

MAE5911/IME: Fundamentos de Estatística e Machine Learning. Prof.: Alexandre Galvão Patriota

Questão 01: Defina formalmente o valor-p (**aproximado**) para testar uma hipótese geral e discuta os problemas de interpretação usuais ao utilizar a sua versão condicional à hipótese nula. Apresente um exemplo numérico e simulações de Monte Carlo, usando um modelo de regressão Poisson, para ilustrar a sua aplicação.

Questão 02: Apresente um Teorema da Aproximação Universal que generalize o Teorema de Funahashi, discuta as diferenças entre os resultados por meio de exemplos de redes neurais.

Questão 03: Proponha uma função de estimação robusta para estimar os parâmetros de uma regressão Binomial, $Y|X = x \sim \text{Bin}(m, \mu_\theta(x))$, em que $\mu_\theta(\cdot)$ é uma função com imagem em $[0, 1]$ e θ é o vetor de parâmetros. Apresente um exemplo numérico e uma simulação de Monte Carlo, como feito em sala, para ilustrar os resultados. Perturbe a distribuição dos dados e mostre que a sua proposta é de fato robusta contra essa perturbação.

Resposta para o Questão 01

O conceito do p-valor, tal como proposto originalmente por Fisher (1925), é uma medida **condicional sob H_0** , definida por

$$\text{valor-p}(H_0, z_n) = \sup_{\theta \in \Theta_0} P_\theta^{(n)}(T_{H_0}(Z_n) \geq T_{H_0}(z_n)),$$

isto é, a probabilidade de se observar uma estatística tão ou mais extrema quanto a obtida, assumindo H_0 verdadeira. O valor-p foi concebido por Fisher como um *instrumento de contraste* entre o dado e hipótese, mas não como uma medida de verdade. Seu objetivo era expressar o grau de incompatibilidade empírica entre o fenômeno observado e as consequências lógicas de H_0 .

Posteriormente, a escola Neyman–Pearson reinterpreta esse conceito dentro de uma estrutura de decisão. Nessa transição, o valor-p perdeu seu caráter exploratório e passou a ser tratado como um critério binário, externo e supostamente não condicionado a H_0 , capaz de expressar a decisão: “rejeita” ou “não rejeita” H_0 . Esta leitura gerou um equívoco lógico: o valor-p, que é definido dentro do universo em que H_0 é verdadeira, passou a ser interpretado como se pudesse mensurar a probabilidade de H_0 ser verdadeira.

O **valor-p assintótico**, que é o valor usualmente utilizado, na ausência da distribuição exata de T_{H_0} , é definido por:

$$p^{(a)}(H_0, z_n) = \sup_{\theta \in \Theta_0} P_\theta(W_\theta \geq T_{H_0}(z_n)),$$

em que W_θ representa a distribuição-limite de T_{H_0} quando $n \rightarrow \infty$.

Confirmando essa natureza epistêmica, o valor-p assintótico nem sempre converge para o valor-p exato, isto porque, estando definido condicionalmente a H_0 , pode não ser sensível a falhas estruturais do modelo, como a ausência de independência entre as observações, a presença de hipóteses condicionais aninhadas, ou a violação das condições de regularidade que garantem a validade assintótica — como homocedasticidade.

Por isso, o valor-p deve ser entendido apenas como originalmente proposto — uma **medida de discrepância entre o real e o teórico** — que opera em um único sentido: como evidência contra a hipótese nula, mas nunca para confirmá-la.

Em termos das σ -álgebras envolvidas, a construção inferencial pode ser representada como uma sequência de aplicações mensuráveis, onde cada espaço de probabilidade possui a sua própria σ -álgebra, refletindo níveis distintos de abstração:

$$(\Omega_n, \mathcal{F}_n, \mathbb{P}_n) \xrightarrow{Z_n} (\mathcal{Z}_n, \mathcal{B}_{Z_n}, \mathbb{P}_{Z_n}) \xrightarrow{T} (\mathcal{T}_n, \mathcal{B}_{T_n}, \mathbb{P}_{T_n}).$$

- \mathcal{F}_n — descreve o universo empírico de incertezas, isto é, os eventos do mundo observável;

- \mathcal{B}_{Z_n} — corresponde ao espaço amostral modelado, onde as variáveis aleatórias Z_n são definidas sob \mathbb{P}_{Z_n} ;
- \mathcal{B}_{T_n} — é a σ -álgebra induzida pela estatística T , onde vivem as distribuições teóricas dos testes e, em particular, o valor-p.

O valor-p pertence ao universo lógico em que a hipótese nula H_0 é assumida como verdadeira. Ele não pertence à σ -álgebra empírica \mathcal{F}_n , e portanto não pode ser interpretado como uma probabilidade sobre o mundo real ou sobre a veracidade de H_0 . Tratá-lo dessa forma constitui um erro de violação da σ -álgebra — um deslize epistemológico em que se tenta extrair informação empírica de uma construção que em sua definição é epistêmica e condicional.

Implementações

O exemplo numérico apresentado na Listing 1 tem o objetivo de ilustrar o comportamento do valor- p sob diferentes hipóteses nulas, mantendo fixos os dados observados. O experimento baseia-se em um modelo de regressão de Poisson definido por

$$Y_i \sim \text{Poisson}(\lambda_i), \quad \text{com} \quad \log(\lambda_i) = \theta_0 + \theta x_i,$$

onde $\theta_0 = 1$ e o valor verdadeiro do parâmetro é $\theta = 0,20$. Uma única amostra de tamanho $n = 100$ foi gerada segundo esse modelo, representando o conjunto de dados “observados” em um mundo empírico fixo.

A partir dessa amostra, foram testadas três hipóteses nulas distintas sobre o mesmo conjunto de dados:

$$H_0^{(A)} = 0,20, \quad H_0^{(B)} = 0,28, \quad H_0^{(C)} = 0,60.$$

Cada hipótese foi avaliada por meio da estatística de Wald,

$$Z = \frac{\hat{\theta} - \theta_{H_0}}{\text{se}(\hat{\theta})},$$

e o valor- p correspondente foi calculado de forma:

$$p = P(Z \leq z_{\text{obs}}),$$

correspondendo à hipótese alternativa $H_1 : \theta < \theta_{H_0}$.

Os resultados obtidos encontram-se na Tabela 1. Observa-se que, quando a hipótese nula coincide com o valor real do parâmetro (θ), o valor- p é elevado e a hipótese não é rejeitada. À medida que o valor hipotético se afasta de θ , o valor- p decresce rapidamente, levando à rejeição de H_0 nos casos de maior discrepância ($H_0^{(C)}$). Essa simulação evidencia o papel do valor- p como medida de compatibilidade entre modelo e dado: ele quantifica o grau de discrepância entre a hipótese estatística e a estrutura empírica observada.

Tabela 1: Resultados do cálculo do valor- p para diferentes hipóteses nulas sob o modelo de regressão de Poisson.

Cenário	Hipótese nula	Valor- p	Decisão	$\hat{\theta}$	θ
Cenário A ($H_0^{(A)}$ exato)	$\theta_{H_0} = 0,20$	0,547473	Não rejeita H_0	0,211726	0,20
Cenário B ($H_0^{(B)}$ com pequeno desvio)	$\theta_{H_0} = 0,28$	0,243689	Não rejeita H_0	0,211726	0,20
Cenário C ($H_0^{(C)}$ com grande desvio)	$\theta_{H_0} = 0,60$	0,000039	Rejeita H_0	0,211726	0,20

O experimento de Monte Carlo apresentado na Listing 2 tem como objetivo avaliar, de forma empírica, o comportamento assintótico do valor- p quando múltiplas amostras são geradas a partir de um mesmo modelo de regressão de Poisson. Considera-se o mesmo modelo

$$Y_i \sim \text{Poisson}(\lambda_i), \quad \text{com} \quad \log(\lambda_i) = \theta_0 + \theta x_i,$$

onde o intercepto $\theta_0 = 1$ e o valor verdadeiro do parâmetro é $\theta = 0,20$. Em cada replicação, é gerada uma amostra de tamanho $n = 100$ com covariáveis $x_i \sim \text{Unif}(0, 2)$ e respostas Y_i conforme o modelo acima. O processo é repetido $M = 10,000$ vezes, mantendo θ fixo.

Para cada amostra, ajusta-se um modelo de regressão de Poisson e testam-se três hipóteses nulas distintas sobre o mesmo conjunto de dados:

$$H_0^{(A)} = 0,20, \quad H_0^{(B)} = 0,28, \quad H_0^{(C)} = 0,60.$$

Cada hipótese é avaliada pela estatística de Wald,

$$Z = \frac{\hat{\theta} - \theta_{H_0}}{\text{se}(\hat{\theta})},$$

e o valor-*p* correspondente é calculado da seguinte forma, considerando o teste contra a alternativa $H_1 : \theta < \theta_{H_0}$:

$$p = P(Z \leq z_{\text{obs}}).$$

A partir das M replicações, calcula-se a proporção de rejeição de cada hipótese, a média e a mediana dos valores-*p*, bem como a média dos estimadores $\hat{\theta}$. Os resultados são apresentados na Tabela 2.

Observa-se que, quando $H_0^{(A)}$ coincide com o valor verdadeiro do parâmetro, a proporção de rejeição aproxima-se do nível $\alpha = 0,05$, e os valores-*p* seguem aproximadamente uma distribuição uniforme. À medida que as hipóteses nulas se deslocam para valores superiores ao verdadeiro — como em $H_0^{(B)}$ e $H_0^{(C)}$ — os valores presentes na porção unicaudal à esquerda se tornam mais expressivos, e os valores-*p* tornam-se progressivamente menores. Nos casos de grande discrepância ($H_0^{(C)}$), a rejeição de H_0 é praticamente certa, refletindo a coerência do teste de Wald com o comportamento teórico esperado sob o modelo de Poisson.

Tabela 2: Resultados da simulação de Monte Carlo para o teste de Wald em regressão de Poisson. Cada linha apresenta os resultados obtidos para uma hipótese nula distinta, com $M = 10,000$ replicações e $n = 100$.

Cenário	H0	% de rejeição	Média(<i>p</i>)	Mediana(<i>p</i>)	<i>p</i> _{0,05}	<i>p</i> _{0,95}	$\bar{\theta} / \theta$
Cenário A	$H_0^{(A)} = 0,20$	0,053700	0,494656	0,488848	0,045770	0,950321	0,1983 / 0,2
Cenário B	$H_0^{(B)} = 0,28$	0,204100	0,282815	0,204148	0,006543	0,804670	0,1983 / 0,2
Cenário C	$H_0^{(C)} = 0,60$	0,989600	0,002594	0,000030	0,000000	0,010802	0,1983 / 0,2

```

1 # -----
2 # Exemplo Numerico - Regressao de Poisson
3 #
4 set.seed(111)
5
6 n      = 100
7 beta0  = 1.0
8 theta  = 0.20    # valor verdadeiro
9 alpha   = 0.05
10
11 # Covariável e única amostra observada
12 x      = runif(n, 0, 2)
13 lambda = exp(beta0 + theta * x)
14 y      = rpois(n, lambda)
15
16 # Ajuste único do GLM Poisson
17 modelo = glm(y ~ x, family = poisson)
18
19 # Função: testa H0
20 testa_glm_theta = function(mod, h0, alpha = 0.05) {
21   theta_hat = coef(mod)[2]
22   se_hat   = sqrt(vcov(mod)[2, 2])
23   Z        = (theta_hat - h0) / se_hat
24
25   # valor-p unicaudal (contra H1: theta < h0)
26   p_valor  = pnorm(Z)
27
28   decisao  = ifelse(p_valor < alpha, "Rejeita H0", "Não rejeita H0")
29 }
```

```

30 data.frame(
31   hipotese_nula = paste0("H0: theta = ", format(h0, nsmall = 2)),
32   valor_p      = p_valor,
33   decisao       = decisao,
34   theta_hat    = theta_hat,
35   theta         = theta
36 )
37 }
38
39 # Hipóteses nulas a testar no mesmo conjunto
40 H0_A = 0.20  # H0 exato
41 H0_B = 0.28  # H0 com pouco desvio
42 H0_C = 0.60  # H0 com muito desvio
43
44 resA = testa_glm_theta(modelo, H0_A, alpha)
45 resB = testa_glm_theta(modelo, H0_B, alpha)
46 resC = testa_glm_theta(modelo, H0_C, alpha)
47
48 # Consolida os resultados
49 resumo_glm = rbind(
50   "Cenário A (H0 exato)"      = resA,
51   "Cenário B (H0 pouco desvio)" = resB,
52   "Cenário C (H0 muito desvio)" = resC
53 )
54
55 # Formatação
56 options(scipen = 999)
57
58 resumo_fmt      = resumo_glm
59 resumo_fmt$valor_p  = format(round(resumo_fmt$valor_p, 6), nsmall = 6)
60 resumo_fmt$theta_hat = format(round(resumo_fmt$theta_hat, 6), nsmall = 6)
61 resumo_fmt$theta    = format(round(resumo_fmt$theta, 6), nsmall = 6)
62
63 # Tabela final
64 noquote(resumo_fmt)

```

Listing 1: Exemplo numérico — Regressão de Poisson ($n = 100, \theta = 0,20$).

```

1 # -----
2 # Simulacao de Monte Carlo - Regressao de Poisson
3 # -----
4 set.seed(111)
5
6 ## Parâmetros do experimento
7 n      = 200      # tamanho da amostra
8 beta0  = 1.0      # intercepto fixo
9 theta  = 0.20     # (verdadeiro) = coef. de x, usado para gerar os dados
10 alpha  = 0.05    # nível
11 M      = 10000   # repetições de Monte Carlo
12
13 ## hipóteses nulas a testar (todas no mesmo conjunto de dados por replicação)
14 H0_A = 0.20    # H0 exato
15 H0_B = 0.28    # H0 com pouco desvio
16 H0_C = 0.60    # H0 com grande desvio
17 H0s  = c(H0_A, H0_B, H0_C)
18
19 ## Design: x é gerado 1 vez e mantido fixo
20 x = runif(n, 0, 2)
21
22 ## Função: calcula p-valor unicaudal (esquerda) e decisão para b0
23 p_unicaudal_wald = function(mod, b0, alpha) {
24   theta_hat = coef(mod)[2]
25   se_hat   = sqrt(vcov(mod)[2, 2])
26   Z        = (theta_hat - b0) / se_hat
27   # unicaudal à esquerda: H1: theta < b0
28   p_valor  = pnorm(Z)
29   decisao  = ifelse(p_valor < alpha, "Rejeita H0", "Não rejeita H0")
30   list(p = p_valor, decisao = decisao, theta_hat = theta_hat)
31 }

```

```

32
33 ## Uma replicação: gera y ~ Poisson(exp(beta0 + theta * x)), ajusta GLM
34 ## e testa H0_A, H0_B, H0_C na mesma amostra
35 uma_replicacao = function() {
36   lambda = exp(beta0 + theta * x)
37   y      = rpois(n, lambda)
38   mod    = glm(y ~ x, family = poisson)
39   lapply(H0s, function(b0) p_unicaudal_wald(mod, b0, alpha))
40 }
41
42 ## Roda M replicações e consolida resultados por cenário
43 ## estruturas de acumulação
44 p_list_A = numeric(M); dec_A = character(M); thA = numeric(M)
45 p_list_B = numeric(M); dec_B = character(M); thB = numeric(M)
46 p_list_C = numeric(M); dec_C = character(M); thC = numeric(M)
47
48 for (m in 1:M) {
49   out = uma_replicacao()
50   # A
51   p_list_A[m] = out[[1]]$p
52   dec_A[m]    = out[[1]]$decisao
53   thA[m]      = out[[1]]$theta_hat
54   # B
55   p_list_B[m] = out[[2]]$p
56   dec_B[m]    = out[[2]]$decisao
57   thB[m]      = out[[2]]$theta_hat
58   # C
59   p_list_C[m] = out[[3]]$p
60   dec_C[m]    = out[[3]]$decisao
61   thC[m]      = out[[3]]$theta_hat
62 }
63
64 ## Sumários: proporção de rejeição (potência/erro tipo I),
65 ## estatísticas de p e média de theta_hat
66 resumo_mc = data.frame(
67   hipotese_nula = c(
68     paste0("H0_A: theta = ", format(H0_A, nsmall = 2)),
69     paste0("H0_B: theta = ", format(H0_B, nsmall = 2)),
70     paste0("H0_C: theta = ", format(H0_C, nsmall = 2))
71   ),
72   prop_rejeicao = c(
73     mean(dec_A == "Rejeita H0"),
74     mean(dec_B == "Rejeita H0"),
75     mean(dec_C == "Rejeita H0")
76   ),
77   media_p = c(mean(p_list_A), mean(p_list_B), mean(p_list_C)),
78   mediana_p = c(median(p_list_A), median(p_list_B), median(p_list_C)),
79   q05_p = c(quantile(p_list_A, 0.05), quantile(p_list_B, 0.05), quantile(p_list_C, 0.05)),
80   q95_p = c(quantile(p_list_A, 0.95), quantile(p_list_B, 0.95), quantile(p_list_C, 0.95)),
81   theta_hat_medio = c(mean(thA), mean(thB), mean(thC)),
82   theta_real = theta
83 )
84
85 # formatação
86 options(scipen = 999)
87 resumo_mc$prop_rejeicao = format(round(resumo_mc$prop_rejeicao, 6), nsmall = 6)
88 resumo_mc$media_p = format(round(resumo_mc$media_p, 6), nsmall = 6)
89 resumo_mc$mediana_p = format(round(resumo_mc$mediana_p, 6), nsmall = 6)
90 resumo_mc$q05_p = format(round(resumo_mc$q05_p, 6), nsmall = 6)
91 resumo_mc$q95_p = format(round(resumo_mc$q95_p, 6), nsmall = 6)
92 resumo_mc$theta_hat_medio = format(round(resumo_mc$theta_hat_medio, 6), nsmall = 6)
93 resumo_mc$theta_real = format(round(resumo_mc$theta_real, 6), nsmall = 6)
94
95 # Tabela final
96 rownames(resumo_mc) = c(
97   "Cenário A (H0_A exato)",
98   "Cenário B (H0_B pouco desvio)",
99   "Cenário C (H0_C muito desvio)"

```

```

100 )
101
102 noquote(resumo_mc)

```

Listing 2: Simulação de Monte Carlo do teste de Wald sob regressão de Poisson ($M = 10,000$, $n = 100$, $\theta = 0,20$).

Resposta para o Questão 02

[Leshno–Lin–Pinkus–Schocken, 1993] Seja $K \subset \mathbb{R}^n$ compacto e $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ localmente limitada. As combinações finitas

$$x \mapsto \sum_{j=1}^N a_j \sigma(w_j^\top x + b_j)$$

são densas em $C(K)$ se, e somente se, σ não é um polinômio. Consequentemente, para qualquer $f \in C(K)$ e $\varepsilon > 0$, existem N , pesos e vieses tais que

$$\sup_{x \in K} \left\| f(x) - \sum_{j=1}^N a_j \sigma(w_j^\top x + b_j) \right\| < \varepsilon.$$

O resultado é demonstrado para funções escalares, mas vale consequentemente para saídas vetoriais, isto é, para redes neurais com vários outputs, aproximando cada componente de $f : K \rightarrow \mathbb{R}^m$.

Este resultado generaliza os de Funahashi (1989) e Cybenko (1989), estendendo ao teorema da universalidade funções de ativação não sigmoidais, como a ReLU. A prova mostra que a propriedade essencial não depende da forma específica da ativação, mas apenas da condição fundamental de que ela não seja um polinômio.

Resposta para o Questão 03

Foi construído um modelo de regressão binomial da seguinte forma:

$$Y_i | X_i \sim \text{Bin}(m, \mu_\theta(X_i)), \quad \text{com} \quad \mu_\theta(x) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x)}},$$

onde $\theta = (\theta_0, \theta_1)$ é o vetor de parâmetros a ser estimado. Os parâmetros verdadeiros foram fixados em $\theta_0 = -0.5$ e $\theta_1 = 1.2$, com $n = 200$ observações e $m = 10$ tentativas por ponto.

As variáveis independentes foram geradas por uma distribuição uniforme $X_i \sim U(-2, 2)$ e as variáveis observadas Y_i foram geradas a partir do modelo binomial acima. Para introduzir contaminação, selecionaram-se 10% das observações com $x_i < 0.5$ e aplicou-se a seguinte modificação adversária:

$$x_i \leftarrow x_i + 3, \quad y_i \leftarrow \begin{cases} 0, & \text{com probabilidade 0.5,} \\ m, & \text{com probabilidade 0.5.} \end{cases}$$

Gerando *outliers* com desvio em relação ao eixo x , de resposta, e *outliers* com desvio em relação ao eixo y , de alavancagem.

Função de Estimação Robusta

O estimador de máxima verossimilhança (MLE) é obtido pela minimização do negativo da log-verossimilhança binomial, cuja parametrização em termos do logito conduz ao modelo de regressão logística. Nesse caso, os parâmetros $\theta = (\beta_0, \beta_1)$ são estimados como

$$\hat{\theta}_{\text{MLE}} = \arg \min_{\theta} \left\{ - \sum_{i=1}^n [y_i \log \mu_\theta(x_i) + (m - y_i) \log(1 - \mu_\theta(x_i))] \right\},$$

em que $\mu_\theta(x_i) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_i)}}$ representa a probabilidade de sucesso segundo o modelo logístico.

Para tornar a estimação robusta, definiu-se uma **função de estimação ponderada** baseada no resíduo $r_i = y_i - m\mu_\theta(x_i)$:

$$U_i(\theta) = \begin{pmatrix} w_i(y_i - m\mu_\theta(x_i)) \\ w_i x_i(y_i - m\mu_\theta(x_i)) \end{pmatrix}, \quad w_i = \exp(-(|r_i| - c)_+), \quad c = p \cdot \text{IQR}(r),$$

em que $(a)_+ = \max(0, a)$ indica a parte positiva do argumento, isto é, $(|r_i| - c)_+ = 0$ quando $|r_i| \leq c$ e $(|r_i| - c)_+ = |r_i| - c$ quando $|r_i| > c$. Dessa forma, o argumento da exponencial em $w_i = \exp(-(|r_i| - c)_+)$ atua como uma penalização progressiva aplicada apenas aos resíduos que excedem o limite $c = p \cdot \text{IQR}(r)$, onde $p = 1.5$ controla o ponto de corte do intervalo interquartil. Assim, observações com $|r_i| \leq c$ mantêm peso unitário ($w_i = 1$), enquanto aquelas com $|r_i| > c$ têm seu peso reduzido de forma exponencial, atenuando a influência dos outliers no processo de estimação.

O estimador robusto $\hat{\theta}_R$ é então obtido pela minimização do quadrado da soma das funções de estimação:

$$\hat{\theta}_R = \arg \min_{\theta} \left\| \sum_{i=1}^n U_i(\theta) \right\|^2.$$

Simulação e Resultados

A implementação da regressão binomial robusta é apresentada no *Listing 3* e define uma função de estimação ponderada baseada nos resíduos $r_i = y_i - m\mu_\theta(x_i)$, com pesos adaptativos $w_i = \exp(-(|r_i| - c)_+)$, $c = p \cdot \text{IQR}(r)$, em que o ponto de corte c é proporcional à dispersão interquartil dos resíduos, conforme já apresentada.

A Figura 1 ilustra o comportamento do ajuste em um único experimento, mostrando as curvas estimadas pelo MLE e pelo método robusto em comparação com a curva verdadeira dos dados. Os pontos destacados em vermelho correspondem às observações atenuadas pela ponderação robusta.

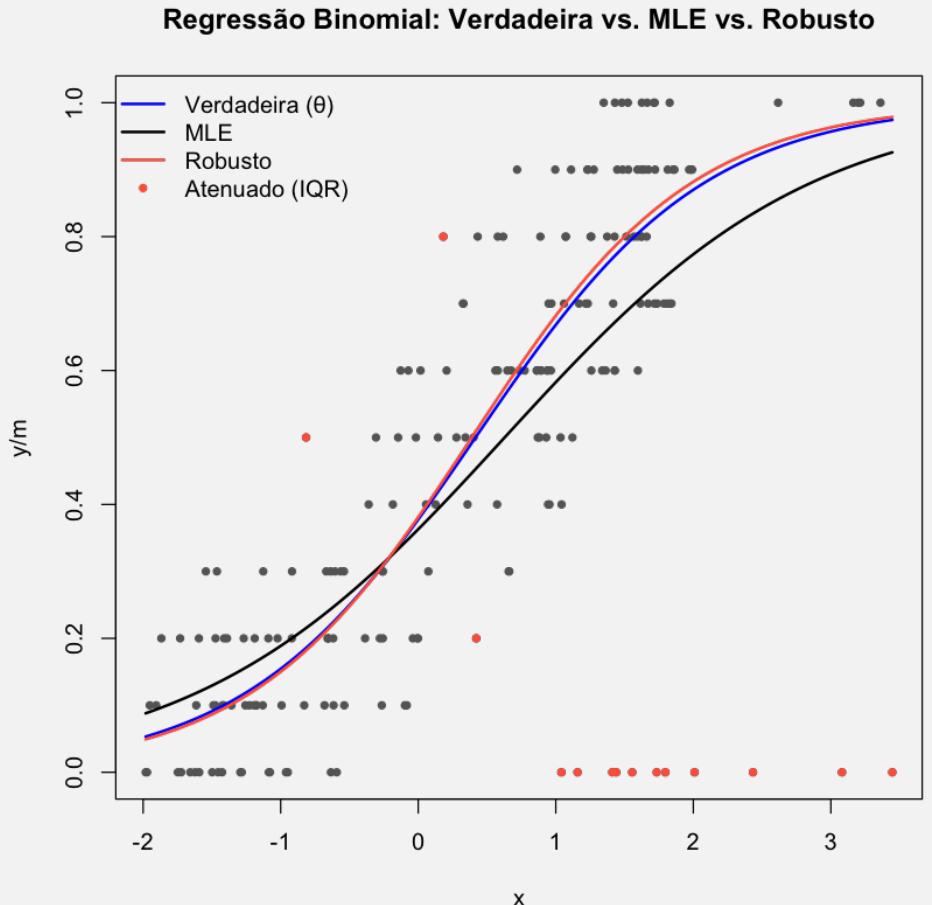


Figura 1: Curvas ajustadas sob contaminação (pontos atenuados em vermelho). Parâmetros: $n = 200$, $m = 10$, $\theta = (-0.5, 1.2)$, contaminação 10%, $p = 1.5$.

O *Listing 4* apresenta o procedimento de simulação de Monte Carlo correspondente, no qual foram realizadas $R = 500$ repetições para comparar o desempenho do estimador de máxima verossimilhança (MLE) com o estimador

robusto proposto. Para cada método, foram calculadas a média, o viés e o erro quadrático médio (RMSE) das estimativas, cujos resultados são apresentados na Tabela 3 abaixo.

Tabela 3: Resumo da simulação Monte Carlo ($R = 500$): comparação entre MLE e estimador robusto.

Estimador	Média	Viés	RMSE
MLE β_0	-0.561	-0.061	0.087
MLE β_1	0.845	-0.355	0.370
Robusto β_0	-0.484	0.016	0.068
Robusto β_1	1.281	0.081	0.104

Os resultados mostram que o MLE apresenta subestimação do parâmetro verdadeiro, com viés negativo em β_1 e maior erro quadrático médio. O estimador robusto, por sua vez, reduziu o viés de β_1 em aproximadamente **77%** e o RMSE em cerca de **72%**. Para β_0 , observou-se uma redução de **74%** no viés e de **22%** no RMSE.

A Figura 2 apresenta as distribuições empíricas das estimativas obtidas nas simulações de Monte Carlo. As curvas vermelhas correspondem às densidades normais ajustadas com base na média e no desvio-padrão empíricos, evidenciando a característica de normalidade assintótica esperada para os estimadores. Observa-se que o estimador robusto manteve as distribuições centradas próximas aos valores verdadeiros, apresentando **menor viés e menor erro quadrático médio (RMSE)** em comparação ao MLE.

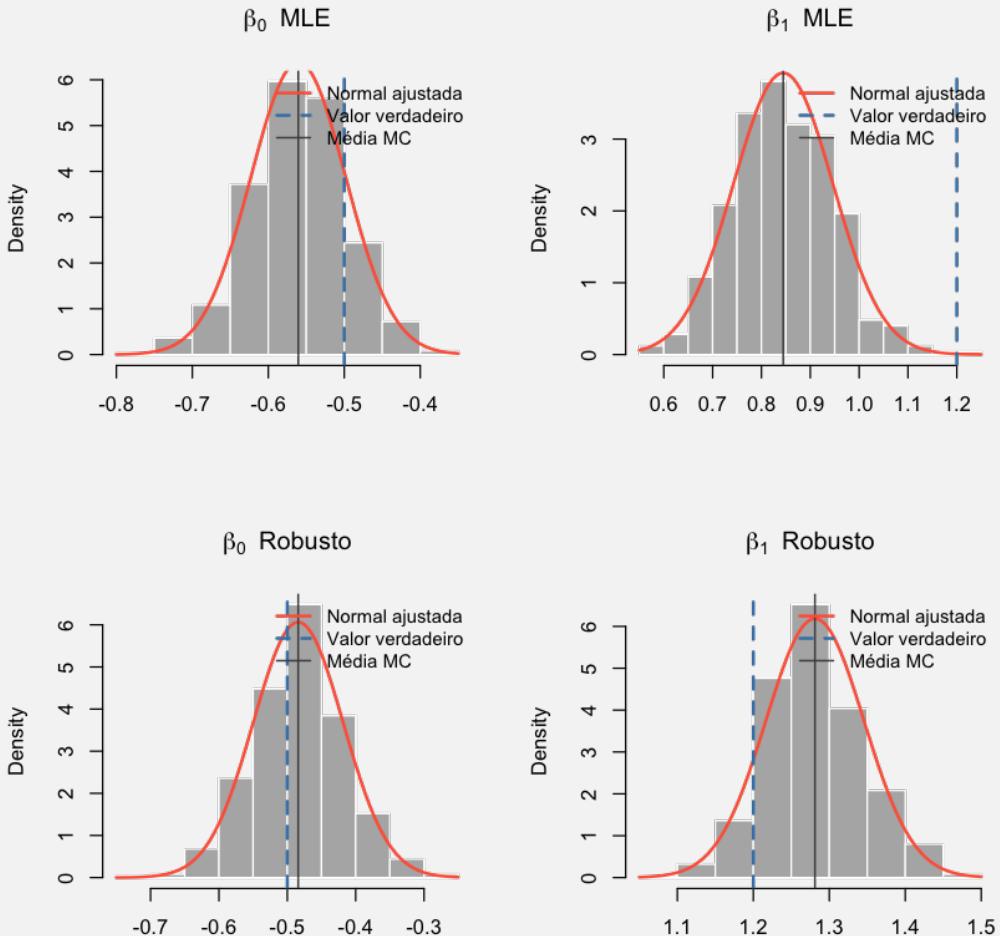


Figura 2: Distribuição das estimativas sob contaminação: comparação entre MLE e estimador robusto.

Conclusão

Segundo a formulação clássica de Fisher da função de máxima verossimilhança (1922) e o limite de eficiência de Cramér–Rao (Cramér, 1946; Rao, 1945), o estimador de máxima verossimilhança (MLE) é assintoticamente eficiente e não-viesado, apresentando variância mínima quando o modelo é corretamente especificado e as observações são independentes e não contaminadas. Notamos que a introdução de 10% de contaminação nos dados foi suficiente para violar essas condições, fazendo com que o MLE apresentasse deslocamento em relação ao valor verdadeiro dos parâmetros — um viés negativo mais evidente em β_1 .

As distribuições empíricas das estimativas obtidas via MLE mostram um deslocamento sistemático em relação à linha tracejada (parâmetro real), enquanto as estimativas robustas permanecem centradas ou mais próximas do valor verdadeiro. As ponderações e atenuações com base no desvio em relação ao intervalo interquartil dos resíduos reduzem com eficiência a influência de observações com maiores desvios de resposta e de alavancagem.

O estimador robusto, portanto, reduziu simultaneamente o viés e a variância das estimativas, levando à minimização do erro quadrático médio (EQM).

```
1 ## =====
2 ##  Regressão Binomial Robusta
3 ## =====
4
5 expit = function(eta) 1/(1+exp(-eta))
6 set.seed(2025)
7
8 ## =====
9 ## DADOS ÚNICO
10 ## =====
11 n = 200
12 m = rep(10, n)
13 x = runif(n, -2, 2)
14
15 theta = c(-0.5, 1.2)          # valores verdadeiros
16 mu = expit(theta[1] + theta[2]*x)
17 y = rbinom(n, size=m, prob=mu)
18
19 ## contaminação
20 aux = sample(which(x < 0.5), round(0.10*n))
21 x[aux] = x[aux] + 3
22 y[aux] = ifelse(runif(length(aux)) < 0.5, 0, m[aux])
23
24 ## MLE de referência
25 fit0 = glm(cbind(y, m-y) ~ x, family=binomial())
26 theta.hat0 = coef(fit0)
27
28 ## ===== Estimador robusto: IQR + exp(-(|r|-c)_+) =====
29 p = 1.5
30
31 L = function(th){
32   eta = th[1] + th[2]*x
33   mu = expit(eta)
34   r = y - m*mu
35   c = p * IQR(r)
36   exc = pmax(0, abs(r) - c)
37   w = exp(-exc)
38   cbind( w*(y - m*mu),
39         w*x*(y - m*mu) )
40 }
41
42 LL = function(th){
43   u = colSums(L(th))
44   sum(u^2)
45 }
46
47 B = function(th){
48   U = L(th)
49   t(U) %*% U / n
50 }
```

```

51 V = function(th){
52   eps = 1e-6
53   U0 = colSums(L(th))
54   J = matrix(NA_real_, 2, 2)
55   for(j in 1:2){
56     thj = th; thj[j] = thj[j] + eps
57     Uj = colSums(L(thj))
58     J[, j] = (Uj - U0)/eps
59   }
60   solve(J) %*% B(th) %*% t(solve(J))
61 }
62
63
64 fit1 = optim(LL, par=theta.hat0, control=list(reltol=1e-12))
65 theta.hati = fit1$par
66 V.hat = V(theta.hati)
67
68 E = eigen(V.hat)
69 V_0.5 = E$vectors %*% diag(1/sqrt(E$values)) %*% t(E$vectors)
70 W1 = sqrt(n) * V_0.5 %*% (theta.hati - theta)
71
72 ## Plot
73 eta.r = theta.hati[1] + theta.hati[2]*x
74 r.now = y - m*expit(eta.r)
75 cut.now= p * IQR(r.now)
76 mask = abs(r.now) > cut.now # TRUE = atenuado
77
78 par(mfrow=c(1,1))
79 plot(y/m ~ x,
80       pch = 20, col = "grey40",
81       main = "Regressão Binomial: Verdadeira vs. MLE vs. Robusto",
82       xlab = "x", ylab = "y/m")
83
84 xs = seq(min(x), max(x), length=200)
85
86 ## 1) curva verdadeira (parâmetros que geraram os dados)
87 lines(xs, expit(theta[1] + theta[2]*xs), lwd=2, col="blue")
88
89 ## 2) MLE e 3) Robusto
90 lines(xs, expit(theta.hat0[1] + theta.hat0[2]*xs), lwd=2, col="black")
91 lines(xs, expit(theta.hati[1] + theta.hati[2]*xs), lwd=2, col="tomato")
92
93 ## pontos atenuados por cima (em vermelho)
94 points(x[mask], (y/m)[mask], pch=20, col="tomato")
95
96 legend("topleft",
97        c("Verdadeira", "MLE", "Robusto", "Atenuado (IQR)"),
98        col = c("blue", "black", "tomato", "tomato"),
99        lwd = c(2, 2, 2, NA),
100       pch = c(NA, NA, NA, 20),
101       bty = "n")
102
103 theta      # verdadeiro
104 theta.hat0 # MLE
105 theta.hati # Robusto
106 W1

```

Listing 3: Implementação analítica do estimador robusto para regressão binomial, com ponderação $w_i = \exp(-(|r_i| - c)_+)$ e corte definido por $c = p \cdot \text{IQR}(r)$.

```

1 #### =====
2 ## MONTE CARLO
3 #### =====
4
5 mc_binom_iqrep = function(R=500, n=200, contam=0.10, p=1.5){
6   out = matrix(NA_real_, R, 4)
7   colnames(out) = c("b0_mle","b1_mle","b0_rob","b1_rob")
8   theta = c(-0.5, 1.2)

```

```

9   m = rep(10, n)
10
11  for(t in 1:R){
12    x = runif(n, -2, 2)
13    mu = expit(theta[1] + theta[2]*x)
14    y = rbinom(n, size=m, prob=mu)
15
16    ## contaminação idêntica ao exemplo
17    aux = sample(which(x < 0.5), round(contam*n))
18    x[aux] = x[aux] + 3
19    y[aux] = ifelse(runif(length(aux)) < 0.5, 0, m[aux])
20
21    ## MLE
22    b_mle = coef(glm(cbind(y, m - y) ~ x, family=binomial()))
23
24    ## Robusto
25    L = function(th){
26      eta = th[1] + th[2]*x
27      mu = expit(eta)
28      r = y - m*mu
29      c = p * IQR(r)
30      exc = pmax(0, abs(r) - c)
31      w = exp(-exc)
32      cbind( w*(y - m*mu),
33             w*x*(y - m*mu) )
34    }
35    LL = function(th){ u = colSums(L(th)); sum(u^2) }
36
37    b_rob = try(optim(LL, par=b_mle, control=list(reltol=1e-12))$par,
38                silent=TRUE)
39    ifinherits(b_rob, "try-error") next
40
41    out[t,] = c(b_mle[1], b_mle[2], b_rob[1], b_rob[2])
42  }
43  out
44}
45
46 set.seed(99)
47 res = mc_binom_iqrexp(R=500, n=200, contam=0.10, p=1.5)
48
49 ## Tabela de resumo
50 summ = function(v, tru) c(mean=mean(v,na.rm=TRUE),
51                         bias=mean(v-tru,na.rm=TRUE),
52                         rmse=sqrt(mean((v-tru)^2,na.rm=TRUE)))
53 TAB = rbind(
54   "MLE beta0" = summ(res[,1], -0.5),
55   "MLE beta1" = summ(res[,2], 1.2),
56   "ROB beta0" = summ(res[,3], -0.5),
57   "ROB beta1" = summ(res[,4], 1.2)
58 )
59 round(TAB, 3)
60
61 ## =====
62 ## PLOTS
63 ## =====
64
65 true_b0 <- -0.5
66 true_b1 <- 1.2
67
68 plot_hist <- function(z, true, titulo){
69   hist(z, prob = TRUE, col = "grey70", border = "white", main = titulo, xlab = "")
70   ## curva normal ajustada aos MC
71   curve(dnorm(x, mean = mean(z, na.rm=TRUE), sd = sd(z, na.rm=TRUE)),
72         add = TRUE, col = "tomato", lwd = 2)
73   ## valor verdadeiro (linha tracejada)
74   abline(v = true, lty = 2, lwd = 2, col = "steelblue")
75   ## média Monte Carlo (opcional)
76   abline(v = mean(z, na.rm=TRUE), lty = 1, lwd = 1.2, col = "grey30")

```

```

77 legend("topright",
78   c("Normal ajustada", "Valor verdadeiro", "Média MC"),
79   col = c("tomato", "steelblue", "grey30"),
80   lty = c(1, 2, 1), lwd = c(2, 2, 1.2), bty = "n", cex = 0.9)
81 }
82
83 par(mfrow = c(2,2))
84 plot_hist(res[,1], true_b0, expression(beta[0]~" MLE"))
85 plot_hist(res[,2], true_b1, expression(beta[1]~" MLE"))
86 plot_hist(res[,3], true_b0, expression(beta[0]~" Robusto"))
87 plot_hist(res[,4], true_b1, expression(beta[1]~" Robusto"))
88

```

Listing 4: Simulação de Monte Carlo para avaliação do desempenho do estimador robusto na regressão binomial ($R = 500$, $n = 200$, contaminação de 10%, $p = 1,5$).

Os códigos desde estudo estão disponibilizados em <<http://academic-codex.github.io/MAE5911-Estatistica-e-Machine-Learning>>.

