

EP4 - MCMC

George Othon - NUSP 103xxxxx
Felipe Zaffalon - NUSP 103xxxxx

Maio de 2020

1 Introdução

Este relatório tem como objetivo calcular uma aproximação para a seguinte integral:

$$\int_0^{+\infty} h(x)g(x)dx$$

Onde $h(x)$ é função indicadora no intervalo $[1,2]$. Com $h(x) = I(1 \leq x \leq 2)$. E $g(x) \propto \text{gamma}(C,x) \cdot |\cos(Rx)|$, ou seja, $g(x) = k \cdot \text{gamma}(C,x) \cdot |\cos(Rx)|$, com $k \in \mathbb{R}$. Com os parâmetros $C = 1.39xxxxxx$ e $R = 1.103xxxxx$.

A aproximação será feita através de dois algoritmos, o de Metropolis e o de Baker, em que os dois utilizam a cadeia de Markov, e a única diferença é o parâmetro de aceitação.

2 Resultados

Para gerar uma aproximação, criamos duas funções no python, uma que utiliza o algoritmo de Metropolis e outra por Baker. A condição de parada foi quando o erro atingisse 1% ou menos, a partir de um número mínimo de passos.

Como resultado obtivemos que:

	Resultado Estimado	Erro Estimado	Nº de Iterações
Metropolis	0.126756	0.01	11392
Baker	0.125939	0.01	20367

Figure 1: Resultados

Para chegar neste erro estimado utilizamos a seguinte métrica:

$$erro = \frac{\sigma}{\sqrt{N}}$$

Em que σ é o desvio padrão da cadeia de Markov, e N o tamanho da cadeia.

Para calcular a função g não foi necessário saber o k , já que, tanto na aceitação α de Metropolis quanto na de Baker calculamos uma razão que cancela o multiplicador k .

Para o algoritmo de Metropolis:

$$\begin{aligned} \alpha &= \frac{g(x_n)}{g(x_{n-1})} \\ &= \frac{k \cdot \text{gamma}(C, x_n) \cdot |\cos(Rx_n)|}{k \cdot \text{gamma}(C, x_{n-1}) \cdot |\cos(Rx_{n-1})|} \\ &= \frac{\text{gamma}(C, x_n) \cdot |\cos(Rx_n)|}{\text{gamma}(C, x_{n-1}) \cdot |\cos(Rx_{n-1})|} \end{aligned}$$

E para o algoritmo de Baker, temos:

$$\begin{aligned} \alpha &= \frac{g(x_n)}{g(x_{n-1}) + g(x_n)} \\ &= \frac{k \cdot \text{gamma}(C, x_n) \cdot |\cos(Rx_n)|}{k \cdot \text{gamma}(C, x_{n-1}) \cdot |\cos(Rx_{n-1})| + k \cdot \text{gamma}(C, x_n) \cdot |\cos(Rx_n)|} \\ &= \frac{\text{gamma}(C, x_n) \cdot |\cos(Rx_n)|}{\text{gamma}(C, x_{n-1}) \cdot |\cos(Rx_{n-1})| + \text{gamma}(C, x_n) \cdot |\cos(Rx_n)|} \end{aligned}$$

Logo, não é necessário saber o k para calcular essa integral pelo método MCMC.

Após diversos testes percebemos que com o aumento do desvio padrão, a quantidade de passos até atingir o erro esperado era maior, e quando o desvio padrão era muito próximo de zero não tínhamos uma boa aproximação.

Para não permitir valores de x negativos, nós colocamos ele em módulo. Isso é possível pois, por estarmos usando uma curva normal centrada no zero para gerar o número aleatório, temos que a probabilidade de um valor ser negativo é a mesma de seu oposto. Logo, não estamos alterando a aleatoriedade do gerador.

Para encontrar o número adequado de passos para obter erro menor ou igual a 1%, a partir de 2000 iterações passamos a calcular o erro e verificar se atendia ao requisito, e enquanto não atendesse nós incrementamos o número de passos.