

1 Abstract

You will learn

- taking definite integral by ordinary Monte Carlo (OMC)
- exact sampling with python provided random number generators

2 Problem

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

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

where

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

The exact value shall be

$$\alpha = 1.98.$$

3 Analysis

3.1 OMC by exact sampling

To estimate

$$\alpha = \mathbb{E}[X], \quad X \sim p(x)$$

one can use random number generator by computer (if possible)

$$\{iid \ X_i \sim p(x) : i = 1, 2, \dots, n, \}.$$

Then, one can compute the approximation of α by

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

We say $\hat{\alpha}_n$ as OMC by *exact sampling*, since the sample X_i produced by random generator has the same distribution as true distribution X , i.e.

$$X_i \sim X, \quad \forall i.$$

The properties of the OMC by exact sampling are listed below:

- X_1 its self can be treated as an unbiased MC, because

$$\mathbb{E}[X_1] = \alpha.$$

However, MSE is big, ie.

$$MSE(X_1) = Var(X) = \int x^2 p(x)dx.$$

- $\hat{\alpha}_n$ is *consistent almost surely* due to LLN, i.e.

$$\hat{\alpha}_n \rightarrow \alpha, \text{ almost surely as } n \rightarrow \infty.$$

Moreover, $\hat{\alpha}_n$ is unbiased too, and

$$MSE(\hat{\alpha}_n) = Var(\hat{\alpha}_n) = \frac{1}{n} Var(X) \rightarrow 0.$$

3.2 Evaluation of integral

Back to our Example 1, we write

$$\alpha = \mathbb{E}[X] = \mathbb{E}[h(Y)],$$

where $X = h(Y)$ and $Y \sim U(0, 1)$. In other words, although X -sampling is not directly available in python, one can use $U(0, 1)$ random generator (see *numpy.random.uniform*) to produce Y_i , then compute $h(Y_i)$ for the sample X_i .

Algorithm 1 Integral by MC - Example 1

1: procedure MCINTEGRAL(N)	▷ N is total number of samples
2: $s \leftarrow 0$	▷ s is the sum of samples
3: for $i = 1 \dots N$ do	
4: generate two numbers Y from $U(0, 1)$	▷ use <i>numpy.random.uniform</i>
5: $s \leftarrow s + h(Y)$	
6: return $\frac{s}{N}$	▷ return the average
