1 Abstract

- We will evaluate the same definite integral
- We use importance sampling to improve its efficiency
- We use inverse transform for exact sampling

2 Problem

Example 1 *Our goal is to compute, using OMC by exact sampling*

$$\alpha = \int_0^1 h(x)dx$$

where

$$h(x) = 100 \cdot I_{(0,1/100]}(x) + 1 \cdot I_{(1/100,1)}(x).$$

Of course, the exact value shall be

$$\alpha = 1.99.$$

Pretended not to know the exact value, we have used OMC with exact sampling of uniform random variable as follows:

Algorithm 1 Integral by MC - Example 1

procedure MCINTEGRAL(N) ▷ N is total number of samples
  $s \leftarrow 0$ ▷ s is the sum of samples
  for $i = 1...N$ **do**
    generate a number Y from $U(0,1)$ ▷ use *numpy.random.uniform*
    $s \leftarrow s + h(Y)$
  return $\frac{s}{N}$ ▷ return the average

Next, we are going to improve the efficiency by using importance sampling. We also extend our skill on exact sampling by using inverse transform.

3 Analysis

3.1 Importance sampling

Recall that, to estimate the above integral α, we use the uniform random variable X, whose density is $p(x) = I_{(0,1)}(x)$, and write

$$\alpha = \mathbb{E}[h(X)|X \sim p] = \int_0^1 h(x)p(x)dx.$$

Naturally, one can sample iid uniform random numbers by computer, denoted by

$$\{\ X_i \sim p : i = 1, 2, \ldots, n\},$$

then taking their average for its approximation of α, i.e.

$$\hat{\alpha}_n = \frac{1}{n}\sum_{i=1}^{n} h(X_i).$$

Example 2 *Compute MSE of $\hat{\alpha}_n$. Verify it with your code.*

Solution. Since it is unbiased, MSE is the same as Variance of $\hat{\alpha}_n$, and it is again equal to $1/n$ of

$$Var[h(X)|X \sim p].$$

Therefore, it is $\frac{97.0299}{n}$. $\square$

IS considers, with a smart choice of a pdf p_1 (to be determined),

$$\alpha = \int_0^1 h(x)\frac{p(x)}{p_1(x)}p_1(x)dx = \mathbb{E}\Big[h(X)\frac{p(X)}{p_1(X)}\Big|X \sim p_1(x)\Big]$$

Since we observe that the interval $(0, 1/100)$ is much more *important* than $(1/100, 1)$, our choice of p_1 is the following:

$$p_1(x) = \frac{1}{C}(2 \cdot I_{(0,1/100]}(x) + 1 \cdot I_{(1/100,1)}(x)),$$

where $C = 101/100$ is the normalizing constant to make p_1 to be a valid pdf.

Algorithm 2 Integral by importance sampling - Example 1

procedure ISINTEGRAL(N) ▷ N is total number of samples
  $s \leftarrow 0$ ▷ s is the sum of samples
  for $i = 1...N$ **do**
    generate a number $Y \sim p_1$ by inverse transform method ▷ ITM to be explained
    $s \leftarrow s + h(Y) \cdot \frac{p(Y)}{p_1(Y)}$
  return $\frac{s}{N}$ ▷ return the average

Example 3 *Prove that MSE of $\hat{\alpha}_n$ is $51.4999/n$. Verify it with your code.*

3.2 Inverse transform method

To implement the IS, we shall generate p_1 samples. But this is not directly available by python. Inverse transform method provides exact sampling as long as the inverse of CDF is explicitly available. Its theoretic basis is given next.

Algorithm 3 ITM sample generation for $X \sim F$ given F^{-1}

procedure ISINTEGRAL(F^{-1}) ▷ F^{-1} is the inverse of CDF
  generate a number Y from $U(0, 1)$ ▷ use *numpy.random.uniform*
  $X = F^{-1}(Y)$
  return X

Proposition 1 *Suppose X has its CDF F and its inverse F^{-1} exists, then $F^{-1}(U) \sim X$, where $U \sim U(0, 1)$.*

PROOF:

$$\mathbb{P}(F^{-1}(U) \leq x) = \mathbb{P}(U \leq F(x)) = F(x).$$

$\square$