

O algoritmo de Passeio Aleatório (PA) é um método de Monte Carlo baseado nas Cadeias de Markov (MCCM) da classe de algoritmos de *Metropolis-Hastings*, onde o próximo valor na cadeia $X_t = X_{t-1} + \varepsilon_t$, sendo ε_t uma perturbação aleatória com distribuição G simétrica ($g(x) = g(-x)$), sendo g a função de distribuição de probabilidade (f.d.p) de G). Exemplos destas são a distribuição uniforme e a distribuição normal, levando a que $X_{t+1} \sim \mathcal{U}(X_t - \lambda, X_t + \lambda)$ ou $X_{t+1} \sim \mathcal{N}(X_t, \sigma^2)$.¹

O algoritmo de *Metropolis-Hastings* consiste no seguinte:

1. Escolher um x_0 aleatório do domínio da função de d.p. objetivo $f(x) \propto P(x)$, e escolher uma distribuição candidata G (com f.d.p. $g(x_t | x_{t-1})$) de mesmo domínio.

O PA usará uma distribuição simétrica como candidata, de forma a garantir que $g(x_{t-1}|x_t^*) = g(x_t^*|x_{t-1})$. As propriedades da distribuição (como σ na distribuição normal) terão de ser escolhidas também, influenciando a cadeia.

2. Gerar um $x_t^* \sim G$.
3. Gerar u a partir de $\mathcal{U}(0, 1)$ (Monte Carlo).

4. Calcular $\alpha = \frac{f(x_t^*)}{f(x_{t-1})} \frac{g(x_{t-1}|x_t^*)}{g(x_t^*|x_{t-1})}$.

Como a distribuição candidata é simétrica, $\frac{g(x_{t-1}|x_t^*)}{g(x_t^*|x_{t-1})} = 1 \Rightarrow \alpha = \frac{f(x_t^*)}{f(x_{t-1})}$.

5. Comparar u com α

Se $\alpha > u$ então x_t^* é aceite como x_t , e o algoritmo repete a partir do ponto 2 com o novo $t \leftarrow t + 1$ (ou acaba dependendo da quantidade da amostra que pretendemos)

Se não, então x_t^* é rejeitado e o algoritmo volta para o ponto 2 sem alterar t

Como isto é uma cadeia de Markov, para uma amostra suficientemente grande, há uma convergência para a distribuição $f(x)$. Como isto é um PA, e a distribuição candidata é simétrica, G não terá influência na condição de aceitação, mas sim na velocidade de convergência e na autocorrelação dos pontos. A calibração das propriedades de G são importantes para chegar à aproximação de f , e a Figura 1 presente no **Problem Set 2** reflete isso.

A Figura 1 apresenta evolução de várias cadeias geradas com o PA com uma distribuição Qui-Quadrado χ_2^2 como distribuição objetivo, usando uma distribuição normal $\mathcal{N}(0, \sigma^2)$ como candidata, sendo a principal diferença entre as cadeias o σ escolhido.

Na cadeia *rw.l*, $\sigma = 0.05$, fazendo com que gere somente valores consecutivos altos, em que todos são aceites.

No gráfico 2, à semelhança do 3, vemos que a função converge para o domínio da Qui-Quadrado, aceitando quase na sua totalidade os pontos gerados. Contudo, para o gráfico 2 a convergência é lenta e no gráfico 3 é bastante mais elevada, devido à variância ser maior (diferença dos passos seguintes ser maior). No gráfico 4, em contradição com o gráfico 1, a convergência é efetuada muito rapidamente, contribuindo para que o algoritmo não aceite pontos gerados e fique preso no mesmo sítio na execução das iterações (ter retas horizontais nas iterações vendo que permaneceu no último estado).

Talvez o mais ideal seria optar pelo gráfico 3 ou por um valor de desvio padrão entre os dois últimos gráficos, de forma a melhorar a eficácia da convergência, mas não rejeitando os pontos tal como acontece no gráfico 4, para um desvio padrão de 16.

¹Robert, C. P., & Casella, G. (2010). *Introducing Monte Carlo Methods with R*. Em *Springer eBooks*. <https://doi.org/10.1007/978-1-4419-1576-4>