Homework 4

Question 1: Importance Sampling

Suppose X is a normal variable with mean μ and variance σ^2 . Let $f(x)$ be the density function of X. For a constant θ, define the tilted density function as

$$f_\theta(x) = \frac{f(x)e^{\theta x}}{M_\theta},$$

where M_θ is the normalization constant to ensure that $f_\theta(x)$ is a probability density function.

(a) Find the expression of M_θ in terms of $theta$, μ and σ.

$$M_\theta = \exp(\mu\theta + \sigma^2\theta^2/2).$$

(b) What kind is the tilted distribution corresponding to $f_\theta(x)$?

Since the tilted distribution has density function

$$f_\theta(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{1}{2\sigma^2}\left(x-(\mu+\sigma^2\theta)\right)^2\right),$$

it is a normal distribution with mean $\mu + \sigma^2\theta$ and variance σ^2.

(c) Suppose $\mu = 0$ and $\sigma = 1$. Our goal is to estimate $P(X > 10)$ which is a rare event probability. Design a importance sampling algorithm using the tilted distribution. Specify your choice of θ and explain why it leads to variance reduction.

Let $p = P(X > 10)$ and $\theta = 10$. Below is the importance sampling algorithm.

1. For $i = 1, \ldots, n$: simulate $Y_i \sim f_\theta$.
2. Output:

$$\hat{p} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{Y_i > 10\}\frac{f(Y_i)}{f_\theta(Y_i)} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{Y_i > 10\}\exp(50 - 10Y_i).$$

Then, the variance of $\hat{p}$ is

$$\begin{aligned} Var[\hat{p}] &= \frac{1}{n}E[\mathbf{1}\{Y > 10\}f(Y)^2/f_\theta(Y)^2] - p^2/n \\ &= \frac{1}{n}\int_{10}^{\infty} \exp(50 - 10x)f(x)dx - p^2/n \\ &< (p - p^2)/n \qquad (\exp(50 - 10x) < 1 \text{ when } x > 10), \end{aligned}$$

where $(p - p^2)/n$ is the variance of the naive MC estimator, thus importance sampling leads to variance reduction. **For students who define the importance sampling estimator as $\hat{p} = \mathbf{1}\{Y > 10\}f(Y)/f_\theta(Y)$, where $Y \sim f_\theta$, would get full scores as long as the variance reduction is justified properly.**

Question 2: Conditional Monte Carlo vs. Stratified Sampling

Suppose that Y is a binomial random variable with parameters $n = 10$ and $p = 0.5$. Suppose that, conditioned on $Y = y$, X is a normal random variable with mean y and variance 4. We want to use simulation to efficiently estimate $\theta = P(X \geq 5)$.

(a) Design a simulation algorithm that only generates i.i.d. random variables.

1. For $i = 1, \ldots, N$:
   A. Generate $U_j \overset{iid.}{\sim}$ Unif(0,1), $j = 1, \ldots, 10$.
   B. Compute $Y_i = \sum_{j=1}^{10} \mathbf{1}\{U_j > 0.5\}$
   C. Generate $Z_i \sim N(0, 1)$.
   D. Compute $X_i = Y_i + 2Z_i$.
2. Output:

$$\hat{\theta}_{NMC} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\{X_i \geq 5\}.$$

(b) Design a simulation algorithm that uses conditional Monte Carlo method. Clearly specify the estimator you are using. Calculate the variance of the conditional Monte Carlo estimator.

Since X is a normal random variable with mean y and variance 4 given that $Y = y$,

$$P(X \geq 5|Y) = P\left(\frac{X - Y}{2} \geq \frac{5 - Y}{2} \,\middle|\, Y\right) = 1 - \Phi\left(\frac{5 - Y}{2}\right).$$

Below is the conditional Monte Carlo algorithm.

1. For $i = 1, \ldots, N$:
   A. Generate $U_j \overset{iid.}{\sim}$ Unif(0,1), $j = 1, \ldots, 10$.
   B. Compute $Y_i = \sum_{j=1}^{10} \mathbf{1}\{U_j > 0.5\}$
2. Output:

$$\hat{\theta}_{CMC} = 1 - \frac{1}{N} \sum_{i=1}^{N} \Phi\left(\frac{5 - Y_i}{2}\right),$$

where $\Phi(\cdot)$ is the cdf of a standard normal random variable, and $\hat{\theta}_{CMC}$ is the conditional Monte Carlo estimator.

Since Y is a binomial random variable with parameters $n = 10$ and $p = 0.5$,

$$\begin{aligned}
m_1 = \mathbb{E}\left[\Phi\left(\frac{5 - Y}{2}\right)\right] &= \sum_{y=0}^{10} \Phi\left(\frac{5 - y}{2}\right) P(Y = y) \\
&= \sum_{y=0}^{4} \Phi\left(\frac{5 - y}{2}\right) P(Y = y) + \sum_{y=6}^{10} \Phi\left(\frac{5 - y}{2}\right) P(Y = y) + \Phi(0) P(Y = 5) \\
&= \sum_{y=0}^{4} \Phi\left(\frac{5 - y}{2}\right) P(Y = y) + \sum_{y=0}^{4} \left(1 - \Phi\left(\frac{5 - y}{2}\right)\right) P(Y = y) + \Phi(0) P(Y = 5) \\
&= \sum_{y=0}^{4} P(Y = y) + \frac{1}{2} P(Y = 5) = 1/2. \\
m_2 = \mathbb{E}\left[\Phi^2\left(\frac{5 - Y}{2}\right)\right] &= \sum_{y=0}^{10} \Phi^2\left(\frac{5 - y}{2}\right) P(Y = y) \approx 0.313762.
\end{aligned}$$

Therefore,

$$Var[\hat{\theta}_{CMC}] = \frac{1}{N}(m_2 - m_1^2) \approx \frac{0.06376}{N}.$$

(c) Design a simulation algorithm that uses stratified sampling. Assuming that each stratum k contains np_k samples. Clearly specify the estimator you are using. Calculate the expected variance of this stratified sampling method.

Let $n_k = N \cdot p_k$ and $p_k = P(Y = k)$. Below is the stratified sampling algorithm.

1. For $k = 0, \ldots, 10$:
   - For $i = 1, \ldots, n_k$:
     A. Generate $Z_i^{(k)} \sim N(0, 1)$.
     B. Compute $X_i^{(k)} = k + 2Z_i$.
2. Output:

$$\hat{\theta}_{SS} = \frac{1}{N} \sum_{k=0}^{10} \sum_{i=1}^{n_k} \mathbf{1}\{X_i^{(k)} \geq 5\},$$

   which is the stratified sampling estimator.

By this algorithm,

$$h_k = \mathbb{E}[\mathbf{1}\{X_i^{(k)} \geq 5\}] = P(X \geq 5 | Y = k) = 1 - \Phi\left(\frac{5-k}{2}\right).$$

Therefore,

$$Var[\hat{\theta}_{SS}] = \frac{1}{N} \sum_{k=0}^{10} P(Y = k) \cdot Var[\mathbf{1}\{X_i^{(k)} \geq 5\}] = \frac{1}{N} \sum_{k=0}^{10} p_k \cdot (h_k - h_k^2) = \frac{0.1862}{N}.$$

(d) If we use the optimal number of samples for part (c), will your variance result change?

Let $\sigma_k^2 = Var[\mathbf{1}\{X \geq 5\} | Y = k] = h_k - h_k^2$. We know the optimal number of samples for part (c) is

$$n_k^* = \left(\frac{p_k \sigma_k}{\sum_{j=0}^{10} p_j \sigma_j}\right) N.$$

Below is the optimal stratified sampling algorithm.

1. For $k = 0, \ldots, 10$:
   - For $i = 1, \ldots, n_k^*$:
     A. Generate $Z_i^{(k)} \sim N(0, 1)$.
     B. Compute $X_i^{(k)} = k + 2Z_i$.
2. Output:

$$\hat{\theta}_{SS}^* = \sum_{k=0}^{10} \frac{p_k}{n_k^*} \sum_{i=1}^{n_k^*} \mathbf{1}\{X_i^{(k)} \geq 5\}.$$

The variance of $\hat{\theta}_{SS}^*$ is

$$Var[\hat{\theta}_{SS}^*] = \sum_{k=0}^{10} \frac{p_k^2 \sigma_k^2}{n_k^*} = \frac{\left(\sum_{k=0}^{10} p_k \sigma_k\right)^2}{N} \approx \frac{0.1790}{N},$$

which is smaller than the variance of $\hat{\theta}_{SS}$ in part (c).

(e) Compare the variance reduction results of part (b) and part (c) against part (a) and explain why one is greater than the other.

Recall $\hat{\theta}_{NMC} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}\{X_i \geq 5\}$ and

$$Var[\hat{\theta}_{NMC}] = Var[E[\hat{\theta}_{NMC} \mid Y]] + E[Var[\hat{\theta}_{NMC} \mid Y]].$$

For part (b), we have $Var[\hat{\theta}_{CMC}] = Var[E[\hat{\theta}_{NMC} \mid Y]] \leq Var[\hat{\theta}_{NMC}]$. Similarly, for part (c), $Var[\hat{\theta}_{SS}] = E[Var[\hat{\theta}_{NMC} \mid Y]] \leq Var[\hat{\theta}_{NMC}]$.

Question 3: Antithetic Sampling vs. Control Variate

Suppose we wanted to estimate θ, where

$$\theta = \int_0^1 e^{x^2} dx.$$

(a) Design an antithetic sampling algorithm to estimate θ.

1. For $k = 1, \ldots, n$:
   - Generate $U_k \sim U[0, 1]$;
   - Compute $\theta_{2k-1} = e^{U_k^2}$ and $\theta_{2k} = e^{(1-U_k)^2}$.
2. Output:

$$\hat{\theta} = \frac{1}{2n} \sum_{k=1}^{2n} \theta_k.$$

In [5]:
```
def antithetic_sampling(size:int):
    u = np.random.random(int(size/2))
    theta = np.hstack([np.exp(u**2), np.exp((1-u)**2)])

    return theta.mean()
```

(b) Do 100 simulation runs of 1000 samples to calculate the sample variance of the antithetic sampling estimator.

In [6]:
```
np.random.seed(1)
theta_antithetic = np.array([antithetic_sampling(1000) for i in range(100)])
print(f'Antithetic Sampling Estimator Variance: {theta_antithetic.var(ddof=1)}')
```

```
Antithetic Sampling Estimator Variance: 5.888068389452984e-05
```

(c) Design a simulation algorithm using a control variate to estimate θ.

1. For $k = 1, \ldots, n$:
   - Generate $U_k \sim U[0, 1]$;
   - Compute $X_k = e^{U_k^2}$.
2. Compute

$$c = -\frac{\sum_{k=1}^{n}(X_k - \bar{X})(U_k - \bar{U})}{\sum_{k=1}^{n}(U_k - \bar{U})^2}.$$

3. Output:

$$\hat{\theta} = \frac{1}{n} \sum_{k=1}^{n}(X_k + c(U_k - 1/2)).$$

In [7]:
```
def control_variate(size:int):
    u = np.random.random(size)
    x = np.exp(u**2)
    c = -np.sum((x-x.mean())*(u-u.mean())) / np.sum((u-u.mean())**2)
    theta = x + c*(u-1/2)

    return theta.mean()
```

(d) Use the same seeds, do 100 simulation runs of 1000 samples to calculate the sample variance of the estimator obtained in (c).

In [8]:
```
np.random.seed(1)
theta_control = np.array([control_variate(1000) for i in range(100)])
print(f'Control Variate Estimator Variance: {theta_control.var(ddof=1)}')
```

```
Control Variate Estimator Variance: 2.9887054457305277e-05
```

(e) Which of the two types of variance reduction techniques work better in this example?

The control variate technique works better in this example.