Modelos lineares generalizados: linearidade com esteróides

Motivação

Como vimos até aqui, o modelo linear é bastante flexível e poderoso. No entanto, o modelo normal tem uma limitação importante: o suporte da distribuição dos dados, que fica limitado a $\mathbb{R}$. Como muitos fenômenos de interesse podem ser modelados como variáveis aleatórias com suporte restrito (e.g. $(0, 1)$ ou $\mathbb{R}_+$) e também discreto, como é o caso do modelos de contagem.

A solução se encontra na formulação dos chamados **modelos lineares generalizados** (*generalised linear models*, GLM), em que o preditor linear é conectado à esperança condicional por meio de uma função especial, chamada função de ligação.

A estrutura básica de um GLM

Sejam $\boldsymbol{Y} = (Y_1, \ldots, Y_n)$ e $\boldsymbol{X}$ o vetor de variáveis dependentes e a matriz $(n \times P)$ de desenho, respectivamente. Defina $\mu_i(\boldsymbol{X}) = \mu_i := E[Y_i \mid \boldsymbol{X}]$ como a média condicional de cada Y_i. Em um GLM, escrevemos

$$g(\boldsymbol{\mu}) = \boldsymbol{X}\boldsymbol{\beta},$$

onde $g(\cdot)$ é uma função monotônica e diferenciável, chamada de **função de ligação**. Além disso, suponha que cada Y_i tenha distribuição da família exponencial com parâmetro canônico θ_i, isto é,

$$f(y_i; \theta_i) = \exp\left\{y_i\theta_i - b(\theta_i) + c(y_i)\right\}.$$

Logo,

$$\begin{aligned} f(\boldsymbol{y}; \boldsymbol{\theta}) &= \prod_{i=1}^{n} f(y_i; \theta_i), \\ &= \exp\left\{\sum_{i=1}^{n} y_i\theta_i - \sum_{i=1}^{n} b(\theta_i) + \sum_{i=1}^{n} c(y_i)\right\}. \end{aligned}$$

Suponha que g é a função de ligação canônica, isto é, que $g(\mu_i) = \boldsymbol{x}_i^T\boldsymbol{\beta} = \theta_i$. Então, a log-verossimilhança para $\boldsymbol{\beta}$ é

$$
\begin{aligned}
\ell(\boldsymbol{\beta}) &= \sum_{i=1}^{n} y_i\theta_i - \sum_{i=1}^{n} b(\theta_i) + \sum_{i=1}^{n} c(y_i), \\
&= \sum_{i=1}^{n} y_i\boldsymbol{x}_i^T\boldsymbol{\beta} - \sum_{i=1}^{n} b(\boldsymbol{x}_i^T\boldsymbol{\beta}) + \sum_{i=1}^{n} c(y_i).
\end{aligned}
$$

Em notação matricial, temos

$$
\ell(\boldsymbol{\beta}) = \boldsymbol{y}^T\boldsymbol{X}\boldsymbol{\beta} - \mathbf{1}^T b(\boldsymbol{X}\boldsymbol{\beta}) + \mathbf{1}^T c(\boldsymbol{y}),
$$

onde $\mathbf{1}$ é um vetor de uns de dimensão n. Logo,

$$
\begin{aligned}
\frac{\partial \ell(\boldsymbol{\beta})}{\partial \beta_k} &= \sum_{i=1}^{n} y_i x_{ik} - \sum_{i=1}^{n} x_{ik} b'(\boldsymbol{x}_i^T\boldsymbol{\beta}), \\
\frac{\partial^2 \ell(\boldsymbol{\beta})}{\partial \beta_k \partial \beta_l} &= -\sum_{i=1}^{n} x_{ik} x_{il} b''(\boldsymbol{x}_i^T\boldsymbol{\beta}),
\end{aligned}
$$

ou seja,

$$
\begin{aligned}
\nabla \ell(\boldsymbol{\beta}) &= \boldsymbol{X}^T(\boldsymbol{y} - b'(\boldsymbol{X}\boldsymbol{\beta})), \\
\nabla^2 \ell(\boldsymbol{\beta}) &= -\boldsymbol{X}^T \mathrm{diag}\left\{b''(\boldsymbol{X}\boldsymbol{\beta})\right\} \boldsymbol{X}.
\end{aligned}
$$

Ajustando um GLM: métodos numéricos

Para estimar $\boldsymbol{\beta}$ por máxima verossimilhança, podemos usar o método de Newton-Raphson ou o método de Fisher scoring. Dado um valor inicial $\boldsymbol{\beta}^{(0)}$, a iteração t do primeiro é dada por

$$
\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} - \left[\nabla^2 \ell(\boldsymbol{\beta}^{(t)})\right]^{-1} \nabla \ell(\boldsymbol{\beta}^{(t)}),
$$

enquanto a iteração t do segundo é dada por

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} + \left[\mathcal{I}(\boldsymbol{\beta}^{(t)})\right]^{-1} \nabla\ell(\boldsymbol{\beta}^{(t)}),$$

onde $\mathcal{I}(\boldsymbol{\beta}^{(t)})$ é a informação de Fisher. Se a função de ligação for canônica, então os métodos são equivalentes.

Para maior eficiência computacional, podemos definir $\boldsymbol{X}_t = \text{diag}\left\{b''(\boldsymbol{X}\boldsymbol{\beta}^{(t)})\right\}^{1/2}\boldsymbol{X}$ e obter a decomposição QR, $\boldsymbol{X}_t = Q_t R_t$. Definindo $w_t = b''(\boldsymbol{X}\boldsymbol{\beta}^{(t)})$ e $z_t = \boldsymbol{y} - b'(\boldsymbol{X}\boldsymbol{\beta}^{(t)})$, é possível mostrar que

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} + R_t^{-1} Q_t^T \left\langle \text{diag}\{w_t\}^{-1/2}, z_t \right\rangle,$$

o que resulta no seguinte sistema linear:

$$R_t(\boldsymbol{\beta}^{(t+1)} - \boldsymbol{\beta}^{(t)}) = Q_t^T \left\langle \text{diag}\{w_t\}^{-1/2}, z_t \right\rangle.$$

Exemplo: regressão Poisson

Considere $Y_i \mid \boldsymbol{X} \sim \text{Poisson}(\theta_i)$. Primeiro, vamos expressar a f.d.p. em termos da família exponencial. Temos

$$\begin{aligned} f(y;\theta) &= \frac{\exp\{-\theta\}\theta^y}{y!}, \\ &= \exp\left\{y\log(\theta) - \theta - \log(y!)\right\}, \end{aligned}$$

ou seja, o parâmetro canônico é $\eta = \log(\theta)$, $b(\eta) = \exp(\eta)$ e $c(y) = -\log(y!)$. Lembre-se que a função de ligação canônica é aquela que conecta o parâmetro canônico η_i com μ_i e $\boldsymbol{x}_i^T\beta$ de modo que $\eta_i = \boldsymbol{x}_i^T\beta$. Logo, como $\mu_i = \exp(\eta_i)$, temos que $g(t) = \log(t)$ e, portanto, $\mu_i = \exp(\boldsymbol{x}_i^T\beta)$.

Agora vamos obter a função score e a Hessiana, necessárias para a estimação dos parâmetros. Pelos cálculos anteriores, a função score é dada por

$$\nabla\ell(\boldsymbol{\beta}) = \boldsymbol{X}^T(\boldsymbol{y} - \exp(\boldsymbol{X}\boldsymbol{\beta}))$$

e a Hessiana,

$$\nabla^2 \ell(\boldsymbol{\beta}) = -\boldsymbol{X}^T \text{diag}\left\{\exp(\boldsymbol{X}\boldsymbol{\beta})\right\} \boldsymbol{X}.$$

Por fim, basta usar o método de Newton-Raphson:

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} + \left[\boldsymbol{X}^T \text{diag}\left\{\exp(\boldsymbol{X}\boldsymbol{\beta}^{(t)})\right\} \boldsymbol{X}\right]^{-1} \boldsymbol{X}^T(\boldsymbol{y} - \exp(\boldsymbol{X}\boldsymbol{\beta}^{(t)})),$$

o que leva ao seguinte sistema, como visto anteriormente:

$$R_t(\boldsymbol{\beta}^{(t+1)} - \boldsymbol{\beta}^{(t)}) = Q_t^T \left\langle \exp\left(-\frac{1}{2}\boldsymbol{X}\boldsymbol{\beta}^{(t)}\right), \boldsymbol{y} - \exp(\boldsymbol{X}\boldsymbol{\beta}^{(t)})\right\rangle.$$

Para fazer isso no R, vamos implmentar uma função que ajusta um modelo GLM para distribuições com um parâmetro:

```r
# função para ajustar um GLM para distribuições com um parâmetro
glm1 <- function(y, X, bp, bpp) {
  # pega o número de variáveis
  p <- NCOL(X)
  # inicializa o vetor de parâmetros
  beta_k <- rep(0, p)
  # inicializa a lista de parâmetros
  list_beta <- list()
  # critério de parada
  stop_error <- 1e-6
  # inicializa o contador
  j <- 1L
  # inicializa o erro
  current_error <- 1

  while (current_error > stop_error) {
    # define variaveis auxiliares
    eta_k <- X %*% beta_k
    z_k <- y - bp(eta_k)
    w_k <- bpp(eta_k)
    X_k <- drop(w_k^(1/2)) * X
    wz_k <- w_k^(-1/2) * z_k
```

```r
    # calcula a decomposição QR
    qr_out <- qr(X_k)
    Q_k <- qr.Q(qr_out)
    R_k <- qr.R(qr_out)
    # calcula a solução do sistema
    a_k <- backsolve(R_k, crossprod(Q_k, wz_k))

    # guarda o valor de beta_k
    list_beta[[j]] <- a_k + beta_k
    # atualiza o erro
    current_error <- max(abs(beta_k - list_beta[[j]]))
    # atualiza o valor de beta_k
    beta_k <- list_beta[[j]]
    # atualiza o contador
    j <- j + 1L
  }

  do.call(cbind, list_beta)
}
```

Agora, vamos construir uma função para o GLM Poisson:

```r
# função para ajustar um GLM Poisson
poiReg <- function(formula, data) {
  # constroi o modelo
  mf <- model.frame(formula, data = data)
  # pega a variável resposta
  y <- model.response(mf)
  # pega a matriz de desenho
  X <- model.matrix(formula, mf)
  # define a derivada de b(eta)
  bp <- function(theta) exp(theta)
  # define a segunda derivada de b(eta)
  bpp <- function(theta) exp(theta)
  glm1(y, X, bp, bpp)
}
```

Para checar as funções, vamos simular dados::

```r
# define a semente
set.seed(20032025)
```

```r
# define os parâmetros
n <- 500
X <- cbind(1, rnorm(n), runif(n))
betas <- c(1, -0.5, 0.5)
eta <- drop(X %*% betas)
lambda <- exp(eta)
# simula os dados
y <- rpois(n, lambda)
sim_data <- data.frame(y = y, x1 = X[, 2], x2 = X[, 3])
```

Agora, vamos comparar os resultados:

```r
# ajusta o modelo com a função poiReg
out_poiReg <- poiReg(formula = y ~ x1 + x2, data = sim_data)
# ajusta o modelo com a função nativa glm
out_glm <- glm(formula = y ~ x1 + x2, data = sim_data, family = "poisson")
cbind(true = betas,
      logReg = out_poiReg[, ncol(out_poiReg)],
      glm = coef(out_glm))
```

```
            true    logReg        glm
(Intercept)  1.0  0.9385882  0.9385882
x1          -0.5 -0.5159732 -0.5159732
x2           0.5  0.5943188  0.5943188
```

Exercícios de fixação: Regressão logística

Seja $Y_i \mid \boldsymbol{X} \sim \text{Bernoulli}(\theta_i)$, com $E[Y_i \mid \boldsymbol{X}] = \mu_i = g^{-1}(\boldsymbol{x}_i^\top \boldsymbol{\beta})$.

1. Mostre que a distribuição de Y_i pertence à família exponencial e encontre a função de ligação canônica.
2. Encontre a log-verossimilhança e exiba a função score e a Hessiana.
3. Mostre como obter o estimador de máxima verossimilhança para $\boldsymbol{\beta}$.
4. Implemente uma função para ajustar uma regressão logística e compare os resultados com a função `glm` usando dados de sua escolha.
5. (**Desafio**) Mostre como obter intervalos de confiança para $\boldsymbol{\beta}$ usando o método de Wald – ver Seção 5.4 de Dobson (2018). Calcule os intervalos para um exemplo empírico e compare com o output da função `confint()` do R.

Referências

- Dobson, A. J., & Barnett, A. G. (2018). An introduction to generalized linear models. CRC press. (Caps 3, 4 e 5)
- Gelman, A., Hill, J., & Vehtari, A. (2020). Regression and other stories. Cambridge University Press. (Cap 15)