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．

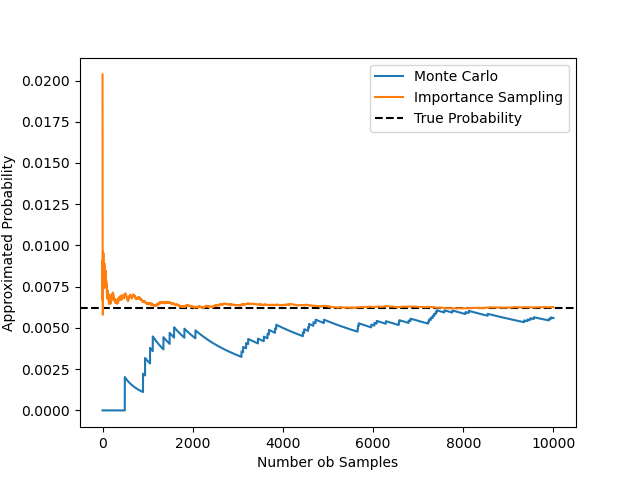


図 1モンテカルロ・重点サンプリングによるPr(*X* > 2.5)の近似値

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, …, ”*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．

図 2 モンテカルロ近似に使用したソースコード

|  |  |
| --- | --- |
| 1 | import numpy as np |
| 2 |  |
| 3 | np.random.seed(0) |
| 4 |  |
| 5 | sampling\_num = 10000 |
| 6 | sample = np.random.randn(sampling\_num) |
| 7 |  |
| 8 | prob = map(lambda x: np.mean(sample[:x] > 2.5), \  range(1, sampling\_num + 1)) |

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．

図 3重点サンプリングに使用したソースコード

|  |  |
| --- | --- |
| 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)) |

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．