SOLUTIONS TO EXERCISES 5

Solution 5.1. Monte Carlo sampler is just sampling $\mathcal{N}(0,1)$ and keeping samples $X > 4$ as we explained several times – the IS estimator is also provided in lecture notes. Below is one solution. Here the proposal is

$$q(x) = \mathcal{N}(x; 6, 1).$$

```python
import numpy as np
import matplotlib.pyplot as plt

xx = np.linspace(4, 20, 100000)

def p(x):
    return np.exp(-x**2/2)/np.sqrt(2*np.pi)

def q(x, mu, sigma):
    return np.exp(-(x-mu)**2/(2*sigma**2))/(np.sqrt(2*np.pi)*sigma)

def w(x, mu, sigma):
    return p(x)/q(x, mu, sigma)

I = np.trapz(p(xx), xx) # Numerical computation of the integral

print('Integral of p(x) from 4 to infinity: ', I)

N = 10000

x = np.random.normal(0, 1, N) # iid samples from p(x)

I_est_MC = (1/N) * np.sum(x > 4)
print('Monte Carlo estimate: ', I_est_MC)

mu = 6
sigma = 1

x_s = np.zeros(N)
weights = np.zeros(N)

for i in range(N):
    x_s[i] = np.random.normal(mu, sigma, 1)
    weights[i] = w(x_s[i], mu, sigma)

I_est_IS = (1/N) * np.sum(weights * (x_s > 4))
print('Importance sampling estimate: ', I_est_IS)
```

Solution 5.2. This solution allows us to find the minimum variance (optimal) proposal due to its (tautological) structure. But it is good for practice. If you notice that test function also looks like an exponential density, the problem here is to just find the resulting exponential parameter from the multiplication $p(x)\varphi(x)$ - which is trivial. But we will treat the problem as if we did not notice this!

Recall from lecture notes (Remark 4.5) that we have the variance expression

$$\mathsf{var}_q(\hat{\varphi}^N_{\text{IS}}) = \frac{1}{N}\left(\mathbb{E}_{q_\mu}\left[w^2(X)\varphi^2(X)\right] - \bar{\varphi}^2\right).$$

OF course in this expression, note that $w(x)$ also depends on μ as it is the ratio. Now we would like to minimise this expression w.r.t. μ – which means that we will only deal with the first term (as $\bar{\varphi}$ is a constant). Specifically, we can also ignore $1/N$ as it won't change the result. Next, we write the integral

$$\begin{aligned}
\mathbb{E}_{q_\mu}\left[w^2(X)\varphi^2(X)\right] &= \int \frac{p^2(x)}{q_\mu^2(x)}\varphi^2(x)q_\mu(x)\mathrm{d}x, \\
&= \int \frac{p^2(x)}{q_\mu(x)}\varphi^2(x)\mathrm{d}x, \\
&= \int \frac{e^{-2x}}{\mu e^{-\mu x}}e^{-Kx}\mathrm{d}x, \\
&= \frac{1}{\mu}\int e^{(\mu-2-K)x}\mathrm{d}x, \\
&= \frac{1}{\mu(K+2-\mu)}\int (K+2-\mu)e^{(\mu-2-K)x}\mathrm{d}x, \\
&= \frac{1}{\mu(K+2-\mu)},
\end{aligned}$$

where the last integral is just one as it is the integral of the exponential density. Note that, for this integral to be finite, we impose a condition[2]

$$\mu - 2 - K < 0 \implies \mu < K + 2.$$

The minimum variance proposal then can be found using

$$\begin{aligned}
\mu_\star &= \arg\min_\mu \mathbb{E}_{q_\mu}\left[w^2(X)\varphi^2(X)\right] \\
&= \arg\min_\mu \frac{1}{\mu(K+2-\mu)}.
\end{aligned}$$

We take the derivative of the log of this expression and setting it to zero (check and let me know if there is a mistake)

$$\frac{1}{\mu} = \frac{1}{K+2-\mu},$$

which implies that

$$\mu = \frac{K}{2} + 1.$$

So what is variance? Plug this back in the expression:

$$\left.\frac{1}{\mu(K+2-\mu)}\right|_{\mu=\frac{K}{2}+1} = \frac{1}{\left(\frac{K}{2}+1\right)^2}.$$

Note that this is the first part of the variance expression with $\mu_\star$ plugged it in, we also need the second part to observe the variance reduction:

$$\mathsf{var}_q(\hat{\varphi}_{\mathsf{IS}}^N) = \frac{1}{N}\left(\mathbb{E}_{q_\mu}\left[w^2(X)\varphi^2(X)\right] - \bar{\varphi}^2\right).$$

[2] You should look at the exponents of exponentials appearing in these integrals and impose conditions so they don't grow to infinity with x.

So what is $\bar{\varphi}$? It is just the integral

$$\begin{aligned}\int e^{-\frac{K}{2}x} e^{-x} \mathrm{d}x &= \int e^{-\left(\frac{K}{2}+1\right)x} \mathrm{d}x \\ &= \frac{1}{\frac{K}{2}+1}\end{aligned}$$

which we can find by matching the expression inside the integral with an exponential density with rate $\frac{K}{2}+1$. So

$$\bar{\varphi}^2 = \frac{1}{\left(\frac{K}{2}+1\right)^2},$$

then,

$$\mathsf{var}_q(\hat{\varphi}_{\mathrm{IS}}^N) = \frac{1}{N}\left(\mathbb{E}_{q_{\mu_\star}}\left[w_{\mu_\star}^2(X)\varphi^2(X)\right] - \bar{\varphi}^2\right) = 0!$$

What happened, how did we end up with a zero variance proposal? As I said in the beginning, we did something trivial: Note that $\varphi(x)p(x)$ is just another exponential density. You just found corresponding parameter that matches q_μ to $\varphi(x)p(x)$ via a painful procedure. However, (i) it is good practice, (ii) it told you that it is a good idea to choose proposals similar to $\varphi(x)p(x)$.

Solution 5.3. This exercise will show something rather remarkable: By choosing a minimum variance proposal, you can obtain Gaussian samples with lower variance than i.i.d sample. In other words, sampling directly from $\mathcal{N}(0,1)$ to estimate the mean of the density is not optimal from variance minimisation perspective. Note that this trivially generalises to $\mathcal{N}(\mu,\sigma^2)$, we avoid this to not clutter the notation.

As stated in the exercise, we first show that the variance of the standard MC estimator is $1/N$. We write as given in the lecture notes

$$\mathsf{var}_p(\hat{\varphi}_{\mathrm{MC}}^N) = \frac{1}{N}\mathsf{var}_p(\varphi).$$

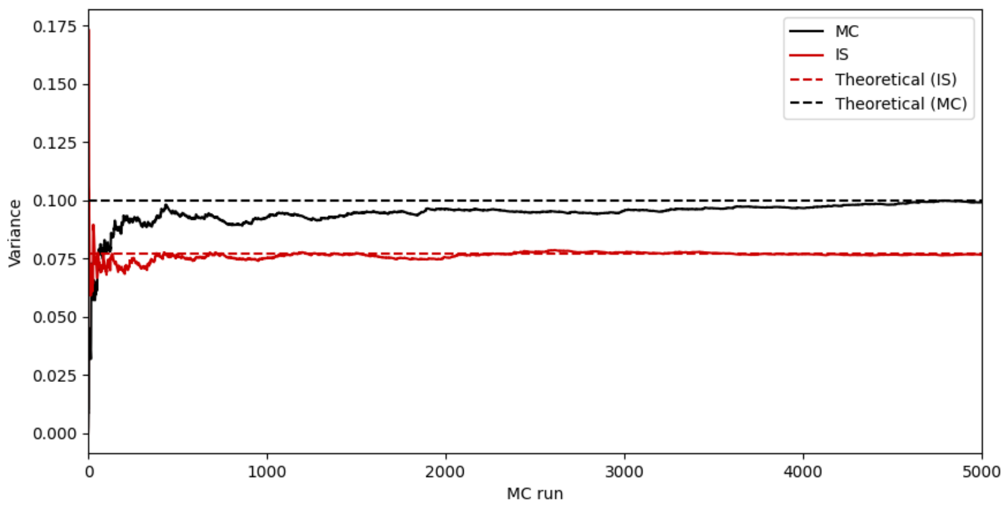
The second term here is exactly the variance of $\mathcal{N}(0,1)$ which is 1, hence the result.

Now we will do a similar thing to the last exercise to choose a minimum variance proposal. Note that in the last exercise, we showed that, we have to solve the problem

$$\lambda_\star = \arg\min_\lambda \mathbb{E}_{q_\lambda}\left[w_\lambda^2(X)\varphi^2(X)\right],$$

again $w(x)$ depends λ as it involves q_λ. Note that $\varphi(x) = x$ since we try to estimate the mean. We write the expectation explicitly

$$\begin{aligned}\mathbb{E}_{q_\lambda}\left[w_\lambda^2(X)\varphi^2(X)\right] &= \int \frac{p^2(x)}{q_\lambda(x)}\varphi^2(x)\mathrm{d}x \\ &= \int \frac{\frac{1}{2\pi}e^{-x^2}}{\frac{1}{\sqrt{2\pi(1/\lambda)}}e^{-\frac{\lambda x^2}{2}}} x^2 \mathrm{d}x \\ &= \frac{\sqrt{2\pi\lambda^{-1}}}{2\pi}\int x^2 e^{-(1-\frac{\lambda}{2})x^2}\mathrm{d}x.\end{aligned}$$



Empirical variances match theoretical variance

The last exponential in the integral can be seen as a zero mean Gaussian with appropriate variance. To see this, we rewrite it

$$\mathbb{E}_{q_\lambda}\left[w_\lambda^2(X)\varphi^2(X)\right] = \frac{\sqrt{2\pi\lambda^{-1}}}{2\pi}\sqrt{2\pi\sigma^2}\int x^2 \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \mathrm{d}x.$$

where it can be seen that

$$\frac{1}{2\sigma^2} = 1 - \frac{\lambda}{2}$$

which implies that

$$\sigma^2 = (2-\lambda)^{-1}.$$

Therefore, we finally obtain (by seeing that the last integral is now just σ^2):

$$\mathbb{E}_{q_\lambda}\left[w_\lambda^2(X)\varphi^2(X)\right] = \lambda^{-1/2}(2-\lambda)^{-3/2}.$$

Optimising this value (check this) gives us

$$\lambda_\star = 1/2.$$

Note that $\bar{\varphi} = 0$, so computing this value at the variance expression gives us (again, fill this gap)

$$\mathsf{var}_q(\hat{\varphi}_{\mathrm{IS}}^N) = \frac{1}{N}\left\{\lambda^{-1/2}(2-\lambda)^{-3/2}\right\}\Big|_{\lambda=1/2} = \frac{0.7698}{N}.$$

So we've got an estimator which has lower variance than the i.i.d estimator! Note that this is not a theoretical exercise as you can also numerically show this. Let us plot for $N = 10$, empirical variance estimates of two estimators – see the plot above. The code is attached.

```
import numpy as np
import matplotlib.pyplot as plt

```

```python
def p(x):
    return 1/np.sqrt(2*np.pi) * np.exp(-x**2/2)

def q(x, lam):
    return 1/np.sqrt(2*np.pi * 1/lam) * np.exp(-x**2/(2/lam))

lam = 1/2

N = 10

I_MC = np.array([])
I_IS = np.array([])
var_MC = np.array([])
var_IS = np.array([])

var_th = (1/N) * (2 - lam)**(-3/2) * lam**(-1/2)

MC = 5000

fig, ax = plt.subplots(1, 1, figsize=(10, 5))

for i in range(MC):
    x = np.random.normal(0, 1, N)
    I_MC = np.append(I_MC, np.mean(x))

    x_s = np.sqrt(1/lam) * np.random.normal(0, 1, N)
    weights = p(x_s) / q(x_s, lam)

    I_IS = np.append(I_IS, (1/N) * np.sum(weights * x_s))

    var_MC = np.append(var_MC, np.var(I_MC))
    var_IS = np.append(var_IS, np.var(I_IS))
    if (i+1) % 100 == 0:
        print('MC variance: ', var_MC[-1])
        print('IS variance: ', var_IS[-1])
        print('Theoretical variance: ', var_th)
        print('MC mean: ', np.mean(I_MC))
        print('IS mean: ', np.mean(I_IS))
        ax.cla()
        ax.plot(var_MC, label='MC', color='k')
        ax.plot(var_IS, label='IS', color=[0.8, 0, 0])
        ax.plot([0, MC], [var_th, var_th], '--', label='Theoretical (
                                          IS)', color=[0.8, 0, 0])
        ax.plot([0, MC], [1/N, 1/N], 'k--', label='Theoretical (MC)')
        ax.set_xlim([0, MC])
        ax.set_xlabel('MC run')
        ax.set_ylabel('Variance')
        ax.legend()
        plt.show(block=False)
        plt.pause(0.01)


print('MC variance: ', var_MC[-1])
print('IS variance: ', var_IS[-1])
print('Theoretical variance: ', var_th)
print('MC mean: ', np.mean(I_MC))
print('IS mean: ', np.mean(I_IS))
ax.cla()
ax.plot(var_MC, label='MC', color='k')
ax.plot(var_IS, label='IS', color=[0.8, 0, 0])
ax.plot([0, MC], [var_th, var_th], '--', label='Theoretical (IS)',
                                    color=[0.8, 0, 0])
ax.plot([0, MC], [1/N, 1/N], 'k--', label='Theoretical (MC)')
ax.set_xlim([0, MC])
ax.set_xlabel('MC run')
ax.set_ylabel('Variance')
ax.legend()
plt.show()
```

Solution 5.4. We have a model

$$p(x) = \mathcal{N}(x; 0, 1)$$
$$p(y|x) = \mathcal{N}(y; x, 1),$$

and given the integral expression $p(y) = \int p(y|x)p(x)\mathrm{d}x$.

1. The integral is analytically computable and provided in, e.g., Example 3.11. The density is given as

   $$p(y) = \mathcal{N}(y; 0, 2).$$

   On my computer, I have

   $$p(y) \approx 4.52 \times 10^{-10}.$$

2. The standard estimator is given as follows. In the integral described in the question, we can identify that we have a test function

   $$\varphi(x) = p(y = 9|x),$$

   that depends on x (since $y = 9$ is fixed). The classical estimator is given as

   $$\varphi_{\text{MC}}^N = \frac{1}{N}\sum_{i=1}^N p(y = 9|X_i),$$

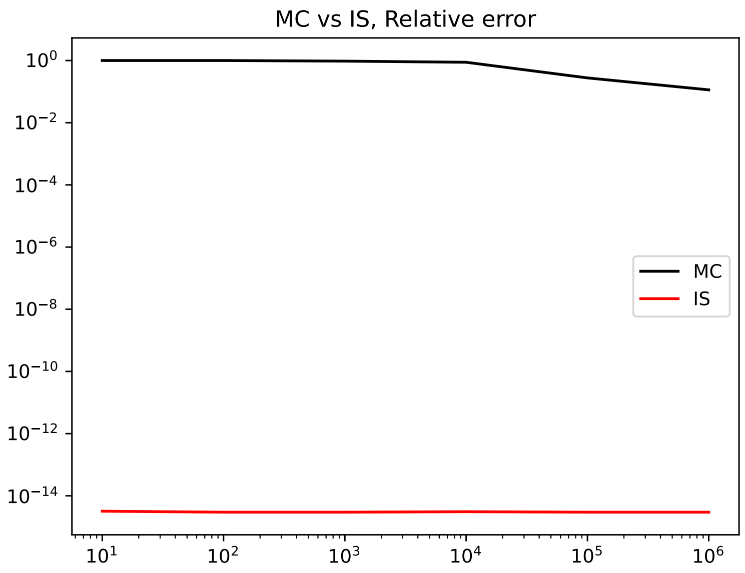
   where $X_i \sim p(x)$ for $i = 1, \ldots, N$. I will plot the graph below (both graphs together)

3. In order to minimise the variance of the estimator, we want to find

   $$\mu^\star = \arg\min_\mu \mathbb{E}_{q_\mu}[p^2(y_0|X)w_\mu^2(X)].$$

   This minimisation problem should be clearly stated in the answer. We next derive this expectation

   $$\mathbb{E}_{q_\mu}[p^2(y_0|X)w_\mu^2(X)] = \int p^2(y_0|x)\frac{p^2(x)}{q_\mu(x)}\mathrm{d}x = \int \frac{\frac{1}{2\pi}\exp\left(-(y_0 - x)^2\right)\frac{1}{2\pi}\exp(-x^2)}{\frac{1}{\sqrt{\pi}}\exp\left(-(x-\mu)^2\right)}\mathrm{d}x,$$
   $$= \frac{\sqrt{\pi}}{4\pi^2}\int \exp(-y_0^2 + 2y_0 x - x^2 - x^2 + x^2 - 2x\mu + \mu^2)\mathrm{d}x$$
   $$= \frac{\sqrt{\pi}}{4\pi^2}\int \exp(-y_0^2 + 2(y_0 - \mu)x - x^2 + \mu^2)\mathrm{d}x.$$



One can see that IS is much more efficient than MC in terms of RAE.

Let $g(\mu) = \mathbb{E}_{q_\mu}[p^2(y_0|X)w_\mu^2(X)]$ for a shorthand notation. We complete the square here to obtain a Gaussian density:

$$\begin{aligned} g(\mu) &= \frac{\sqrt{\pi}}{4\pi^2}\int \exp\left(-(y_0-\mu)^2\right)\exp\left(2(y_0-\mu)x\right)\exp(-x^2)\exp(2\mu^2)\exp(-2y_0\mu)\mathrm{d}x, \\ &= \frac{\pi}{4\pi^2}\exp\left(-2y_0\mu\right)\exp\left(2\mu^2\right)\int \mathcal{N}(x; y_0-\mu, 1/2)\mathrm{d}x, \\ &= \frac{1}{4\pi}\exp\left(-2y_0\mu\right)\exp(2\mu^2). \end{aligned}$$

We can easily maximise this by taking the log

$$\log g(\mu) = -\log\frac{1}{4\pi} - 2y_0\mu + 2\mu^2,$$

taking its derivative w.r.t. μ and setting it to zero gives

$$-2y_0 + 4\mu = 0,$$

which implies $\mu^\star = y_0/2$ as asked. This sampler puts mass of the proposal close to the observation point. If y_0 has very small probability under original model, the MC estimator could fail (as mentioned in part(a)), but the optimal IS estimator will sample close to the observation and will provide an accurate estimate.

4. The plot roughly looks like in the Figure. The MC error is expected to be much higher than the IS error.

We can provide the Python code as follows.

```python
import numpy as np
import matplotlib.pyplot as plt

rng = np.random.default_rng(41)
```

```python

def p(x):
    return 1/np.sqrt(2*np.pi) * np.exp(-x**2/2)

def q(x, mu, sigma):
    return 1/np.sqrt(2*np.pi * sigma**2) * np.exp(-(x-mu)**2/(2*sigma
                                            **2))

def likelihood(y, x):
    return p(y-x)

def marginal_likelihood(y):
    sigma = np.sqrt(2)
    return 1/np.sqrt(2*np.pi * sigma**2) * np.exp(-(y)**2/(2*sigma**2)
                                            )


y = 9
I = marginal_likelihood(y)
print(marginal_likelihood(y))

mu = y/2 # derived in the exercise

N_range = [10, 100, 1000, 10000, 100000, 1000000, 10000000]

fig = plt.figure()

var_IS_run = np.array([])
var_MC_run = np.array([])


I_MC = np.array([])
I_IS = np.array([])
var_IS = np.array([])
var_MC = np.array([])
K = np.array([])

for N in N_range:
    weights = np.array([])
    x = rng.normal(0, 1, N)
    I_MC = np.append(I_MC, np.mean(likelihood(y, x)))

    x_s = rng.normal(mu, np.sqrt(1/2), N)
    weights = p(x_s)/q(x_s, mu, np.sqrt(1/2))
    I_IS = np.append(I_IS, np.mean(weights * likelihood(y, x_s)))
    K = np.append(K, N)

print('True value: ', I)
print('IS value: ', I_IS)
print('MC value: ', I_MC)


plt.clf()
plt.loglog(K, np.abs((I_MC - I))/I, 'k-', label='MC')
plt.loglog(K, np.abs((I - I_IS))/I, 'r-', label='IS')
plt.legend()
plt.title('MC vs IS, Relative error')
plt.show()
```

Solution 5.5. The implementation of the log-trick is given below. The prints provide the output, the first one is NaN, while the second one is stable.

```python
import numpy as np
import matplotlib.pyplot as plt

logw = [1000, 1001, 999, 1002, 950]

w = np.exp(logw)/np.sum(np.exp(logw))
w2 = np.exp(logw - np.max(logw))/np.sum(np.exp(logw - np.max(logw)))

print(w)
print(w2)
```