南方科技大学

贝叶斯统计期末报告

作者：刘润祺
学号：11711331
课程：贝叶斯统计
指导老师：蔡敬衡
日期：2020 年 8 月 15 日

背景

使用 logistic 回归模型分析数据，即$y_i \sim Bernoulli(\pi_i)$，$logit(\pi_i) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} \doteq \boldsymbol{x_i^T \beta}$，其中$\boldsymbol{x_i} = (1, x_{i1}, x_{i2})^T$和$\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2)^T$。现有数据$(x_{i1}, x_{i2}, y_i), i = 1, \dots, n$，并记$\boldsymbol{y} = \{y_1, \dots, y_n\}$。假设$\beta$的先验分布为无信息先验，即$p(\boldsymbol{\beta}) \propto c, c$为常数。

1.$\boldsymbol{\beta}$的后验分布$\boldsymbol{p(\beta|y)}$

已知 $y_i \sim Bernoulli(\pi_i), f(y_i) = \pi_i^{y_i}(1-\pi_i)^{1-y_i}$

$$logit(\pi_i) = \boldsymbol{log}\left(\frac{\pi_i}{\boldsymbol{1} - \pi_i}\right) = \boldsymbol{x_i^T \beta}$$

$$\pi_i = \frac{\boldsymbol{exp(x_i^T \beta)}}{\boldsymbol{1 + exp(x_i^T \beta)}}$$

$$\boldsymbol{f(y|\beta)} = \prod_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{f}(y_i) = \prod_{\boldsymbol{i=1}}^{\boldsymbol{n}} \frac{(\boldsymbol{exp(x_i^T \beta)})^{y_i}}{\boldsymbol{1 + exp(x_i^T \beta)}}$$

根据贝叶斯公式，得到$\boldsymbol{\beta}$的后验分布$\boldsymbol{p(\beta|y)}$

$$\boldsymbol{p(\beta|y)} \propto f(y|\boldsymbol{\beta})p(\boldsymbol{\beta}) \propto \prod_{\boldsymbol{i=1}}^{\boldsymbol{n}} \frac{(\boldsymbol{exp(x_i^T \beta)})^{y_i}}{\boldsymbol{1 + exp(x_i^T \beta)}}$$

2. 用 Metropolis-Hastings 算法从$\boldsymbol{p(\beta|y)}$中抽取样本

算法步骤：

1. 从$p(\boldsymbol{\beta})$中抽取β^0作为起始点
2. 对于 t=1,2，⋯：

a. 把高斯分布作为跳跃分布(Jumping kernel)，令 $J_t(\beta^*|\beta^{t-1}) = N(\beta^*|\beta^{t-1}, 0.5^2 I)$，从$J_t(\beta^*|\beta^{t-1})$中抽取$\beta^*$。

b. 计算密度比$r = \frac{\boldsymbol{p(\beta^*|y)}}{\boldsymbol{p(\beta^{t-1}|y)}}$

c. 令$\beta^t = \begin{cases} \beta^*, & \text{概率为} \min(r, 1) \\ \beta^{t-1}, & \text{其他情况} \end{cases}$

步骤 c 的实现：从均匀分布 uniform(1)中抽取 u，若 u 小于$\min(r, 1)$，令$\beta^t = \beta^*$；否则令$\beta^t = \beta^{t-1}$。

3.用 r 语言实现 MH 算法

代码见附件。

4.用 r 语言实现模型 DIC 的计算

代码见附件。

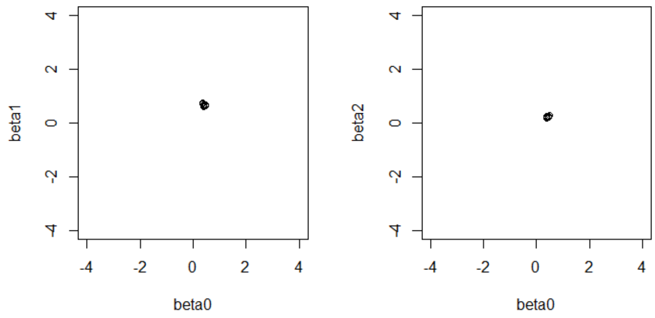
5. 分析数据集 Bayes_test.txt 并计算模型 DIC

数据集 Bayes_test.txt 有 1000 个样本，数据前两列为x_1和x_2，第三列为 y，取值为 0 或 1。

	V1	V2	V3
1	-1.2978850000	2.21663700	1
2	0.4988127000	1.35216300	1
3	1.6538780000	-1.23499700	0
4	-0.0891374500	1.34822400	1
5	-0.4125455000	0.30747860	0
6	0.5123374000	-2.54112300	1
7	-0.2628806000	-4.54253500	0
8	-1.3660010000	-3.34073300	1
9	0.2925877000	-2.91097000	0
10	-1.4565620000	-1.17496100	0

部分数据预览

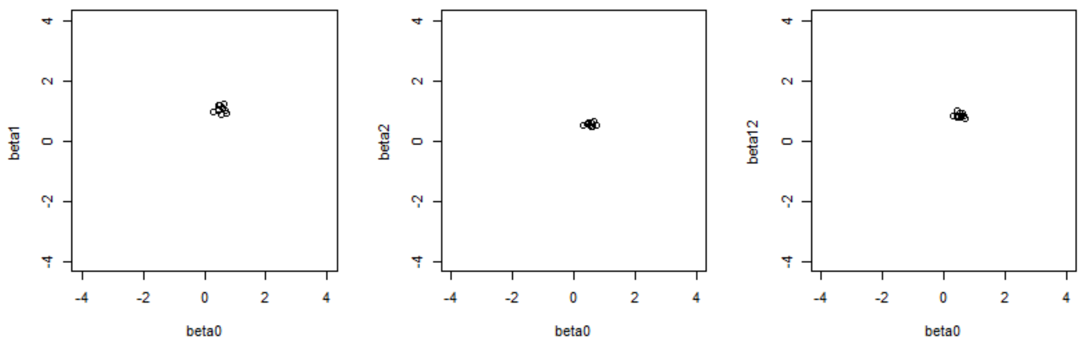
利用 3 中的 r 程序，用 MH 算法分析该数据集。以 M0:$logit(\pi_i) = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2}$为模型，选取（1，0，0），（1，1，0.1），（0.5，0.5，0.5）和（-0.1，-0.1，0）四个点作为起始点。选取高斯分布作为跳跃分布（Jumping kernel），即$J_t(\beta^*|\beta^{t-1}) = N(\beta^*|\beta^{t-1}, 0.5^2 I)$，迭代 800 次。做出四个点后 200 次迭代得到的模拟点的图像，观察算法收敛情况。可以看到这些模拟点都收敛到了一个小范围内，说明迭代 600 次之后，算法收敛性良好。



用四个起始点的后 200 个模拟点的均值作为$\boldsymbol{\beta}$的估计，得到β_0,β_1,β_2的估计值分别为 0.39，0.61 和 0.23。用 4 中的 r 程序计算 DIC，算出该模型 DIC 为 1243.29。

6.用带交互项的模型分析数据集 Bayes_test.txt

选用新模型 M1：$logit(\pi_i)=\beta_0+\beta_1x_{i1}+\beta_2x_{i2}+\beta_{12}x_{i1}x_{i2}$，用 3，4 给出的程序分析数据集 Bayes_test.txt 并计算 DIC。选取（1，1，1，1)，(1，1，0.1，0.1)，(0.5，0.5，0.5，0.5）和（-0.1，-0.1，0，1）四点作为起始点，用与 5 中相同的跳跃函数，迭代 800 次。下图为后 200 次迭代得到的模拟点的散点图，可以看出，在该模型下，算法的收敛性依旧很好。



用四个起始点的后 200 个模拟点的均值作为$\boldsymbol{\beta}$的估计，得到$\beta_0,\beta_1,\beta_2,\beta_{12}$的估计值分别为 0.50，1.08，0.53 和 0.85。计算出模型 M1 的 DIC 为 1045.30，小于 M0 的 DIC，因此 M1 模型更优。

7. 用 glm 方法分析数据集

调用 r 语言中的函数 glm()，分别用 M0 和 M1 模型分析数据集 Bayes_test.txt。下图为用两个模型得到的结果。可以看到，对于 M0 模型，用

glm 方法得到的β_0,β_1,β_2的估计值分别为 0.39，0.61 和 0.23，与 5 中用 MH 算法得到的结果基本相同；对于 M1 模型，用 glm 方法得到的$\beta_0,\beta_1,\beta_2,\beta_{12}$的估计值分别为 0.51，1.01，0.52 和 0.84，与 6 中用 MH 算法得到的结果基本相同。此外，比较用 glm 算法得到的两个模型的 AIC，可见 M1 的 AIC 较小，说明 M1 模型更优，这个结论和我用 MH 算法得到的结论相符。可见，虽然 MH 算法通过迭代更新的方法得到参数估计值，glm 方法通过最小二乘法得到估计值，两个算法思路不同，但都给出了相近的结果。

```
> #model0
> summary(glm(data$V3~data$V1+data$V2,family="binomial"))

Call:
glm(formula = data$V3 ~ data$V1 + data$V2, family = "binomial")

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7606  -1.1581   0.6405   0.9598   2.1056

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.39011    0.06908   5.647 1.63e-08 ***
data$V1      0.60988    0.07232   8.433  < 2e-16 ***
data$V2      0.23238    0.04224   5.502 3.76e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1361.2  on 999  degrees of freedom
Residual deviance: 1238.5  on 997  degrees of freedom
AIC: 1244.5

Number of Fisher Scoring iterations: 4
```

M0 模型结果

```
> #model1
> summary(glm(data$V3~data$V1+data$V2+data$V1*data$V2,family="binomial"))

Call:
glm(formula = data$V3 ~ data$V1 + data$V2 + data$V1 * data$V2,
    family = "binomial")

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4450  -0.9883   0.2364   0.9369   2.2438

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)      0.51466    0.08255   6.234 4.53e-10 ***
data$V1          1.00918    0.10106   9.986  < 2e-16 ***
data$V2          0.52199    0.06192   8.430  < 2e-16 ***
data$V1:data$V2  0.83799    0.07490  11.188  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1361.2  on 999  degrees of freedom
Residual deviance: 1033.3  on 996  degrees of freedom
AIC: 1041.3

Number of Fisher Scoring iterations: 6
```

M1 模型结果

为了探究两种算法分析数据集 Bayes_test.txt 的效率，我计下了它们运行的时间。对于模型 M0，选取 4 个起始点迭代 800 次，用我写的 MH 算法的运行时间为 40.66 秒，计算 DIC 用时 5.00 秒；而用 glm()函数得到$\boldsymbol{\beta}$的估计并计算 AIC 共用时不到 0.03 秒，可见用 glm()函数的运行效率显著高于我写的程序的运行效率。