

Variações do método de Monte Carlo usando amostra Quasi-Aleatória

Matheus Oliveira da Silva

Maio 2023

1 Introdução

O objetivo deste experimento é verificar quais alterações aparecem ao alterar a geração de amostras aleatórias para quasi-aleatórias. Os métodos de integração que utilizam dessas amostras são os métodos de integração de integração. Sendo esses métodos: o Crude Monte Carlo, o Monte Carlo Hit or Miss, o Importance Sampling e o Control Variates.

A hipótese testada será a de que a geração de amostras quasi-aleatórias melhora a eficácia da integração utilizada pelos métodos anteriormente citados.

Para tal hipótese, os critérios utilizados serão o de velocidade de execução em uma máquina específica, visto que temos por objetivo analisar os efeitos empíricos da alteração de amostras.

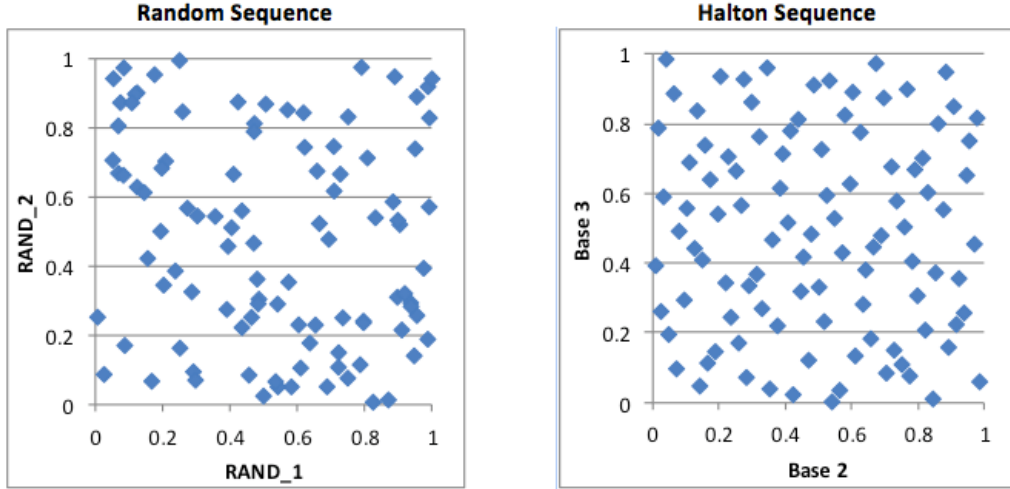
2 Testes

Altere o programa em Python que foi utilizado para implementar os métodos de integração. A priori as amostras aleatórias utilizadas eram geradas pela biblioteca *random*. A posteriori o programa usa a biblioteca *scipy.stats* com o submódulo *qmc* (Quasi-Monte Carlo) e o motor que utiliza o método *Halton*.

A alteração do código foi bem simples e substituiu os números aleatórios gerados pela biblioteca *random* pelos números aleatórios gerados pela biblioteca *scipy*.

Como o cálculo do tamanho da amostra definitivo é feito de acordo com a variância de uma amostra piloto, recalculei o tamanho e a variância dos quatro métodos de integração. Não houve diferença significativa no tamanho

Figura 1:



da amostra e, portanto, para efeito de comparação decidi manter os mesmos valores que os testes anteriores.

2.1 Crude Monte Carlo

Com $n = 10699142$. O método do Crude Monte Carlo é melhor, se considerar o tamanho da amostra necessária, apenas que o Hit or Miss. E, portanto, não é um dos mais eficientes entre os métodos.

Visto essa falta de eficiência, a alteração de biblioteca defasou a geração da amostra, que é uma das maiores, e portanto o tempo de execução do Crude Monte Carlo se tornou muito pior do que antes da alteração da biblioteca.

2.2 Monte Carlo Hit or Miss

Com $n = 86470557$. O método Hit or Miss, sendo o pior na questão tamanho da amostra, demonstrou uma piora extremamente significativa no tempo de execução, por conta do seu extenso número de pontos aleatórios necessários.

2.3 Importance Sampling

Com $n = 948576$. O método Importance Sampling, sendo o segundo método com o menor tamanho de amostra, demorou cerca de 740 segundos, e também não aproximou o suficiente a integração se comparado com os testes anteriores.

2.4 Control Variates

Com $n = 248163$. O método Control Variate é o de menor tamanho de amostra, e, portanto, teve o melhor tempo de execução em relação aos testes usando as amostras quasi-aleatórias.

3 Resultados e discussões

Considerando nossa hipótese inicial, de que a utilização de bibliotecas com geradores quasi-aleatórios aumenta a eficiência dos métodos de integração, a partir dos testes realizados utilizando a implementação, conclui-se que a hipótese é falsa.

Além do tempo de execução ser maior na geração de números quasi-aleatórios, a aproximação apresentou uma maior discrepância do resultado estimado por testes utilizando números aleatórios.

Até mesmo os métodos mais eficientes apresentaram tais problemas, tornando o programa ineficiente e incorreto.

De acordo com a Lei dos Grandes Números [1], a variância e a média amostral tendem à variância e média populacional quando o tamanho da amostra é suficientemente grande, ou tende ao infinito. Partindo dessa lei, podemos observar que para amostras suficientemente grandes, a geração de números aleatórios e quasi-aleatórios se aproxima, não revelando grandes diferenças entre si a não ser o seu método de implementação computacional.

Portanto, como o método de geração quasi-aleatório exige maior desempenho computacional do que a geração aleatória, conclui-se que para testes e simulações com o tamanho da amostra suficientemente grandes, é dispensável o uso de geradores quasi-aleatórios, como nessa simulação.

Referências

- [1] Sheldon Ross. *A First Course in Probability*. Pearson, Boston, MA, 10th edition, 2019.