2/2 + x^2/2$.

```
function rwm_example(a=3, n=10_000)
    X = zeros(n); x = 0; L_px = -.5*x^2
    for k = 1:n
        y = x + (2rand()-1)*a
        L_py = -.5*y^2      # NB L_px calculated only once!
        if (rand() < exp(L_py-L_px))
            x = y; L_px = L_py
        end
        X[k] = x
    end
    X
end
```

Example 6.27 belongs to the following class of Metropolis-Hastings algorithms.

Definition 6.28. If $q(x,y) = q(y,x)$ for all $x, y \in \mathbb{X}$, then $\alpha(x,y) = \min\{1, p(y)/p(x)\}$. Such an algorithm is often called a *Metropolis* algorithm. More specifically, in a *symmetric random walk Metropolis* algorithm

$$Y_n = X_{n-1} + Z_n, \qquad Z_n \sim \tilde{q},$$

where the increment density $\tilde{q}$ is symmetric: $\tilde{q}(z) = \tilde{q}(-z)$ for all $z \in \mathbb{R}^d$.

The symmetricity of $\tilde{q}$ implies $q(x,y) = \tilde{q}(y-x) = \tilde{q}(x-y) = q(y,x)$. It is common to take $\tilde{q}$ to be density of $N(0,\Sigma)$, which implies that $Y_n \mid (X_{n-1} = x) \sim N(x, \Sigma)$.

Example 6.29 (Bivariate distribution with Gaussian random walk Metropolis).

$$\log p_u(x) = -\frac{1}{2} y(x)^T \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}^{-1} y(x), \qquad \text{where} \qquad y(x) = \begin{pmatrix} a^{-1}x_1 \\ ax_2 + ab(x_1^2 + a^2) \end{pmatrix},$$

and with $a = b = 1$.

For a proposal distribution q we want something simple to sample. Let's try bivariate standard normal, that is,

$$Y_k = X_{k-1} + Z_k, \qquad Z_k \sim N(0, I_2).$$

Note that this is symmetric random walk Metropolis algorithm. We choose to start from $x_0 = (0,0)^T$.

```
using Distributions
function log_p(x; a=1, b=1) # Log-pdf of a 'banana-shaped' distribution
  y = [x[1]/a, x[2]*a + a*b*(x[1]^2 + a^2)]
  logpdf(MvNormal([1 0.9; 0.9 1]), y)
end
function metropolis(n=10_000, d=2, log_p=log_p)
    X = zeros(d,n); x = zeros(d); px = log_p(x)
    for k = 1:n
        y = x + randn(2); py = log_p(y)  # Proposal & its density value
        if rand() < exp(py-px)
            x = y; px = py               # Accept
        end
        X[:,k] = x                       # Save output
    end
    X
end
```

6.5 On tuning of random-walk Metropolis (*)

Suppose that $\hat{q}$ is some symmetric distribution, that is, $\hat{q}(z) = \hat{q}(-z)$, and let $L \in \mathbb{R}^{d \times d}$ be an invertible matrix. If the proposals Y_k are formed as follows

$$Y_k = X_{k-1} + L\hat{Z}_k, \qquad \hat{Z}_k \sim \hat{q}.$$

The question is how the proposal 'shape/size' L should be chosen so that the algorithm would be 'efficient'.

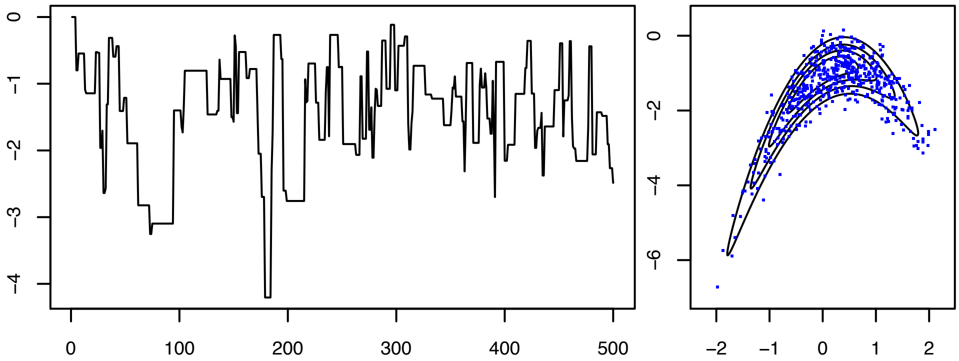

Figure 11: Simulation of Example 6.29: Second coordinate of the MCMC (left); The samples and the density contours (right).

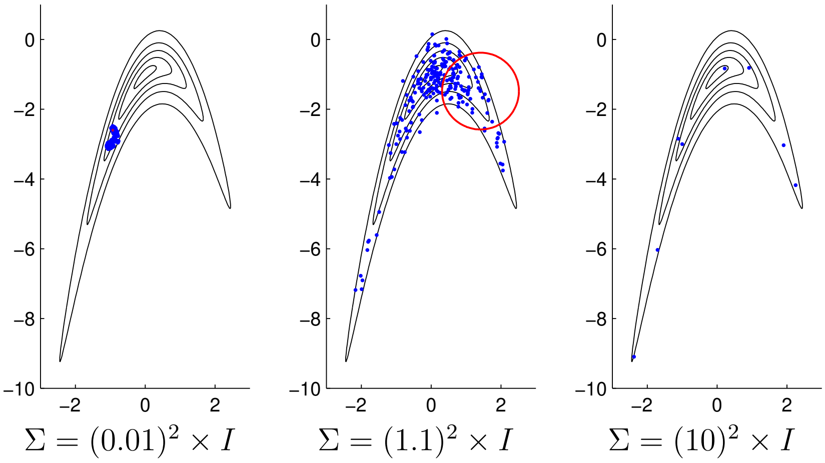

Figure 12: 1000 samples of the random walk Metropolis algorithm in $\mathbb{R}^2$ with $\tilde{q} = N(0, \Sigma)$. Contours of 'banana-shaped' p are shown in black.

Remark 6.30. There are some theoretical optimality results determining which L is 'the best' when $\hat{q}$ is standard normal [e.g. 24].

(a) First rule of thumb: Set $LL^T \approx \theta \mathrm{Cov}(p)$ where $\theta \in (0, \infty)$ is a scaling parameter.

(b) Second rule of thumb: Set θ such that around 25% of the samples should be accepted on average.[11]

These heuristics are often useful when p is (close to) unimodal.

Because $\mathrm{Cov}(p)$ is usually not available, $\mathrm{Cov}(p)$ is often estimated by a 'trial' MCMC targetting p, and θ is found also by trial and error.

Remark 6.31. There are various *adaptive MCMC* algorithms which can be used to automatise this process, and learn L 'progressively' [e.g. 10, 2]. Such methods have been observed to work well in practice, but the theoretical results ensuring the validity of the methods require subtle technical conditions.

Example 6.32. Implementation of an adaptive MCMC which finds 'good' L automatically [28].

```
using LinearAlgebra
function ram_adapt!(C, z, k, acc; gam=0.66, acc_opt=0.234)
  nz = norm(z); u = nz>0 ? z/nz : 0*z; step = (k+1)^(-gam); fact = acc-acc_opt
  dx = sqrt(step*abs(fact))*(C.L * (z/nz))
  if fact >= 0 lowrankupdate!(C, dx) else lowrankdowndate!(C, dx) end
end
function adapt_mcmc(log_p, x0, n)
  d = length(x0);  x = x0;  p_x = log_p(x); C = cholesky(diagm(ones(d)))
  X = zeros(d, n); acc = 0; z = zeros(d)
  for k = 1:n
      z = randn(d); y = x + C.L * z                # Proposal
      p_y = log_p(y); alpha = min(1, exp(p_y-p_x)) # Acc.prob.
      if (rand() <= alpha)
          x = y; p_x = p_y; acc += 1
      end
      X[:,k] = x
      ram_adapt!(C, z, k, alpha)     # Adapt the proposal covariance
  end
  (X=X, L=C.L, acc_rate=acc/n)
end
```

6.6 Componentwise updates

In higher dimensions, it is often difficult to design efficient proposal distributions $q(x, y)$. Instead, it is easier to design rules to update a *single coordinate* or a *block of coordinates* in each iteration.

In order to consider such updates, consider $\mathbb{X}$ to be d-dimensional, $\mathbb{X} = \mathbb{X}_1^d$; for instance, $\mathbb{X} = \mathbb{Z}^d$ or $\mathbb{X} = \mathbb{R}^d$. Let us introduce the following shorthand notation

$$x^{(-i)} := (x^{(1)}, \ldots, x^{(i-1)}, x^{(i+1)}, \ldots, x^{(d)})$$

for the vector $x \in \mathbb{X}$ with i:th coordinate omitted.

[11] The theoretical value, 0.234, is optimal in high dimensions under very strong assumptions.