EXERCISES 4

Exercise 4.1 (Conditional Bayes theorem)**.** Prove that, we can write the conditional version of the Bayes theorem

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

Note that **no conditional independence assumptions** are required for this. It basically says that the Bayes theorem goes through by conditioning all the distributions to a third random variable z.

Exercise 4.2. Using the definition of conditional independence and Bayes rule, prove that, if X and Y are conditionally independent given Z, then

$$p(x|y,z) = p(x|z)$$

and

$$p(y|x,z) = p(y|z).$$

In other words, if a random variable X is conditionally independent from Y given Z, then conditioning on Y and Z is no different than conditioning on just Z as Z breaks the 'link' between X and Y.

Exercise 4.3 (Curse of dimensionality for rejection samplers)**.** Assume $x \in \mathbb{R}^d$ and $p(x)$ and $q(x)$ are two d-dimensional probability distributions. If

$$p(x) = \prod_{i=1}^{d} p_0(x_i) \quad \text{and} \quad q(x) = \prod_{i=1}^{d} q_0(x_i)$$

where $x_i \in \mathbb{R}$ and

$$\sup_{x\in\mathbb{R}} \frac{p_0(x)}{q_0(x)} = K,$$

then find M in terms of K and prove that the acceptance rate in rejection sampling will go to zero as $d \to \infty$.

Apply this result to

$$\begin{aligned} p(x) &= \mathcal{N}(x; 0, \sigma_p^2 I_d) \\ q(x) &= \mathcal{N}(x; 0, \sigma_q^2 I_d), \end{aligned}$$

where $\sigma_q > \sigma_p$. Find the optimal M and show that the acceptance rate will approach zero as $d \to \infty$.

Exercise 4.4 (Estimator variance)**.** We have seen two ways to compute $\mathbb{P}(X > 2)$ for a Cauchy r.v. X with density

$$p(x) = \frac{1}{\pi(1+x^2)}.$$

We discussed in the lecture that

$$\mathbb{P}(|X| > 2) = 2I,$$

where $I = \mathbb{P}(X > 2)$. We can also use the fact that

$$\mathbb{P}(|X| > 2) = 1 - P(-2 < X < 2),$$

which is also

$$\mathbb{P}(|X| > 2) = 1 - 2\mathbb{P}(0 < X < 2)$$

due to symmetry (see slides or lecture notes). Then, we can write

$$\lambda = \mathbb{P}(0 < X < 2)$$

and obtain

$$I = \frac{1}{2} - \lambda.$$

- Estimate

$$\mathbb{P}(0 < X < 2) = \int_0^2 \frac{1}{\pi(1 + x^2)} \mathrm{d}x$$

  using Monte Carlo integration. Note that you can convert this into an expectation by choosing a suitable density on $[0, 2]$ (the simplest one works but be careful about constants).

- Determine your test function and compute the variance of this estimate as we have done in the class (using true value of this integral – can you deduce it since we know $\mathbb{P}(X > 2) = 0.1476$?)

- The variance of this estimate (of λ) would be the same as the variance of the main estimate I, why?

Exercise 4.5. Consider the following integral:

$$I = \int_0^1 (1 - x^2)^{1/2} \mathrm{d}x.$$

Take the sampling distribution as uniform on $[0, 1]$. Build the Monte Carlo estimate for varying N and compute the mean and the variance, i.e., for $\varphi(x) = (1 - x^2)^{1/2}$ and given $X_1, \ldots, X_N$ from $\mathrm{Unif}(0, 1)$, compute

$$\hat{\varphi}^N = \frac{1}{N} \sum_{i=1}^{N} \varphi(X_i) \tag{1}$$

and the empirical variance

$$\widehat{\mathsf{var}}[\hat{\varphi}^N] = \frac{1}{N^2} \sum_{i=1}^{N} (\varphi(X_i) - \hat{\varphi}^N)^2. \tag{2}$$

Discuss the difference between this empirical estimator vs. the correct one for the variance (using the true value). Plot your mean estimates (1), standard deviation estimates (by taking the square root of (2)), and the true value $I = \pi/4$. Can you always trust the variance estimates? Here is a code snippet for this one:

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

def phi(x):
    return np.sqrt((1 - x**2))

I = np.pi / 4 # true value

N_max = 10000 # go up to 10,000 samples

U = np.random.uniform(0, 1, N_max)
I_est = np.zeros(N_max - 1) # this is longer than we need
I_var = np.zeros(N_max - 1)

fig = plt.figure(figsize=(10, 5))

k = 0

K = np.array([])

# We are not computing for every N for efficiency

for N in range(1, N_max, 5):

    I_est[k] = # Your mean estimate here
    I_var[k] = # Your variance estimate here

    k = k + 1 # We index estimators with k as we jump N by 5
    K = np.append(K, N)

    if (N-1) % 200 == 0:
        plt.clf()
        plt.plot(K, I_est[0:k], 'k-', label='MC estimate')
        plt.plot(K, I_est[0:k] + np.sqrt(I_var[0:k]), 'r', label='$\
                                            sigma$', alpha=1)
        plt.plot(K, I_est[0:k] - np.sqrt(I_var[0:k]), 'r', alpha=1)
        plt.plot([0, N_max], [I, I], 'b--', label='True Value', alpha=
                                            1, linewidth=2)
        plt.legend()
        plt.xlabel('Number of samples')
        plt.ylabel('Estimate')
        plt.xlim([0, N_max])
        plt.show(block=False) # For animation
        plt.pause(0.1)
```