模拟：牛顿迭代法求解 Logistic 回归模型 —— 广义线性模型（2021，春）

龚梓阳　220071400021

meetziyang@outlook.com

考虑如下 Logistic 回归模型，

$$Y \sim \text{Bern}\left(g\left(\hat{\beta}_0 + \beta_1 X_1 + \beta_2 X_2\right)\right)$$

，其中 $g(z) = \frac{\exp(z)}{1+\exp(z)}$，$X_1 \sim N(0,1)$，$X_2 \sim \text{Bern}(0.5)$。

试使用牛顿迭代法给出当 $\beta_0 = 0, \beta_1 = 1, \beta_2 = 1.5$ 时该模型的参数估计。

证明. 参考课上所给定的符号与假定，

$$Y_i \sim b\left(m_i, \pi_i\right), i = 1, 2, \ldots, n$$

，其中连接函数为 $\eta_i = g\left(\pi_i\right) = \log\left(\frac{\pi_i}{1-\pi_i}\right) = \sum_{j=1}^{p} \beta_j x_{ij}$，且 $E(Y_i) = \mu_i = m_i \pi_i$。

因此，

$$\begin{aligned}
\ell(\beta) &= \log[f(\pi \mid y)] = \sum^{n} l_i(\beta) \\
&= \sum_{i=1}^{n}\left[y_i \log\left(\frac{\pi_i}{1-\pi_i}\right) + m_i \log\left(1-\pi_i\right)\right] + \sum_{i=1}^{n} \log\left[\binom{m_i}{y_i}\right]
\end{aligned}$$

对其求一阶导及二阶导可得

$$\begin{aligned}
\frac{\partial \ell(\beta)}{\partial \beta_r} &= \sum_{i=1}^{n}\left(y_i - m_i \pi_i\right) x_{ir} \\
\frac{\partial^2 \ell(\beta)}{\partial \beta_s \partial \beta_r} &= -\sum_{i=1}^{n} m_i \pi_i \left(1-\pi_i\right) x_{is} x_{ir}
\end{aligned}$$

写成矩阵形式为

$$\nabla \ell(\beta) = X'(y-\mu), \quad -d^2 \ell(\beta) = X'WX$$

，其中 $W = \operatorname{diag}\left(w_1, \ldots, w_n\right)$，$w_i = m_i \pi_i \left(1-\pi_i\right)$。

因此，

$$
\begin{aligned}
\beta^{(t+1)} &= \beta^{(t)} + \left[-d^2\ell\left(\beta^{(t)}\right)\right]^{-1} \nabla\ell\left(\beta^{(t)}\right) \\
&= \beta^{(t)} + \left(X'W^{(t)}X\right)^{-1} X'\left(y-\mu^{(t)}\right) \\
&= \left(X'W^{(t)}X\right)^{-1} X'W^{(t)}\left[X\beta^{(t)} + \left(W^{(t)}\right)^{-1}\left(y-\mu^{(t)}\right)\right] \\
&= \left(X'W^{(t)}X\right)^{-1} X'W^{(t)} z^{(t)}
\end{aligned}
$$

，其中

$$
z^{(t)} = X\beta^{(t)} + \left(W^{(t)}\right)^{-1}\left(y-\mu^{(t)}\right)
$$

$$
W^{(t)} = \operatorname{diag}\left(w^{(t)}\right), w^{(t)} = m\pi^{(t)}\left(1-\pi^{(t)}\right)
$$

$$
\mu^{(t)} = m\pi^{(t)}, \pi^{(t)} = \frac{\exp\left(X\beta^{(t)}\right)}{1+\exp\left(X\beta^{(t)}\right)}
$$

在此处，牛顿迭代法等价于 Fisher's Scoring Method。 □

在本题中，所生成的数据为未分组数据，即 $m_i = 1, i = 1, 2, \ldots, n$。

在此次，我们考虑在带有常数项与不带常数项、不同数据规模（200，2000）情况下重复进行 100 次估计的估计结果，结果如下：[1]

表 1: 带有常数项与不带常数项、不同数据规模下，牛顿迭代法的估计结果

n	Bias($\hat{\beta}_0$) (sd.)	Bias($\hat{\beta}_1$) (sd.)	Bias($\hat{\beta}_2$) (sd.)
200		0.0331 (0.2270)	0.1141 (0.3414)
200	0.0281 (0.2168)	0.0385 (0.2287)	0.0887 (0.4173)
2000		-0.0050 (0.0672)	0.0019 (0.0780)
2000	0.0027 (0.0668)	-0.0042 (0.0671)	-0.0005 (0.0982)

因此，我们可以得到如下结论：

- 当 $\beta_0 = 0$ 时，估计时是否包括常数项对于估计结果影响不大；
- 当样本增大时，对于参数 β 的估计更加精确。

[1]本次模拟由 Matlab 实现，具体代码可见于 https://github.com/SignorinoY/infestor。

```
function [X, y] = faker(n, beta, seed)
%faker generate faker logistic regression sample
%    Y ~ Bern(logis(X * beta)), where X0=1, X1 ~ N(0,1), X2 ~ Bern(0.5)
% Args:
%      n: sample size
%      beta: logistic regression parameters
%      seed: random seed, default to 0
% Returns:
%      X: covariates, without intercept
%      y: binary reponse

if nargin < 3, seed = 0;
end
rng(seed);
X = [randn(n, 1), rand(n, 1), <, 0.5];
eta = [ones(n, 1), X] * beta;
pi = exp(eta) ./ (1. + exp(eta));
y = rand(n, 1) < pi;
end
```

Listing 1: 仿真数据生成函数 *faker.m*

```matlab
function beta = logisticRegression(X, y, beta0, intercept, MaxIter, epsilon)
% logisticRegression fits the logistic regression model with (X, y)
% Args:
%      X : covariates, do not contain a column of 1s as an intercept
%      y : binary reponse
%      beta0: initial parameters for logistic regression
%      intercept: whether to include the intercept term
%      MaxIter maximum number of iterations, the default is 50
%      epsilon maximum termination tolerance error, the default is 1e-8
% Returns:
%      beta: estimated parameters for logistic regression

if nargin < 6, epsilon = 1e-8;
end
if nargin < 5, MaxIter = 50;
end
if nargin < 4, intercept = true;
end
if intercept
    [n, ~] = size(X);
    X = [ones(n, 1), X];
end
beta = beta0;
m = 1;
for iter = 1:MaxIter
    beta0 = beta;
    eta = X * beta0;
    pi = exp(eta) ./ (1. + exp(eta));
    mu = m * pi;
    W = diag(m * pi .* (1 - pi));
    z = eta + W^(-1) * (y - mu);
    beta = (X' * W * X)^(-1) * X' * W * z;
    if (norm(beta - beta0) < epsilon)
        break
    end
end
end
```

Listing 2: 牛顿迭代法求解 Logistic 回归模型函数 *logisticRegression.m*

```matlab
clc, clear

% Parameters
n = 200;
repeat = 100;

%% With Intercept
% Parameters
beta = [0, 1, 1.5]';
beta0 = [0; 0; 0];
% Simulation
bias = zeros(repeat, 3);
for seed = 1:repeat
    [X, y] = faker(n, beta, seed);
    betaHat = logisticRegression(X, y, beta0, true);
    bias(seed, :) = betaHat - beta;
end
% Estimated Result
mean(bias)
std(bias)

%% Without Intercept
% Parameters
beta = [0, 1, 1.5]';
beta0 = [0; 0];
% Simulation
bias = zeros(repeat, 2);
for seed = 1:repeat
    [X, y] = faker(n, beta, seed);
    betaHat = logisticRegression(X, y, beta0, false);
    bias(seed, :) = betaHat - beta(2:3);
end
% Estimated Result
mean(bias)
std(bias)
```

Listing 3: 在考虑常数项与不考虑常数项的情况下，重复进行 100 次模拟 *test.m*