Implementação Bayesiana via Stan do modelo de regressão logística

Prof. Vinícius D. Mayrink - Departamento de Estatística - ICEx - UFMG

Estatística Bayesiana - 1º Semestre de 2025

Instale o *software* `R` e o pacote `rstan` em seu computador. Para maiores informações sobre o `Stan` visite a página `mc-stan.org`.

Limpe a área de trabalho do `R` e carregue o pacore `rstan`. Use os comandos inseridos na caixa exibida abaixo para esta tarefa.

```
rm(list=ls(all=TRUE))
library(rstan)
# comando para evitar recompilar.
rstan_options(auto_write = TRUE)
# comando para executar diferentes cadeias em paralelo.
options(mc.cores = parallel::detectCores())
# Fixando semente para garantir reproducibilidade.
set.seed(2019)
```

Introdução

A regressão logística é um tipo de modelo linear generalizado construído para lidar com variáveis respostas do tipo binária (1 = ocorrência do evento de interesse, 0 = não ocorrência). O objetivo é avaliar o impacto de um grupo de K covariáveis $X_{1i}, X_{2i}, \cdots, X_{Ki}$ sobre a probabilidade do evento resposta codificado como $Y_i = 1$; assuma $i = 1, \cdots, n$. Assim como estudado na regressão linear múltipla, adote novamente a notação: $\boldsymbol{Y} = (Y_1, Y_2, \cdots, Y_n)^\top$ e a matriz de covariáveis estruturada como segue:

$$\boldsymbol{X} = \begin{bmatrix} 1 & X_{11} & X_{21} & \cdots & X_{K1} \\ 1 & X_{12} & X_{22} & \cdots & X_{K2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & X_{1n} & X_{2n} & \cdots & X_{Kn} \end{bmatrix}.$$

O modelo estatístico é escrito da seguinte forma:

$$Y_i \sim \text{Bernoulli}(\theta_i) \quad \text{com} \quad \ln\left(\frac{\theta_i}{1-\theta_i}\right) \;=\; \beta_0 + \beta_1 X_{1i} + \cdots + \beta_K X_{Ki} \;=\; \boldsymbol{X}_{i\bullet}\beta.$$

Recorde que $\beta = (\beta_0, \beta_1, \cdots, \beta_K)^\top$ e $\boldsymbol{X}_{i\bullet}$ é o vetor localizado na i-ésima linha da matriz $\boldsymbol{X}$. O modelo de regressão sob estudo é composto por $q = K + 1$ coeficientes, sendo um deles o intercepto

β_0. Um elemento importante a ser enfatizado na formulação acima é a razão $\theta_i/(1-\theta_i)$, a qual é comumente chamada de *odds*. Valores da *odds* maiores do que 1, sugerem $P(Y_i = 1) > P(Y_i = 0)$. Por outro lado, valores da *odds* $\in (0,1)$, indicam $P(Y_i = 1) < P(Y_i = 0)$. Note que a *odds* é sempre positiva, consequentemente a log-*odds* será um número real (negativo ou positivo). A função log tem um papel de ligação, ou seja, ela estabelece uma conexão entre o preditor linear e a probabilidade de sucesso θ_i do indivíduo i. A sequência de cálculos desenvolvida abaixo determina um resultado final que ajuda a entender melhor a relação do preditor linear com a probabilidade θ_i:

$$\ln\left(\tfrac{\theta_i}{1-\theta_i}\right) \;=\; \boldsymbol{X}_{i\bullet}\beta, \qquad \tfrac{\theta_i}{1-\theta_i} \;=\; \exp\left\{\boldsymbol{X}_{i\bullet}\beta\right\}, \qquad \theta_i \;=\; \exp\left\{\boldsymbol{X}_{i\bullet}\beta\right\} - \theta_i \exp\left\{\boldsymbol{X}_{i\bullet}\beta\right\},$$

$$\theta_i\Big(1+\exp\left\{\boldsymbol{X}_{i\bullet}\beta\right\}\Big) \;=\; \exp\left\{\boldsymbol{X}_{i\bullet}\beta\right\}, \qquad \theta_i \;=\; \tfrac{\exp\left\{\boldsymbol{X}_{i\bullet}\beta\right\}}{1+\exp\left\{\boldsymbol{X}_{i\bullet}\beta\right\}}.$$

Os parâmetros de interesse na regressão logística são basicamente os coeficientes de regressão em $\beta = (\beta_0, \beta_1, \cdots, \beta_K)^\top$. Não há um termo de erro ϵ_i adicionado ao preditor linear nesta modelagem. No modelo de regressão linear múltipla a variabilidade era governada pela distribuição Normal que tinha uma variância a ser estimada. Na regressão logística, esta variabilidade é estabelecida através da distribuição Bernoulli; a variância da Bernoulli é $\theta_i(1-\theta_i)$. Os coeficientes em β são desconhecidos e, portanto, na inferência Bayesiana especificamos uma distribuição *a priori* para descrever nossa incerteza inicial sobre eles. Admita que, dado β, temos independência condicional entre $Y_1, Y_2, \cdots, Y_n$.

Em termos de verossimilhança do modelo logístico, a seguinte expressão é estabelecida para o i-ésimo indivíduo:

$$f_{Y_i|\beta,\boldsymbol{X}_{i\bullet}}(y_i) \;=\; (\theta_i)^{y_i}\;(1-\theta_i)^{1-y_i} \quad \text{com} \quad \theta_i \;=\; \frac{\exp\left\{\boldsymbol{X}_{i\bullet}\beta\right\}}{1+\exp\left\{\boldsymbol{X}_{i\bullet}\beta\right\}}.$$

Por independência condicional, escreve-se:

$$f_{\boldsymbol{Y}|\beta,\boldsymbol{X}}(\boldsymbol{y}) \;=\; \prod_{i=1}^{n} (\theta_i)^{y_i}\;(1-\theta_i)^{1-y_i} \quad \text{com} \quad \theta_i \;=\; \frac{\exp\left\{\boldsymbol{X}_{i\bullet}\beta\right\}}{1+\exp\left\{\boldsymbol{X}_{i\bullet}\beta\right\}}.$$

Por simplicidade de notação a ser usada mais adiante, denote: $\theta = (\theta_1, \theta_2, \cdots, \theta_n)^\top$. O próximo passo do nosso esquema de estudo, configurado neste material, é a geração de dados artificiais a serem ajustados pelo modelo Bayesiano.

Gerando dados artificiais

Esta análise é focada na investigação de dados sintéticos gerados com base na estrutura do modelo de regressão logística. Valores verdadeiros dos coeficientes de regressão são indicados na próxima caixa. Lembre-se que durante o ajuste Bayesiano via `Stan`, iremos ignorar o conhecimento destes valores reais. Na análise dos resultados *a posteriori*, uma comparação "real versus estimado" será feita para julgamento da qualidade da estimação.

O tamanho amostral e os valores reais dos parâmetros são os seguintes:

```r
n = 200                                    # Tamanho amostral.
beta = c(0.5, 0.7, -0.7, 1.0, -1.0)  # Coeficientes reais.
q = length(beta)                           # Número de coeficientes.
b_real = beta                              # Objeto contendo valores reais.
```

Na próxima etapa, geramos as covariáveis. Assim como no modelo linear normal, adote a primeira covariável sendo do tipo binária. As demais são contínuas e provenientes da distribuição $U(-1, 1)$.

```r
x = array(1, c(n, q))                      # Matriz de 1's.
x[,2] = rbinom(n, 1, 0.5)                  # Covariável binária.
for(i in 3:q){ x[,i] = runif(n, -1, 1) }  # Covariáveis contínuas.
```

Uma vez obtida a matriz de covariáveis, calcule os valores reais de θ_1, θ_{100} e θ_{200} para uma investigação de qualidade de ajuste a ser feita após executar o MCMC.

```r
# Valor real de theta[1], theta[100] e theta[200]
t_real = rep(0,3)
t_real[1] = exp(x[1,] %*% b_real) / (1 + exp(x[1,] %*% b_real))
t_real[2] = exp(x[100,] %*% b_real) / (1 + exp(x[100,] %*% b_real))
t_real[3] = exp(x[200,] %*% b_real) / (1 + exp(x[200,] %*% b_real))
```

A terceita etapa do processo de geração de dados sintéticos diz respeito à simulação da variável resposta binária. A próxima caixa mostra os comandos de geração.

```r
y = numeric(n); theta = numeric(n);     # Vetores de tamanho n.
for(i in 1:n){
 aux = x[i,] %*% beta                    # Preditor linear.
 theta[i] = exp(aux) / (1 + exp(aux))  # Probabilidade de sucesso.
 y[i] = rbinom(1, 1, theta[i])          # Variável resposta Bernoulli.
}
```

Após gerar os dados, a próxima seção descreve a escolha das distribuições *a priori* para $\beta_0, \beta_1, \cdots, \beta_K$.

Especificações a priori

A informação inicial sobre $\beta = (\beta_0, \beta_1, \cdots, \beta_k)^\top$ será expressa através da distribuição Normal Multivariada $\beta \sim N_q(m_\beta, S_\beta)$. A mesma discussão sobre esta escolha multivariada, simétrica e com suporte real, apresentada no material sobre a regressão linear múltipla, é também válida aqui. A escolha da normalidade para os coeficiente é adequada, pois este tipo de parâmetro pode assumir tanto valores negativos quanto positivos. A flexibilidade de formatos, estabelecidos pelas escolhas da média e variância, faz da normal uma opção sempre interessante para descrever nossa incerteza sobre um coeficiente de regressão. Os hiperparâmetros da distribuição *a priori* são:

```r
# Normal Multivariada.
m_beta = rep(0, q)     # Vetor de médias.
S_beta = 10 * diag(q)  # Matriz de covariâncias.
```

A próxima seção organiza as informações requeridas para executar o NUTS através do `Stan`.

Transmitindo informações para o Stan

O passo 1 envolve organizar uma listagem contendo: tamanho amostral, dados e hiperparâmetros *a priori*. O código para esta tarefa é simples, veja a próxima caixa.

```
data = list(n = n, q = q, y = y, x = x, m_beta = m_beta, S_beta = S_beta)
```

O passo 2 requer a definição da lista com os nomes dos parâmetros a serem salvos na execução do NUTS. Salve todos parâmetros para os quais há interesse em realizar uma ánlise das estimativas *a posteriori*.

```
# Lista requisitando que o vetores beta e theta sejam salvos.
# Lembrete: neste exemplo, são 5 betas e 200 thetas.
pars = c("beta", "theta")
```

O passo 3 é a especificação de sementes de inicialização do MCMC. Iremos optar pela análise de 2 cadeias para cada parâmetro, então considere a lista escrita na p̂roxima caixa. Valores iniciais são definidos apenas para os parâmetros que receberão alguma distribuição *a priori*. Note que, qualquer θ_i é escrito em função de β, então θ_i não recebe uma distribuição de probabilidade *a priori* diretamente. Isto implica que não há a necessidade de definir semente para θ_i.

```
# Lista de sementes de inicialização (admite 2 cadeias):
init = list()
init[[1]] = list(beta = rep(0, q))
init[[2]] = list(beta = runif(q, -1, 1))

# Alternativamente, pode-se especificar:
# init = "random"
# init = "0"
```

O passo 4 é indicar o número de iterações do MCMC, período de aquecimento e o número de cadeias. Considere os valores reportados a seguir.

```
iter = 2000     # Total de iterações (incluindo burn-in).
warmup = 1000   # Número de iterações do burn-in.
chains = 2      # Número de cadeias do MCMC.
# chains = 1 (Stan retorna amostras a posteriori após burn-in da cadeia única).
# chains > 1 (Stan junta as amostras após burn-in de todas as cadeias).
```

O código `Stan`, mostrado abaixo, contém a verossimilhança e as distribuições *a priori* dos coeficientes. Este código será salvo no arquivo `RegLogistica.stan`. Recomenda-se que o usuário salve este arquivo na mesma pasta escolhida como diretório de trabalho do `R`. É possível escolher o diretório de trabalho do `R` por meio do comando `setwd()`.

```
// Bloco de declaração de dados.
// Declare aqui todos os objetos passados do R para o Stan.
// Estes objetos são aqueles dentro da listagem "data".
data{
  int<lower=1> n;
  int<lower=1> q;
```

```
  int<lower=0,upper=1> y[n];
  matrix[n,q] x;
  vector[q] m_beta;
  matrix[q,q] S_beta;
}

// Bloco de declaração de parâmetros.
// Declare aqui todos os parâmetros para os quais
// uma distribuição a priori é especificada.
parameters{
  vector[q] beta;
}

// Bloco de parâmetros transformados.
// Se necessário, declare aqui novos parâmetros
// construídos como função daqueles
// declarados no bloco anterior.
transformed parameters{
  vector[n] theta;
  for(i in 1:n){
    theta[i] = exp(x[i,] * beta) / (1 + exp(x[i,] * beta));
  }
}

// Bloco do modelo.
// Defina aqui a verossimilhança e as distribuições a priori.
model{
  // Verossimilhança
  for(i in 1:n){ y[i] ~ bernoulli(theta[i]); }

  // Priori 1: Normal Multivariada com
  // vetor de médias e matriz de covariâncias.
  beta ~ multi_normal(m_beta, S_beta);

}
// Deixe vazia a última linha do arquivo ".stan" (isso evita "warnings").
```

Finalmente as informações estão devidamente organizadas para transmissão ao `Stan`. Use o comando a seguir no `R` para requisitar a execução do NUTS. Note que o argumento `file` agora invoca o arquivo `RegLogistica.stan`.

```
output = stan(file = "RegLogistica.stan", data = data,
              iter = iter, warmup = warmup, chains = chains,
              pars = pars, init = init, verbose = FALSE)
```

Explorando os resultados

Os códigos `R` exibidos nesta seção são destinados a desenvolver uma análise exploratória das amostras geradas para formar as cadeias de Markov após *burn-in*. O objeto `output`, salvo na área de tabalho do `R`, contém os resultados do ajuste Bayesiano. Este objeto é da classe `stanfit`, o qual é um formato particular estabelecido pelo `rstan` para guardar informações de saída do NUTS.

O sumário descritivo, gerado pela função `print` para o objeto `output` (classe `stanfit`), é obtido em duas versões conforme mostra a próxima caixa.

```
# Sumário global do objeto stanfit.
print(output, pars = c("beta"))
# Sumário focado em beta2, beta3, theta[1], theta[100] e theta[200].
print(output, pars = c(paste0("beta[",c(2,3),"]"),
                       paste0("theta[",c(1,100,200),"]")))
```

O gráfico sequencial das cadeias de Markov ao longo das iterações e após *burn-in* é facilmente construído por meio do comando `traceplot`. As duas cadeias solicitadas ao `Stan` aparecerão sobrepostas e com cores diferentes.

```
traceplot(output, pars = c("beta", "theta[1]"))
```

Para uma análise inferencial mais aprofundada, extraia do objeto `output` (tipo `stanfit`) uma ou mais matrizes contendo em suas colunas a junção das duas cadeias de Markov geradas para cada parâmetro.

```
# Extração em formato de lista;
# beta e theta em matrizes separadas na lista.
samp = extract(output)

# Extração alternativa em formato de matriz;
# beta e theta juntos na mesma matriz.
# samp = as.matrix(output)
```

Os próximos gráficos avaliam as densidades estimadas de β_0 e θ_1. Estas densidades acompanham o formato dos histogramas *a posteriori*. O valor real destes parâmetros é identificado através de uma linha vertical vermelha.

```
par(mfrow = c(1,2))

{ plot( density(samp$beta[,1]), cex.lab = 1.5, cex.axis = 1.5, lwd = 2,
      main = "Densidade a posteriori de beta0", col = "blue" )
abline( v = b_real[1], lwd = 2, col = "red" ) }

{ plot( density(samp$theta[,1]), cex.lab = 1.5, cex.axis = 1.5, lwd = 2,
      main = "Densidade a posteriori de theta[1]", col = "blue")
abline( v = t_real[1], lwd = 2, col = "red" ) }
```

A próxima caixa de comandos mostra como os *traceplots* das cadeias podem ser alternativamente construídos a partir das matrizes extraídas do objeto `stanfit`. Destaca-se que o gráfico da cadeia exibido neste resultado é uma junção das duas cadeias solicitadas ao `Stan`.

```
par(mfrow = c(1,2))

{ plot( samp$beta[,1], type = "l", cex.lab = 1.5, cex.axis = 1.5,
        xlab = "iterações", ylab = "beta0",
        main = "Traceplot de beta0", col = "blue" )
abline( h = b_real[1], lwd = 2, col = "red" ) }

{ plot( samp$theta[,1], type = "l", cex.lab = 1.5, cex.axis = 1.5,
      xlab = "iterações", ylab = "theta[1]",
      main = "Traceplot de theta[1]", col = "blue")
abline( h = t_real[1], lwd = 2, col = "red" ) }
```

O código a seguir é destinado a criar uma tabela sumarizadora com os principais resultados de inferência pontual e intervalar *a posteriori*. Os intervalos de credibilidade HPD de 95% são obtidos por meio do pacote `coda`.

```
require(coda)

aux = cbind( samp$beta, samp$theta[,c(1,100,200)] )
me = apply(aux, 2, mean)     # média
md = apply(aux, 2, median)   # mediana
sd = apply(aux, 2, sd)       # desvio padrão
aux = as.mcmc(aux)
hpd = HPDinterval(aux)
tab = cbind(c(b_real,t_real), me, md, sd, hpd[,"lower"], hpd[,"upper"])
rownames(tab) = c( paste0("beta",0:(q-1)), paste0("theta[",c(1,100,200),"]") )
colnames(tab) = c("true", "mean", "median", "s.d.", "HPD_inf", "HPD_sup")
round(tab,4) # mostrar saída com 4 casas decimais.
```

Exercício

Os gerentes de uma empresa resolveram desenvolver uma pesquisa coletando dados sobre o número de defeitos observados na superfície de um tipo de peça produzida pelo setor de fabricação. Suponha que a variável Y_i representa o número de defeitos registrados na peça i (Y_i é uma contagem e seus valores possíveis são $0, 1, 2, \cdots$). Dados referentes a uma amostra de tamanho $n = 300$ foram coletados, ou seja, $i = 1, 2, \cdots, 300$. Além de Y_i, a base de dados também contém duas covariáveis: X_{1i} = covariável binária (1 = usou maquinário novo na fabricação, 0 = usou maquinário antigo) e X_{2i} = anos de experiência do funcionário que operou a máquina de fabricação (unidade de medida: anos/10). Os dados relativos a este experimento estão disponíveis no arquivo `DadosDefeitos.txt` (veja o Moodle do curso).

Para trabalhar com este problema, admita que $Y_i \sim \text{Poisson}(\theta_i)$, sendo $\theta_i > 0$ a média de defeitos esperados na superfície de uma peça i. Iremos utilizar a regressão Poisson (que é um modelo linear generalizado) para estabelecer uma relação entre as covariáveis (X_{1i},X_{2i}) e a resposta Y_i. O modelo é escrito como segue:

$$Y_i \sim \text{Poisson}(\theta_i) \quad \text{com} \quad \ln(\theta_i) = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i}.$$

A função log aparece nesta modelagem com o papel de ligação entre o preditor linear $\beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i}$ e a média de defeitos θ_i. A média de defeitos é sempre positiva, então $\ln(\theta_i) \in \mathbb{R}$ é uma configuração mais adequada para representar um preditor linear que assume valores reais negativos ou positivos. Note que podemos escrever também:

$$\theta_i \;=\; \exp\{\beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i}\}.$$

Continue usando a notação estabelecida nas aulas sobre o modelo logístico, ou seja, $\beta = (\beta_0, \beta_1, \beta_2)^\top$, $\theta = (\theta_1, \theta_2, \cdots, \theta_{300})^\top$, $\boldsymbol{Y} = (Y_1, Y_2, \cdots, Y_{300})^\top$ e $\boldsymbol{X}$ é a matriz

$$\boldsymbol{X} = \begin{bmatrix} 1 & X_{11} & X_{21} \\ 1 & X_{12} & X_{22} \\ \vdots & \vdots & \\ 1 & X_{1n} & X_{2n} \end{bmatrix}.$$

O termo $\boldsymbol{X}_{i\bullet}$ é a i-ésima linha da matriz $\boldsymbol{X}$ e, consequentemente, $\boldsymbol{X}_{i\bullet}\beta = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i}$.

Em termos de verossimilhança, neste problema escrevemos a seguinte expressão para o indivíduo i:

$$f_{Y_i|\beta,\boldsymbol{X}_{i\bullet}}(y_i) \;=\; \frac{\theta_i^{y_i}}{y_i!}\exp\{-\theta_i\} \quad \text{com} \quad \theta_i = \exp\{\boldsymbol{X}_{i\bullet}\beta\}.$$

Sob independência condicional, escreve-se a conjunta:

$$f_{\boldsymbol{Y}|\beta,\boldsymbol{X}}(\boldsymbol{y}) \;=\; \prod_{i=1}^{300} \frac{\theta_i^{y_i}}{y_i!}\exp\{-\theta_i\} \quad \text{com} \quad \theta_i = \exp\{\boldsymbol{X}_{i\bullet}\beta\}.$$

Assim como na regressão logística, os coeficientes de regressão são os nosso parâmetros de interesse neste problema. Iremos adotar aqui a mesma especificação *a priori* normal multivariada definida nos outros modelos de regressão investigados neste curso, ou seja, $\beta \sim N_3(\mathbf{0}_3, 10\ \boldsymbol{I}_3)$ sendo $\mathbf{0}_3 = (0, 0, 0)^\top$.

<u>Tarefa:</u> Tomando como base os códigos do modelo logístico, escreva a versão relacionada ao modelo Poisson e execute o NUTS via `Stan` para estimar β_0, β_1, β_2, θ_1, θ_{150} e θ_{300}. Faça todos os gráficos sugeridos nas análises do modelo logístico e comente os seus resultados.

Observações: Este trabalho deve ser entregue no Moodle em formato de arquivo PDF. O prazo é até 26/06/2025 às 07h:20 da manhã. Vários alunos serão sorteados na aula do dia 26/06/2025 para apresentar este trabalho (a aula inteira será de sorteios com apresentações). Aluno ausente ou que não souber explicar o trabalho entregue, perderá pontos.