Assignment 4

1. Pr($X > 2.5$)の近似

私は，プログラミング言語に Python を用いて，Pr($X > 2.5$)の近似を行った．その結果を図 1 に示す．図 1 において，横軸はサンプル数，縦軸は近似した確率をそれぞれ表す．以下に実験方法を概説する．

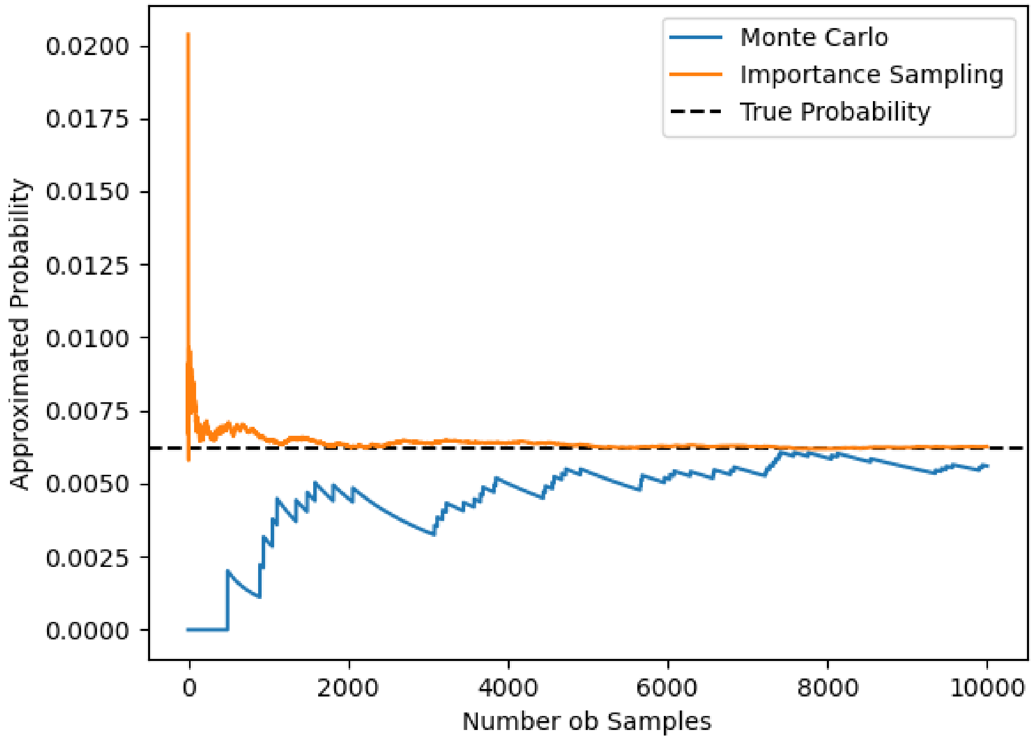

図 1 モンテカルロ・重点サンプリングによる Pr($X > 2.5$)の近似値

A) モンテカルロ法による近似

図 2 のコードによって，モンテカルロ法による近似を行った．6 行目で，numpy を使用して確率分布 X からサンプリングを行っている．8 行目では，サンプル数が 1 ～”*sampling_num*”のときの確率の近似値を計算している．すなわち，サンプル数が n ($n = 1,2, \ldots,$”*sampling_num*”)の場合に，サンプルの値が 2.5 より大きいサンプルの割合を求めている．

```
import numpy as np

np.random.seed(0)

sampling_num = 10000
sample = np.random.randn(sampling_num)

prob = map(lambda x: np.mean(sample[:x] > 2.5), ¥
           range(1, sampling_num + 1))
```

図 2 モンテカルロ近似に使用したソースコード

B) 重点サンプリングによる近似

図 3 のコードによって，モンテカルロ法による近似を行った．6 行目で，numpy を使用して確率分布 $N(3, 1)$からサンプリングを行っている．8・9 行目では，元の確率分布を f, サンプリングを行った確率分布を g として定義している．11 行目では，サンプル数が 1〜”*sampling_num*”のときの確率の近似値を計算している．すなわち，サンプル数が n (n = 1,2, …, ”*sampling_num*”)の場合に，サンプルの値が 2.5 より大きいサンプルの割合を f と g を利用して求めている．

```
1   import numpy as np
2
3   np.random.seed(0)
4
5   sampling_num = 10000
6   imp_sample = np.random.randn(sampling_num) + 3
7
8   f = lambda x: (np.exp(-x**2/2)) / np.sqrt(2*np.pi)
9   g = lambda x: (np.exp(-(x-3)**2/2)) / np.sqrt(2*np.pi)
10
11  imp_prob = map(lambda x: np.mean((imp_sample[:x] > 2.5) ¥
          * (f(imp_sample[:x])/g(imp_sample[:x]))), ¥
          range(1, sampling_num + 1))
```

図 3 重点サンプリングに使用したソースコード

2.考察

確率分布 X から，2.5 以上の値がサンプリングされる確率は非常に低い．ここで，(単純な)モンテカルロ法で正確に近似しようとすると，2.5 以上の値を複数サンプリングしている必要がある．したがって，モンテカルロ法では，少ないサンプル数では正確に近似することができない．言い換えると，サンプリング数が少ない場合，近似が不安定になる．

一方，重点サンプリングでは，2.5 以上の値をサンプリングできる確率を高めている．そのため，少ないサンプル数であっても正確な近似ができる．