Estimando o valor de uma integral a partir de seu gerador

Laboratório de Computação e Simulação

(MAP2212)

Alunos:	**Lucka de Godoy Gianvechio e Milton Leal Neto**
Curso:	Bacharelado de Matemática Aplicada Computacional
NUSP:	11352442 e 8973974
Professor:	Julio Stern

São Paulo, 14 de junho de 2021

Conteúdo

1 Apresentação **1**

2 Plotando a função f **2**

3 Estratégia de resolução **3**

4 Definindo n **4**

5 Estrutura do programa **6**

6 Conclusão **7**

7 Referências **8**

1 Apresentação

Este relatório apresenta uma solução para o quarto Exercício Programa (EP) proposto pelo professor Julio Stern no âmbito da disciplina de Laboratório de Computação e Simulação (MAP 2212) do Bacharelado de Matemática Aplicada Computacional (BMAC) do Instituto de Matemática e Estatística (IME) da Universidade de São Paulo (USP).

A título de registro, nos foi solicitado que obtivéssmos uma função $U(v)$ que pudesse estimar, com erro $< 0.05\%$, a função verdade

$$W(v) = \int_{T(v)} f(\theta|x,y)d\theta$$

que representa a massa de probabilidade a posteriori no domínio $T(v)$, ou seja, a massa de probabilidade correspondente da função $f(\theta|x,y)$ que não ultrapassa um determinado nível v.

A função f, que tem distribuição de probabilidades Dirichlet, dada por

$$f(\theta|x,y) = \frac{1}{B(x+y)} \prod_{i=1}^{m} \theta_i^{x_i+y_i-1}$$

representa o modelo estatístico m-dimensional Multinomial e recebe como parâmetros um vetor de observações x, um vetor de informações a priori y e um vetor de probabilidades θ, sendo que $x, y \in N^m, \theta \in \Theta = S_m = \{\theta \in R_m^+ | \theta' 1 = 1\}$ e B representa a distribuição $Beta$. Aqui trabalhamos com $m = 3$.

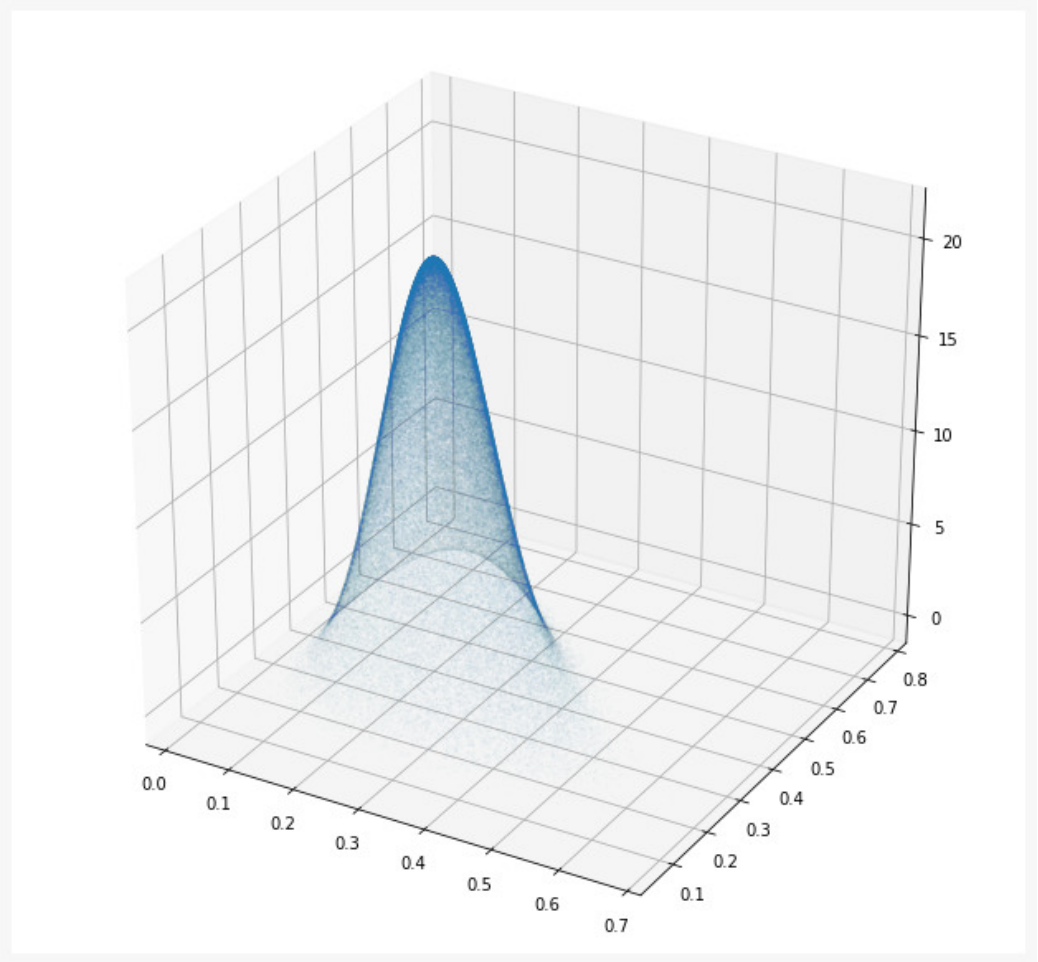
Além disso, vale ressaltar que a região $T(v)$ foi definida como

$$T(v) = \{\theta \in \Theta | f(\theta|x,y) \leq v.$$

2 Plotando a função f

Inicialmente, para fins de entendimento geral, plotamos a função f para um dado conjunto de vetores de entrada, sendo $x = [2, 4, 4]$ e $y = [4, 5, 6]$. A intenção era obter um pouco de intuição sobre a forma da função que iríamos trabalhar.

Abaixo, vemos o plot da f, que mostra no 'plano do chão' os valores assumidos por θ_1 e θ_2 e os valores da imagem da f no eixo vertical. Aqui consideramos $\theta_3 = 1 - (\theta_1 + \theta_2)$



3 Estratégia de resolução

Para estimarmos a $W(v)$, nos foi solicitado que definíssemos k pontos de corte tal que $0 = v_0 < v_1 < v_2 < \cdots < v_k = sup f(\theta)$. Em seguida, a partir de pontos gerados de uma distribuição Dirichlet, nos foi pedido para usar a fração de pontos simulados, θ_t, dentro de cada bin, definido por $v_{j-1} < f(\theta_t) < v_j$, como uma aproximação para $W(v_j) - W(v_{j-1})$. A partir disso, a proposta era ajustar dinamicamente as bordas de cada bin de forma a obter pesos aproximadamente iguais, tal que $W(v_j) - W(v_{j-1}) \approx \frac{1}{k}$.

Ainda que esta estratégia seja eficaz e interessante, principalmente, para problemas que envolvem vetores cujos tamanhos sejam realmente grandes, optamos por adotar um caminho diferente, que julgamos ser mais fácil de implementar e entender.

Nossa estratégia foi gerar pontos aleatórios a partir de uma Dirichlet, avaliar a função f nestes pontos, considerando obviamente os vetores de entrada inseridos pelo usuário, e simplesmente ordenar os valores da imagem de f de modo a obter uma maneira fácil de identificar quantos pontos estavam abaixo de um determinado nível v desejado.

Assim, dado um valor v de corte inserido pelo usuário, simplesmente contamos o número de valores da imagem de f abaixo deste referencial e o dividimos pelo total de pontos gerados para obter uma aproximação da função $W(v)$.

Dessa forma, podemos interpretar que existem $k = n$ intervalos que contêm um único $f(\theta)$ cada, de modo que a aproximação para cada intervalo seja $1/k$.

De posse deste método, passamos a buscar uma maneira de definir o número n de pontos que precisariam ser gerados da Dirichlet de modo a obtermos o nível de acurácia estipulado no enunciado do EP. Vejamos a seguir como obtivemos tal quantidade.

4 Definindo n

Levando em consideração o que foi dito na seção anterior e pensando que existam n intervalos ou *bins*, decidimos utilizar o Teorema Central do Limite para estimar o valor ótimo de n que nos levasse à acurácia desejada, considerando o erro absoluto.

Para tanto, utilizamos a fórmula

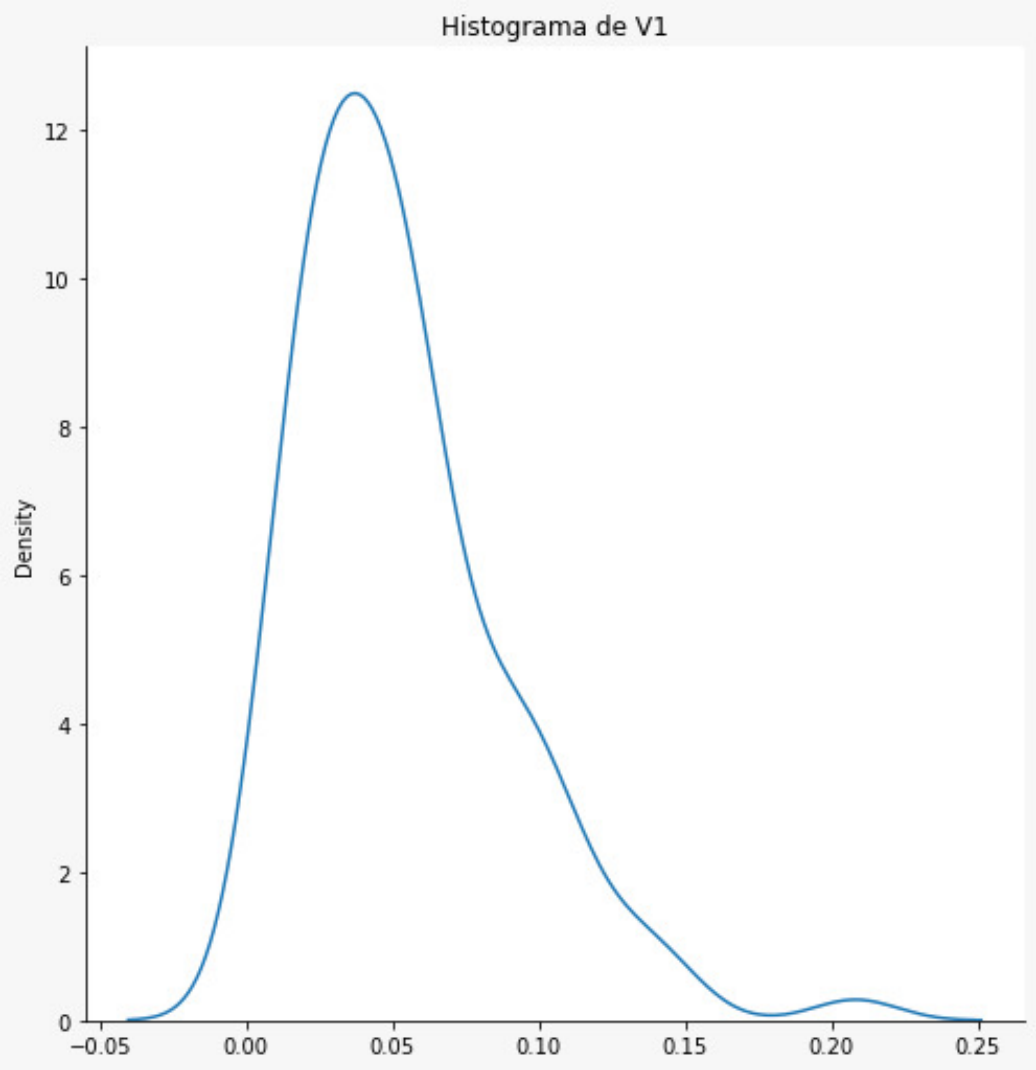
$$n_{final} = \frac{\Phi^{-1}(1 - \frac{\delta}{2})^2 \cdot \hat{\sigma}_2}{\epsilon^2},$$

na qual $\Phi^{-1}(1-\frac{\delta}{2})$ corresponde ao percentil de uma distribuição $Normal(0, 1)$, $\hat{\sigma}^2$ corresponde à variância da amostra piloto e ϵ corresponde ao erro máximo suportado, no caso 0.0005. Neste trabalho, decidimos considerar $\delta = 95\%$.

Para estimarmos a variância, criamos 100 experimentos aleatórios nos quais geramos 1000 pontos aleatórios de uma Dirichlet, avaliamos a f nestes pontos, ordenamos os resultados e consideramos a média entre os dois menores valores obtidos em cada simulação como uma estimativa para a posição daquele que poderia ser considerado o v_1, ou seja, o primeiro nível de corte a ser feito na função.

Dessa forma, obtivemos 100 valores de v_1 e calculamos a variância destes valores e a utilizamos como variânia amostral do nosso experimento. Com essa estratégia, a depender dos vetores de entrada, o valor final de n ficou em torno de 20.000 pontos.

Na próxima página, vemos um histograma dos 100 valores de v_1.


Histograma de V1
Density
12
10
8
6
4
2
0
−0.05
0.00
0.05
0.10
0.15
0.20
0.25

5 Estrutura do programa

O programa escrito em *Python* está estruturado em quatro funções:

1) calcula_n_final():

Realiza o experimento para estimar a variância amostral e calcula o n_{final} a ser utilizado no programa.

2) calcula_f():

Computa a constante de normalização, a função f e ordena os resultados obtidos.

3) estima_W():

Verifica quantos pontos existem abaixo do nível de corte desejado e retorna a proporção de pontos em relação ao total de pontos gerados como resultado da função $U(v)$, que estima a $W(v)$.

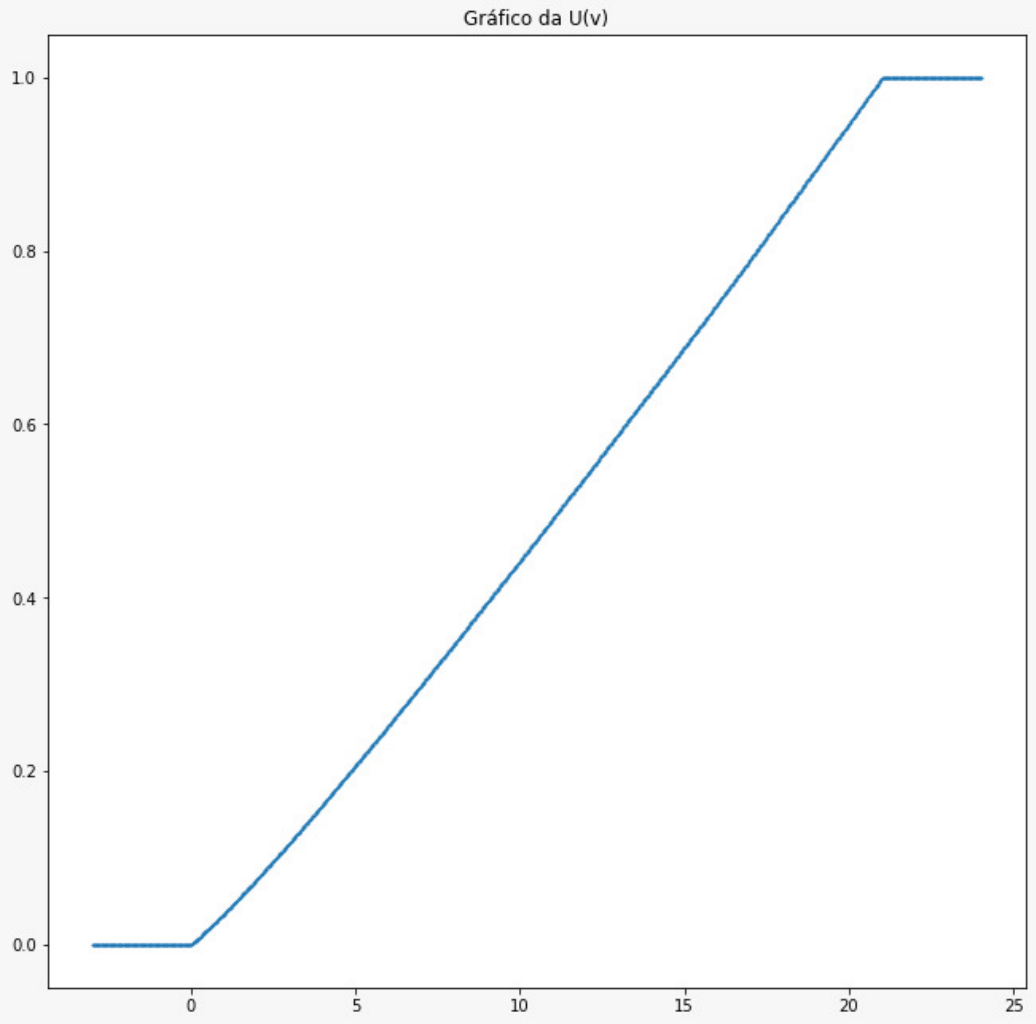
4) main():

Chamada principal do programa. Inclui as linhas de código que interagem com o usuário e que imprimem os resultados na tela.

6 Conclusão

Podemos concluir que o método proposto neste trabalho, que utiliza a própria integral para estimar o valor dela mesma, é eficiente e oferece uma boa alternativa aos métodos utilizados nos EPs anteriores para estimarmos o valor de uma integral.

Abaixo, vemos um gráfico que mostra o comportamento da função $U(v)$ para um determinado par de vetores de entrada x, y. Como era esperado, a função $U(v)$ atua, intuitivamente, como uma espécie de função de distribuição acumulada da função original f.



7 Referências

[1] S.Kaplan, C.Lin, (1987). An Improved Condensation Procedure in Discrete Probability Distribution Calculations. Risk Analysis, 7, 15-19.

[2] C.A.B.Pereira, J.M.Stern, (1999). Evidence and Credibility: Full Bayesian Significance Test for Precise Hypotheses. Entropy Journal, 1, 69-80.