IS 604 – Homework #4

Include the programming code (R, MATLAB, or Python) you used for each problem.

1. In this problem, you will implement and investigate a series of variance reduction procedures for Monte Carlo methods by estimating the expected value of a cost function c(x) which depends on a D-dimensional random variable x.

The cost function you will use is:

$$c(x) = \frac{1}{(2\pi)^{\frac{D}{2}}} \, e^{-\frac{1}{2}x^T x}$$

where

$$x_i \sim U(-5, 5) \quad \text{for} \quad i = 1..D$$

That is, *x* is a *D*-dimensional random variable (i.e., a *D*x1 column vector) and each component of *x* is uniformly distributed between -5 and 5. Note that *c*(*x*) conveniently assumes the form of a standard multivariate normal, and no correlation exists between the components.

Your goal is to estimate E[*c*(*x*)] - the expected value of *c*(*x*) – using Monte Carlo methods and see how it compares to the real value, which you are able to find by hand.
In 1-dimension, the real value of E[*c*(*x*)] can be found as follows:

$$\begin{aligned}
\mathrm{E}[c(x)] &= \int_\Omega c(x)p(x)dx \\
&= \int_{-5}^{5} c(x)p(x)dx \\
&= \int_{-5}^{5} \frac{1}{\sqrt{2\pi}} \, e^{-\frac{1}{2}x^2} \, \frac{1}{5-(-5)} \, dx \\
&= \frac{1}{10}\int_{-5}^{5} \frac{1}{\sqrt{2\pi}} \, e^{-\frac{1}{2}x^2} dx \\
&= \frac{1}{10}\big(\Phi(5) - \Phi(-5)\big) \\
&\approx \frac{1}{10}
\end{aligned}$$

Because both the $c(x)$ and $p(x)$ are separable, you can do a similar analysis in D dimensions to see that:

$$E[c(x)] = \left(\frac{1}{10}\right)^D$$

To simplify this problem, you should code in a way that allows you to easily change D. In addition, you may find it helpful to create a function to evaluate $c(x)$ at N samples in D dimensions (i.e., a DxN matrix)

a) Crude Monte Carlo

For sample sizes of 1000 to 10000 (in increments of 1000), obtain 100 estimates for $E[c(x)]$ when $D = 1$, using crude Monte Carlo sampling.

Calculate the average value of the 100 estimates as well as their standard deviation, and plot them. In your plot, include a line showing the analytical value for $E[c(x)]$.

In addition, use these results to create a table containing the sample size, the average value of the 100 estimates, the standard deviation of the 100 estimates and the coefficient of variation of the 100 estimates (defined as the standard deviation divided by the mean value).

How does the error of your estimate decrease according to the sample size? Repeat this analysis for $D = 2$. Does the error of your estimate increase or decrease as the dimensions increases?

b) Quasi-Random Numbers

We now investigate the variance reduction properties of quasi-random numbers using Sobol numbers.

To see how these Sobol numbers differ from normal random numbers, generate and plot 100 pairs of both. Comment on the differences.

Repeat the analysis of part a) for $D = 1$ and $D = 2$ using Sobol quasi-random numbers. How does the use of Sobol numbers change the average value and standard deviation of your estimates in $D = 1$ and $D = 2$?

c) Antithetic Variates

A simple method that is sometimes used for reducing the variance of the estimate is to induce negative correlation between random draws. When the random variable being generated is uniform, or is a function of a uniform random variable, then one can induce negative correlation between the draws using the following simple procedure:

$$x_1 \sim Uniform(0,1)$$
$$x_2 = 1 - x_1$$

Then the estimate of the function can be formed as:

$$f(x) = \frac{f(x_1) + f(x_2)}{2}$$

Repeat the analysis of part a) for $D = 1$ and $D = 2$ using antithetic variates. Given that antithetic variates require us to generate twice as many random numbers and functional evaluations, be sure to only generate $N/2$ uniform random numbers and then use the procedure defined above to obtain another $N/2$ random numbers. How does the use of antithetic variates change the average value and standard deviation of your estimates? Is this what you expected? Why?

d) Latin Hypercube Sampling

Latin Hypercube Sampling is a variance reduction procedure that is based on the idea of stratified sampling. In this approach, you divide the interval into K equally probable subintervals, and generate a uniform sample from within each subinterval.

Repeat the analysis of part a) for D=1 and D=2 using Latin Hypercube Sampling. How does the use of Latin Hypercube change the average value and standard deviation of your estimates?

e) Importance Sampling

The goal of importance sampling is to estimate E[$c(x)$] by choosing a second density $q(x)$ such that:

$$\begin{aligned}
\mathrm{E}_p[c(x)] &= \int c(x)p(x)dx \\
&= \int c(x)p(x)\frac{q(x)}{q(x)}dx \\
&= \int c(x)q(x)\frac{p(x)}{q(x)}dx \\
&= \int c(x)q(x)w(x)dx \\
&= \mathrm{E}_q[c(x)]
\end{aligned}$$

Here, $\mathrm{E}_p[c(x)]$ and $\mathrm{E}_q[c(x)]$ denotes that the expected value of $c(x)$ is calculated over the probability distribution p and q respectively. In addition, $w(x) = p(x) / q(x)$ is the 'weight' for each value of x that corrects for the bias associated with sampling from the importance sampling density q as opposed to p. As referred to in the literature, one can obtain a minimum variance estimate of the mean using the importance sampling density $q(x) = c(x)\, p(x) / E_p[c(x)]$.

Although the minimum variance importance sampling density is generally not possible to calculate (why?), in our case, it simplifies to $c(x)$ (why?).

In this part we use the minimum variance importance sampling distribution to calculate $E[c(x)]$. To do this, repeat the analysis of part a) for $D = 1$ and $D = 2$ but generate your samples from the minimum variance distribution $q(x)$ instead of a uniform distribution. To obtain the correct answer, be sure to adjust the costs that you obtain for each sample x according to its weight $p(x) / q(x)$.

f) Summary

Plot the estimates of $E[c(x)]$ you obtain in parts a) - e) for $N = 1000$ and $N = 4000$, and $D = 1$ and $D = 2$. Comment on the variance reduction that is possible through each of the methods and how it scales with the sample size and the dimension of the problem.

2. *SCR* Textbook Problems: 6.3, 6.4, 7.1, 7.4