SOLUTIONS 8

Solution 8.1. This is a straightforward adaptation of the proofs we have seen already. Let us recall that for $p_\star(x) = \mathcal{N}(x; \mu, \sigma^2)$, the unadjusted Langevin algorithm of the form given in the exercise takes the form

$$X_{n+1} = X_n - \gamma \frac{X_n - \mu}{\sigma^2} + \sqrt{2\gamma\beta^{-1}} W_{n+1}.$$

Let

$$a = 1 - \frac{\gamma}{\sigma^2}, \quad b = \frac{\gamma}{\sigma^2}\mu.$$

The algorithm then takes the form

$$X_{n+1} = aX_n + b + \sqrt{2\gamma\beta^{-1}} W_{n+1}.$$

We can write now the iterates beginning at x_0 as

$$\begin{aligned}
x_1 &= ax_0 + b + \sqrt{2\gamma\beta^{-1}} W_1, \\
x_2 &= \underbrace{a^2 x_0 + ab + a\sqrt{2\gamma\beta^{-1}} W_1}_{ax_1} + b + \sqrt{2\gamma\beta^{-1}} W_2, \\
x_3 &= \underbrace{a^3 x_0 + a^2 b + a^2\sqrt{2\gamma\beta^{-1}} W_1 + ab + a\sqrt{2\gamma\beta^{-1}} W_2}_{ax_2} + b + \sqrt{2\gamma\beta^{-1}} W_3, \\
&\vdots \\
x_n &= a^n x_0 + \sum_{k=0}^{n-1} a^k b + \sum_{k=0}^{n-1} a^k \sqrt{2\gamma\beta^{-1}} W_{n-k}.
\end{aligned}$$

We can compute the expected value

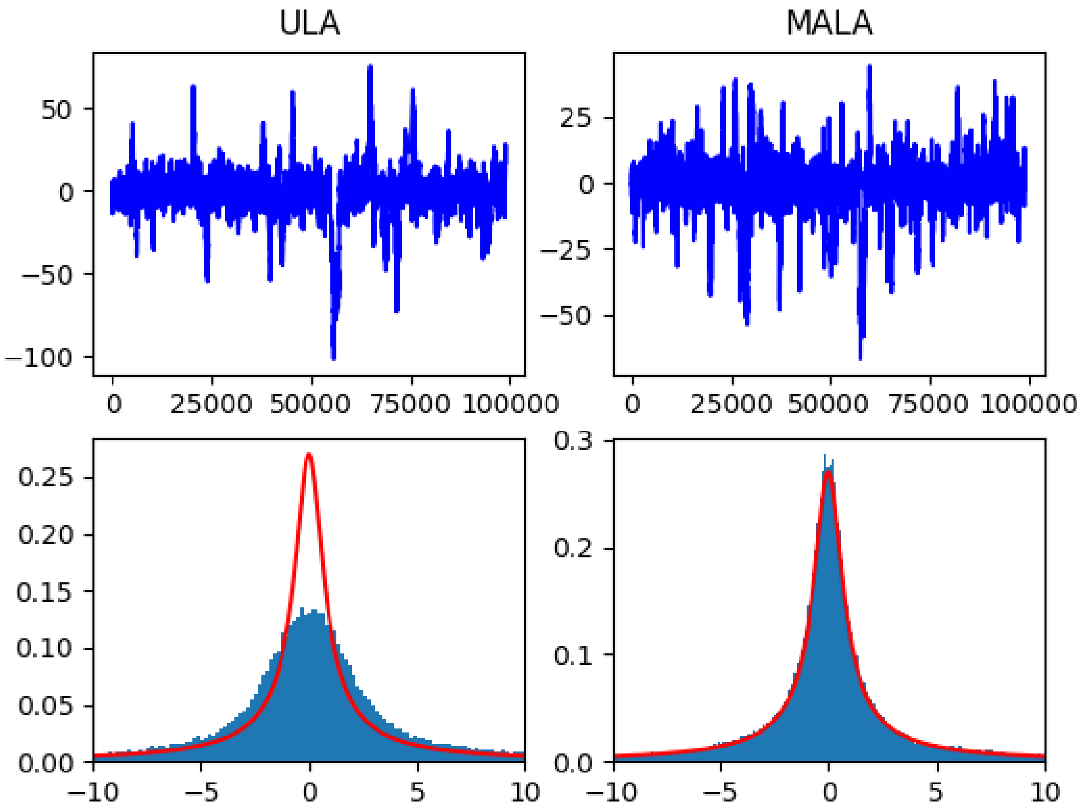
$$\mathbb{E}[X_n] = a^n x_0 + \sum_{k=0}^{n-1} a^k b,$$

since W_k are zero mean. As $n \to \infty$, we have

$$\mu_\infty = \lim_{n\to\infty} \mathbb{E}[X_n] = \sum_{k=0}^{\infty} a^k b = \frac{b}{1-a} = \mu.$$

since $0 < a < 1$. The variance of the iterates as $n \to \infty$ can also be computed. Note that for finite n, we have

$$\begin{aligned}
\mathsf{var}(x_n) &= \mathsf{var}\left(\sum_{k=0}^{n-1} a^k \sqrt{2\gamma\beta^{-1}} W_k\right), \\
&= 2\gamma\beta^{-1} \sum_{k=0}^{n-1} (a^2)^k, \\
&= 2\gamma\beta^{-1} \frac{1 - a^{2n}}{1 - a^2}.
\end{aligned}$$



Student's t samples for ULA and MALA

Therefore, we obtain the limiting variance as

$$
\begin{aligned}
\lim_{n\to\infty} \mathsf{var}(x_n) &= 2\gamma\beta^{-1}\frac{1}{1-a^2} \\
&= 2\gamma\beta^{-1}\frac{1}{1-\left(1-\frac{\gamma}{\sigma^2}\right)^2} \\
&= 2\gamma\beta^{-1}\frac{1}{\frac{2\gamma}{\sigma^2}-\frac{\gamma^2}{\sigma^4}}, \\
&= \frac{2\sigma^4}{\beta(2\sigma^2-\gamma)}.
\end{aligned}
$$

Therefore, we obtained the target measure of ULA as

$$
p_\star^{\gamma,\beta}(x) = \mathcal{N}\left(x; \mu, \frac{2\sigma^4}{\beta(2\sigma^2-\gamma)}\right),
$$

which is different than $p_\star$. The target will concentrate on μ as $\beta \to \infty$.

Solution 8.2. The code is given below. Note that we have implemented the unadjusted Langevin algorithm and the Metropolis-adjusted Langevin algorithm for the Student's t-distribution. For this, we need to derive the gradient $\nabla \log p(x)$, which in this case is given as

$$\nabla \log p(x) = -\frac{x(\nu+1)}{\nu+x^2}.$$

Using this, we can code both methods. The code is given below.

The ULA can be unstable and underperform depending on the context. For example, in this context, when $\nu = 0.5$, it does not work well. However, MALA works well for this specific value of $\nu = 0.5$. See Figure for reference.

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.special as sp

rng = np.random.default_rng(1234)

def grad_log_student_t(x, nu):
    return - (x * (nu + 1)) / (nu + x**2)

def log_student_t(x, nu):
    return - ((nu + 1) / 2) * np.log(1 + x**2 / nu)

def student_t(x, nu):
    return sp.gamma((nu + 1) / 2) / (sp.gamma(nu / 2) * np.sqrt(nu * np.pi)) * (1 + x**2 / nu)**(-(nu + 1) / 2)

T = 100000
burnin = 1000

nu = 0.5

gam = 1

x_ula = np.zeros(T)
x_ula[0] = 0
x_mala = np.zeros(T)
x_mala[0] = 0

def log_MALA_kernel(x_prop, x, gam, nu):
    return - (x_prop - x - gam * grad_log_student_t(x, nu))**2 / (4 * gam)


acc = 0

for t in range(1, T):

    x_ula[t] = x_ula[t-1] + gam * grad_log_student_t(x_ula[t-1], nu) + np.sqrt(2*gam) * rng.normal(0, 1)

    u = np.random.rand()
    x_mala_prop = x_mala[t-1] + gam * grad_log_student_t(x_mala[t-1], nu) + np.sqrt(2*gam) * rng.normal(0, 1)

    log_alpha = log_MALA_kernel(x_mala[t-1], x_mala_prop, gam, nu) - log_MALA_kernel(x_mala_prop,
```

```
                                    x_mala[t-1], gam, nu) \
                + log_student_t(x_mala_prop, nu) - log_student_t(
                                                x_mala[t-1], nu)

    if np.log(u) < log_alpha:
        x_mala[t] = x_mala_prop
        acc += 1
    else:
        x_mala[t] = x_mala[t-1]

print("Acceptance rate: ", acc / T)

xx = np.linspace(-10, 10, 1000)

plt.figure()
plt.subplot(2, 2, 1)
plt.plot(x_ula[burnin:], 'b')
plt.title('ULA')
plt.subplot(2, 2, 2)
plt.plot(x_mala[burnin:], 'b')
plt.title('MALA')
plt.subplot(2, 2, 3)
plt.hist(x_ula[burnin:], bins=1000, density=True)
plt.plot(xx, student_t(xx, nu), 'r')
# set subplot limit to -5, 5 (xlim)
plt.xlim([-10, 10])
plt.subplot(2, 2, 4)
plt.hist(x_mala[burnin:], bins=1000, density=True)
plt.plot(xx, student_t(xx, nu), 'r')
plt.xlim([-10, 10])
plt.show()
```

Solution 8.3. We summarise the solutions as follows.

1. Note that accesing only uniform random numbers in this setting, we need to use the inversion sampler for exponentials. Recall that then, we can draw
   - $X_k = -Y_{k-1}^{-1} \log(1 - U_k)$ where $U_k \sim \text{Unif}(0, 1)$
   - $Y_k = -X_k^{-1} \log(1 - U_k)$ where $U_k \sim \text{Unif}(0, 1)$

2. We need to derive that the joint $p(x, y) \propto \exp(-xy)$ is implied by the full conditionals in the form we had.

   In order to do this, we will first derive a marginal:

$$p(x) = \int p(x, y)\mathrm{d}y = \int \exp(-xy)\mathrm{d}y = \frac{1}{x}\exp(-xy)\Big|_{y=0}^{y=\infty} = \frac{1}{x}.$$

   Since we have

$$p(y|x) = \frac{p(x, y)}{p(x)},$$

   we can see that using $p(x|y)p(x) = p(x, y)$ implies that $p(x|y) \propto x\exp(-xy)$ which is the exponential density. Similar argument holds for $p(x|y)$ which means that the Gibbs sampler would target $p(x, y) \propto \exp(-xy)$.

3. Try to see whether the unnormalised density is integrable or not, as this is the condition for the density to exist. In other words, we should check

$$\int_{\mathbb{R}^2} \exp(-xy)\mathrm{d}x\mathrm{d}y < \infty.$$

But we will see that this is not the case. To see this, notice

$$\int_{\mathbb{R}^2_+} \exp(-xy)\mathrm{d}x\mathrm{d}y = \int_{\mathbb{R}_+} \frac{1}{y}\mathrm{d}y = \infty$$

This means that Gibbs sampler can target a measure that is not a valid density. This is why it is important to check the implied joint from given full conditionals to determine whether the Gibbs sampler is doing something valid.

4. Let us assume full conditionals are truncated to $[0, 1]$. Then we can refine them, i.e.,

$$p(x|y) = \frac{x\exp(-xy)}{\int_0^1 x\exp(-xy)\mathrm{d}y} = \frac{x\exp(-xy)}{1 - e^{-x}}.$$

Similar argument goes again for $p(y|x)$. There is one problem remaining that is how to sample from $p(x|y)$ of this form.

One can derive the conditional CDF (show this)

$$F_{Y|X}(y|x) = \frac{1 - e^{-xy}}{1 - e^{-x}}.$$

Then, we can sample from this conditional CDF using the inverse CDF method. In particular, we can sample from $p(x|y)$ using (derive the inverse and show this)

$$Y = -\frac{1}{x}\log(1 - U(1 - e^{-x})).$$

Solution 8.4. Note that for rejection sampling, we have

$$\alpha_{\text{rejection}}(x) = \frac{p(x)}{Mq(x)},$$

where

$$M = \sup_z \frac{p(z)}{q(z)}.$$

Let us rewrite this:

$$\alpha_{\text{rejection}}(x) = \frac{p(x)/q(x)}{\sup_z p(z)/q(z)}.$$

On the other hand, for independent MH, we have

$$\alpha_{\text{MH}}(x) = \min\left\{1, \frac{p(x)q(x')}{p(x')q(x)}\right\}.$$

First of all, $\alpha_{\text{rejection}}(x) \leq 1$. Therefore, we just only need to check the ratio. To see this, see that

$$\frac{p(x)/q(x)}{p(x')/q(x')} \geq \frac{p(x)/q(x)}{\sup_z p(z)/q(z)},$$

as $\sup$ is always a bigger number.