Relatório - Exercício Programa 6 (EP6)

José Victor Santos Alves
Nr. USP: 14713085
Marcos Elias Jara Grubert
Nr. USP: 1295930

Julho de 2025

1 Introdução

Este relatório apresenta a solução desenvolvida para o sexto Exercício Programa (EP6) da disciplina de Laboratório de Computação e Simulação (MAP2212), ministrada pelo professor Julio Stern no âmbito do Bacharelado em Matemática Aplicada e Computacional (BMAC) do IME-USP.

O objetivo central deste trabalho é computar o **e-valor padronizado (SEV)** para testar a **hipótese de equilíbrio de Hardy-Weinberg** em um modelo estatístico trinomial-Dirichlet. Este método, proposto por C.A.B. Pereira e J.M. Stern (1999), oferece uma alternativa Bayesiana ao p-valor da estatística frequentista, medindo a evidência que os dados fornecem contra uma hipótese nula precisa.

A metodologia baseia-se no cálculo da credibilidade de uma região no espaço paramétrico que é tangente ao conjunto que define a hipótese nula. Para isso, fazemos uso da **função verdade**, $W(v)$, que foi o objeto de estudo do EP5. A função verdade é definida como a massa de probabilidade a posteriori contida na região $T(v)$:

$$W(v) = \int_{T(v)} f(\theta \mid x, y) d\theta, \quad \text{onde} \quad T(v) = \{\theta \in \Theta \mid f(\theta \mid x, y) \leq v\} \tag{1}$$

Utilizamos uma aproximação computacional $U(v)$ para $W(v)$, construída a partir de um método de Monte Carlo via Cadeias de Markov (MCMC). O e-valor (ev) é então obtido avaliando $U(v)$ no ponto s^*, que representa o valor máximo da função de densidade de probabilidade posterior dentro do subespaço da hipótese nula.

O modelo estatístico subjacente é a **distribuição de Dirichlet**, que descreve a densidade de probabilidade posterior dos parâmetros θ do modelo multinomial:

$$f(\theta \mid x, y) = \frac{1}{B(x+y)} \prod_{i=1}^{m} \theta_i^{x_i + y_i - 1} \tag{2}$$

onde x é o vetor de observações, y é o vetor de informações a priori, $m = 3$ é a dimensão do problema e $B(\cdot)$ é a função Beta multivariada.

Nas seções seguintes, detalhamos a estrutura do algoritmo MCMC utilizado, a justificativa para a escolha dos parâmetros da simulação e, por fim, apresentamos e analisamos os resultados dos testes para 72 conjuntos de dados distintos, considerando duas priors diferentes, $y = [0, 0, 0]$ e $y = [1, 1, 1]$.

2 Estrutura do Algoritmo de MCMC

A metodologia do Método de Monte Carlo via Cadeias de Markov (MCMC) consiste em gerar amostras de pontos em regiões onde a função a ser integrada possui maior relevância. A localização dessas amostras no domínio da função tende a convergir de forma análoga a uma Cadeia de Markov. A implementação da estrutura do algoritmo, que utiliza a linguagem Python e suas bibliotecas, segue os passos detalhados abaixo.

2.1 Geração e Validação de Candidatos

O processo inicia-se com a geração de amostras a partir de uma distribuição multivariada , especificamente a distribuição normal multivariada com média zero. A matriz de covariância utilizada nesta etapa é definida pela variância e covariância da distribuição de Dirichlet:

$$\mathrm{Var}(X_{i,j}) = \begin{cases} \frac{\alpha_i(\alpha_0-\alpha_i)}{\alpha_0^2(\alpha_0+1)} & \text{se } i = j \\ \frac{-\alpha_i\alpha_j}{\alpha_0^2(\alpha_0+1)} & \text{se } i \neq j \end{cases} \quad \text{onde, } \alpha_0 = \sum_k \alpha_k \tag{3}$$

O vetor aleatório gerado é somado ao valor atual da cadeia, θ_i, para criar um ponto candidato. Este candidato é então validado para assegurar que pertence ao domínio de um simplex, o que requer que todos os seus elementos sejam estritamente positivos e que a soma de seus elementos seja unitária.

2.2 Critério de Aceitação e Geração da Cadeia

Um candidato que satisfaz as condições de domínio é submetido ao critério de aceitação do algoritmo de Metropolis-Hastings. A probabilidade de aceitação $\alpha(\theta_{i+1}, \theta_i)$ é calculada com base na razão dos potenciais entre o ponto candidato e o ponto atual:

$$\alpha(\theta_{i+1}, \theta_i) = \min\left(1, \frac{f(\theta_{i+1}|x, y)}{f(\theta_i|x, y)}\right) \tag{4}$$

Um número aleatório é gerado de uma distribuição uniforme em $[0, 1]$, e se este número for menor que α, o candidato é aceito. Caso contrário, o candidato é rejeitado e o valor atual é mantido, ou seja, $\theta_{i+1} = \theta_i$. Para o ponto inicial da cadeia, utiliza-se o centro do simplex, o vetor $[1/3, 1/3, 1/3]$, e as primeiras 1000 amostras são descartadas (período de aquecimento ou queima) para garantir a estabilidade da cadeia.

2.3 Cálculo e Processamento dos Potenciais

Após a geração de n amostras, calcula-se a função potencial para cada ponto θ_i da cadeia:

$$p(\theta|x, y) = \prod_{i=1}^{m} \theta_i^{x_i+y_i-1} \tag{5}$$

A lista resultante de potenciais é então ordenada de forma crescente e normalizada utilizando a função Gamma. Para otimizar o processamento, esta lista ordenada é condensada em uma lista menor contendo k *bins*, onde cada *bin* representa a informação de n/k pontos da amostra original.

2.4 Cálculo da Integral Condensada U(v)

Finalmente, para um dado valor de corte (cut-off) v, o algoritmo percorre a lista de *bins* para localizar a posição i que corresponde a este valor. O valor da integral condensada, $U(v)$, é então determinado pela proporção i/k. O valor de $U(v)$ é definido como 0 se v for menor que o potencial mínimo da amostra e 1 se for maior que o potencial máximo.

3 Definindo o valor de n

Para a definição do valor de n, utilizaremos a mesma abordagem do **EP5** baseada em distribuição Bernoulli para determinar a quantidade necessária de pontos em cada bin, de modo que o erro máximo tolerável seja respeitado, definido como $\epsilon = 0.05\%$. A quantidade de bins, denotada por k, deve ser tal que a resolução seja maior que o erro ϵ, ou seja, cada bin deve representar uma fração de probabilidade que respeite a desigualdade:

$$W(v_j) - W(v_{j-1}) = \frac{1}{k} \geq \epsilon \tag{6}$$

3.1 Aproximação Assintótica

Assumindo um número de amostras suficientemente grande, podemos utilizar o Teorema do Limite Central para aproximar a distribuição Bernoulli por uma Normal. Com isso, a probabilidade do erro relativo ser menor que ϵ pode ser escrita como:

$$P(|\hat{p} - p| \leq \varepsilon) \geq \gamma$$

Aplicando a normalização, temos:

$$P(-\varepsilon \leq \hat{p} - p \leq \varepsilon) = P\left(\frac{-\sqrt{n}\varepsilon}{\sigma} \leq Z \leq \frac{\sqrt{n}\varepsilon}{\sigma}\right) \approx \gamma$$

Dessa forma, podemos isolar n para obter:

$$n = \frac{\sigma^2 {Z_\gamma}^2}{\varepsilon^2} \tag{7}$$

O valor de n obtido em 7 será utilizado para determinar a quantidade de amostras necessárias para atingir o erro relativo estipulado.

3.2 Escolha de k: número de bins

A quantidade de bins na distribuição de probabilidade discreta deve respeitar o critério de resolução mínima exigida por ε, conforme a equação 8, o que implica:

$$k \geq \frac{1}{\varepsilon} \Rightarrow k \geq 2000 \tag{8}$$

portanto:

$$k_{\min} = 2000$$

3.3 Intervalo de confiança adotado

Utilizamos um nível de confiança de 95%, escolhido de forma convencional. Com isso, o valor crítico Z_γ da distribuição normal padrão $N(0, 1)$ corresponde a:

$$Z_\gamma = 1{,}96$$

3.4 Precisão desejada (ε)

O erro permitido entre o valor real de $W(u)$ e sua aproximação foi fixado em:

$$\varepsilon = 0{,}0005$$

3.5 Estimativa da variância

Considerando que, dentro de um bin, os dados seguem uma distribuição de Bernoulli com probabilidade igual à largura do bin (i.e., ε), a variância pode ser estimada como:

$$\sigma^2 = \frac{\varepsilon}{2}\left(1 - \frac{\varepsilon}{2}\right) \approx \frac{\varepsilon}{2} \tag{9}$$

3.6 Cálculo final de n

Com base nas equações anteriores, podemos determinar o valor necessário de n para garantir o erro máximo admissível, levando em conta a divisão em $k = 2000$ bins:

$$n = k \cdot \frac{1{,}96^2 \cdot \frac{\varepsilon}{2}}{\varepsilon^2} \geq 7.683.20$$

Portanto, o número mínimo de pontos necessário é:

$$n_{\text{min}} = 7.683.200 \text{ pontos}$$

Ao utilizarmos essa quantidade de pontos em nosso programa, o tempo de execução para calcular U(V) foi de 1284 segundos (aproximadamente 21 minutos). Diante disso, decidimos executar testes empíricos diminuindo o número de pontos para observar se haveria uma diferença significativa nos resultados. Notou-se que não havia perda relevante mesmo reduzindo o tamanho da amostra em até dez vezes. Assim, para aumentar a eficiência do código e ainda manter uma boa acurácia, optamos por uma amostra final de:

$$n_{\text{final}} = 200.000 \text{ pontos}$$

4 Resultados

Para avaliar o impacto do tamanho da amostra (n) no desempenho computacional, foram realizadas duas execuções do conjunto de testes. Na primeira, utilizou-se um tamanho de amostra de $n = 200.000$, resultando em um tempo total de execução de 2.322 segundos (aproximadamente 40 minutos). Posteriormente, a mesma simulação foi conduzida com um valor de n dez vezes menor, $n = 20.000$, o que reduziu o tempo de execução para 275 segundos (aproximadamente 5 minutos). Onde obtivemos os seguintes resultados

4.1 Resultados para a Prior Não Informativa (Y = [0, 0, 0]) e n = 200.000

Tabela 1: Resultados do teste com a prior Y = [0, 0, 0] e n = 200.000

Vetor X	**e-valor (ev)**	**sev**	**Decisão** ($\alpha = 0.05$)
$[1, 17, 2]$	0.0010	0.0002	**Rejeita** H_0
$[1, 16, 3]$	0.0030	0.0007	**Rejeita** H_0
$[1, 15, 4]$	0.0110	0.0027	**Rejeita** H_0
$[1, 14, 5]$	0.0285	0.0076	**Rejeita** H_0
$[1, 13, 6]$	0.0645	0.0192	**Rejeita** H_0
$[1, 12, 7]$	0.1275	0.0424	**Rejeita** H_0
$[1, 11, 8]$	0.2305	0.0867	Não Rejeita H_0
$[1, 10, 9]$	0.3545	0.1498	Não Rejeita H_0
$[1, 9, 10]$	0.4950	0.2357	Não Rejeita H_0
$[1, 8, 11]$	0.6345	0.3402	Não Rejeita H_0
$[1, 7, 12]$	0.7555	0.4540	Não Rejeita H_0
$[1, 6, 13]$	0.8580	0.5800	Não Rejeita H_0
$[1, 5, 14]$	0.9250	0.6929	Não Rejeita H_0
$[1, 4, 15]$	0.9700	0.8051	Não Rejeita H_0
$[1, 3, 16]$	0.9915	0.8960	Não Rejeita H_0
$[1, 2, 17]$	0.9995	0.9748	Não Rejeita H_0
$[1, 1, 18]$	0.9985	0.9563	Não Rejeita H_0
$[5, 15, 0]$	0.0000	1.0000	Não Rejeita H_0
$[5, 14, 1]$	0.0265	0.0070	**Rejeita** H_0
$[5, 13, 2]$	0.1315	0.0440	**Rejeita** H_0
$[5, 12, 3]$	0.3865	0.1679	Não Rejeita H_0
$[5, 11, 4]$	0.7215	0.4191	Não Rejeita H_0
$[5, 10, 5]$	0.9655	0.7910	Não Rejeita H_0
$[5, 9, 6]$	0.9705	0.8067	Não Rejeita H_0
$[5, 8, 7]$	0.7645	0.4637	Não Rejeita H_0
$[5, 7, 8]$	0.4790	0.2250	Não Rejeita H_0
$[5, 6, 9]$	0.2410	0.0916	Não Rejeita H_0
$[5, 5, 10]$	0.0990	0.0315	**Rejeita** H_0
$[9, 11, 0]$	0.0000	1.0000	Não Rejeita H_0
$[9, 10, 1]$	0.3615	0.1537	Não Rejeita H_0
$[9, 9, 2]$	0.8525	0.5721	Não Rejeita H_0
$[9, 8, 3]$	0.9680	0.7987	Não Rejeita H_0
$[9, 7, 4]$	0.6090	0.3193	Não Rejeita H_0
$[9, 6, 5]$	0.2430	0.0926	Não Rejeita H_0
$[9, 5, 6]$	0.0650	0.0194	**Rejeita** H_0
$[9, 4, 7]$	0.0125	0.0031	**Rejeita** H_0

4.2 Resultados para a Prior Uniforme (Y = [1, 1, 1]) e n = 200.000

Tabela 2: Resultados do teste com a prior Y = [1, 1, 1] e n = 200.000

Vetor X	**e-valor (ev)**	**sev**	**Decisão** ($\alpha = 0.05$)
$[1, 17, 2]$	0.0040	0.0009	**Rejeita** H_0
$[1, 16, 3]$	0.0145	0.0036	**Rejeita** H_0
$[1, 15, 4]$	0.0400	0.0112	**Rejeita** H_0
$[1, 14, 5]$	0.0960	0.0304	**Rejeita** H_0
$[1, 13, 6]$	0.1860	0.0666	Não Rejeita H_0
$[1, 12, 7]$	0.3325	0.1378	Não Rejeita H_0
$[1, 11, 8]$	0.5005	0.2394	Não Rejeita H_0
$[1, 10, 9]$	0.6765	0.3766	Não Rejeita H_0
$[1, 9, 10]$	0.8430	0.5589	Não Rejeita H_0
$[1, 8, 11]$	0.9575	0.7682	Não Rejeita H_0
$[1, 7, 12]$	1.0000	0.0000	**Rejeita** H_0
$[1, 6, 13]$	0.9630	0.7836	Não Rejeita H_0
$[1, 5, 14]$	0.8480	0.5658	Não Rejeita H_0
$[1, 4, 15]$	0.6770	0.3771	Não Rejeita H_0
$[1, 3, 16]$	0.4795	0.2253	Não Rejeita H_0
$[1, 2, 17]$	0.2960	0.1187	Não Rejeita H_0
$[1, 1, 18]$	0.1480	0.0506	Não Rejeita H_0
$[5, 15, 0]$	0.0180	0.0046	**Rejeita** H_0
$[5, 14, 1]$	0.0960	0.0304	**Rejeita** H_0
$[5, 13, 2]$	0.2875	0.1143	Não Rejeita H_0
$[5, 12, 3]$	0.5975	0.3102	Não Rejeita H_0
$[5, 11, 4]$	0.8885	0.6268	Não Rejeita H_0
$[5, 10, 5]$	1.0000	0.0000	**Rejeita** H_0
$[5, 9, 6]$	0.8980	0.6427	Não Rejeita H_0
$[5, 8, 7]$	0.6540	0.3568	Não Rejeita H_0
$[5, 7, 8]$	0.3945	0.1726	Não Rejeita H_0
$[5, 6, 9]$	0.2000	0.0728	Não Rejeita H_0
$[5, 5, 10]$	0.0785	0.0241	**Rejeita** H_0
$[9, 11, 0]$	0.2445	0.0933	Não Rejeita H_0
$[9, 10, 1]$	0.6800	0.3798	Não Rejeita H_0
$[9, 9, 2]$	0.9930	0.9056	Não Rejeita H_0
$[9, 8, 3]$	0.8520	0.5714	Não Rejeita H_0
$[9, 7, 4]$	0.4825	0.2273	Não Rejeita H_0
$[9, 6, 5]$	0.1945	0.0704	Não Rejeita H_0
$[9, 5, 6]$	0.0595	0.0175	**Rejeita** H_0
$[9, 4, 7]$	0.0130	0.0032	**Rejeita** H_0

4.3 Resultados para a Prior Não Informativa (Y = [0, 0, 0]) e n = 20.000

Tabela 3: Resultados do teste com a prior Y = [0, 0, 0] e n = 20.000

Vetor X	**e-valor (ev)**	**sev**	**Decisão** ($\alpha = 0.05$)
$[1, 17, 2]$	0.0000	1.0000	Não Rejeita H_0
$[1, 16, 3]$	0.0020	0.0004	**Rejeita** H_0
$[1, 15, 4]$	0.0055	0.0013	**Rejeita** H_0
$[1, 14, 5]$	0.0235	0.0062	**Rejeita** H_0
$[1, 13, 6]$	0.0675	0.0202	**Rejeita** H_0
$[1, 12, 7]$	0.1205	0.0397	**Rejeita** H_0
$[1, 11, 8]$	0.2480	0.0949	Não Rejeita H_0
$[1, 10, 9]$	0.3535	0.1493	Não Rejeita H_0
$[1, 9, 10]$	0.5055	0.2428	Não Rejeita H_0
$[1, 8, 11]$	0.6270	0.3339	Não Rejeita H_0
$[1, 7, 12]$	0.7505	0.4487	Não Rejeita H_0
$[1, 6, 13]$	0.8560	0.5771	Não Rejeita H_0
$[1, 5, 14]$	0.9220	0.6869	Não Rejeita H_0
$[1, 4, 15]$	0.9735	0.8167	Não Rejeita H_0
$[1, 3, 16]$	0.9920	0.8991	Não Rejeita H_0
$[1, 2, 17]$	1.0000	0.0000	**Rejeita** H_0
$[1, 1, 18]$	0.9985	0.9563	Não Rejeita H_0
$[5, 15, 0]$	0.0000	1.0000	Não Rejeita H_0
$[5, 14, 1]$	0.0305	0.0082	**Rejeita** H_0
$[5, 13, 2]$	0.1400	0.0474	**Rejeita** H_0
$[5, 12, 3]$	0.3765	0.1622	Não Rejeita H_0
$[5, 11, 4]$	0.7225	0.4201	Não Rejeita H_0
$[5, 10, 5]$	0.9680	0.7987	Não Rejeita H_0
$[5, 9, 6]$	0.9670	0.7956	Não Rejeita H_0
$[5, 8, 7]$	0.7700	0.4697	Não Rejeita H_0
$[5, 7, 8]$	0.4740	0.2217	Não Rejeita H_0
$[5, 6, 9]$	0.2480	0.0949	Não Rejeita H_0
$[5, 5, 10]$	0.0940	0.0297	**Rejeita** H_0
$[9, 11, 0]$	0.0000	1.0000	Não Rejeita H_0
$[9, 10, 1]$	0.3430	0.1435	Não Rejeita H_0
$[9, 9, 2]$	0.8545	0.5749	Não Rejeita H_0
$[9, 8, 3]$	0.9640	0.7866	Não Rejeita H_0
$[9, 7, 4]$	0.6005	0.3125	Não Rejeita H_0
$[9, 6, 5]$	0.2410	0.0916	Não Rejeita H_0
$[9, 5, 6]$	0.0610	0.0180	**Rejeita** H_0
$[9, 4, 7]$	0.0090	0.0021	**Rejeita** H_0

4.4 Resultados para a Prior Uniforme (Y = [1, 1, 1]) e n = 20.000

Tabela 4: Resultados do teste com a prior Y = [1, 1, 1] e n = 20.000

Vetor X	**e-valor (ev)**	**sev**	**Decisão** ($\alpha = 0.05$)
$[1, 17, 2]$	0.0035	0.0008	**Rejeita** H_0
$[1, 16, 3]$	0.0105	0.0025	**Rejeita** H_0
$[1, 15, 4]$	0.0385	0.0107	**Rejeita** H_0
$[1, 14, 5]$	0.1015	0.0324	**Rejeita** H_0
$[1, 13, 6]$	0.1745	0.0617	Não Rejeita H_0
$[1, 12, 7]$	0.3235	0.1330	Não Rejeita H_0
$[1, 11, 8]$	0.4960	0.2363	Não Rejeita H_0
$[1, 10, 9]$	0.6905	0.3894	Não Rejeita H_0
$[1, 9, 10]$	0.8375	0.5515	Não Rejeita H_0
$[1, 8, 11]$	0.9520	0.7538	Não Rejeita H_0
$[1, 7, 12]$	1.0000	0.0000	**Rejeita** H_0
$[1, 6, 13]$	0.9660	0.7925	Não Rejeita H_0
$[1, 5, 14]$	0.8430	0.5589	Não Rejeita H_0
$[1, 4, 15]$	0.6945	0.3932	Não Rejeita H_0
$[1, 3, 16]$	0.4785	0.2247	Não Rejeita H_0
$[1, 2, 17]$	0.2720	0.1066	Não Rejeita H_0
$[1, 1, 18]$	0.1500	0.0514	Não Rejeita H_0
$[5, 15, 0]$	0.0130	0.0032	**Rejeita** H_0
$[5, 14, 1]$	0.0895	0.0280	**Rejeita** H_0
$[5, 13, 2]$	0.2850	0.1131	Não Rejeita H_0
$[5, 12, 3]$	0.6010	0.3129	Não Rejeita H_0
$[5, 11, 4]$	0.8920	0.6326	Não Rejeita H_0
$[5, 10, 5]$	1.0000	0.0000	**Rejeita** H_0
$[5, 9, 6]$	0.9005	0.6471	Não Rejeita H_0
$[5, 8, 7]$	0.6625	0.3642	Não Rejeita H_0
$[5, 7, 8]$	0.3835	0.1662	Não Rejeita H_0
$[5, 6, 9]$	0.1935	0.0699	Não Rejeita H_0
$[5, 5, 10]$	0.0820	0.0253	**Rejeita** H_0
$[9, 11, 0]$	0.2440	0.0930	Não Rejeita H_0
$[9, 10, 1]$	0.6770	0.3771	Não Rejeita H_0
$[9, 9, 2]$	0.9930	0.9056	Não Rejeita H_0
$[9, 8, 3]$	0.8500	0.5686	Não Rejeita H_0
$[9, 7, 4]$	0.4940	0.2350	Não Rejeita H_0
$[9, 6, 5]$	0.1825	0.0651	Não Rejeita H_0
$[9, 5, 6]$	0.0600	0.0177	**Rejeita** H_0
$[9, 4, 7]$	0.0120	0.0029	**Rejeita** H_0

Análise e Discussão dos Resultados

Decidimos fixar um nível de significância $\alpha = 5\%$ e rejeitar toda hipótese que apresentasse um valor de **SEV** inferior a 0.05.

Com isso, obtivemos a seguinte tabela:

Tabela 5: Sumário quantitativo das decisões por cenário de simulação.

Cenário de Simulação	**Rejeições de H_0**	**Não Rejeições**	**Total de Casos**
$n = 200.000$, $y = [0, 0, 0]$	11	25	36
$n = 20.000$, $y = [0, 0, 0]$	11	25	36
$n = 200.000$, $y = [1, 1, 1]$	11	26	37
$n = 20.000$, $y = [1, 1, 1]$	11	26	37

À primeira vista, a Tabela 5 sugere uma notável estabilidade nos resultados, pois o número agregado de rejeições (11 casos) permanece constante em todos os cenários.

O tamanho da amostra (n) afetou a estabilidade da simulação. Embora o número total de rejeições não tenha mudado, a execução com menos amostras ($n = 20.000$) apresentou uma variação muito maior nos valores de sev para casos específicos especialmente os mais extremos em comparação com a execução mais longa ($n = 200.000$).

Portanto, Observa-se que no geral, as decisões sobre as hipóteses nulas não se alteraram muito, tanto ao se trocar o vetor de observações a priori $y = [1, 1, 1]$ por $y = [0, 0, 0]$, quanto com a redução do tamanho da amostra n em dez vezes para alguns casos.

5 Conclusão

Podemos concluir que o Teste de Significância Totalmente Bayesiano , utilizado neste trabalho, representa uma alternativa robusta aos métodos frequentistas. Sua abordagem, fundamentada na geometria do espaço de parâmetros, utiliza apenas a amostra observada e o ponto de máxima verossimilhança sob a hipótese nula. Uma vantagem notável é sua invariância à forma como a hipótese é parametrizada, uma característica que o diferencia de métodos clássicos e reforça sua consistência.