统计学与 R 语言

第 21 讲　Logistic 回归

张敬信

2022 年 5 月 9 日

哈尔滨商业大学

一. 广义线性模型

- 线性回归，要求因变量是服从正态分布的连续型数据。但实际中，因变量数据可能会是类别型、计数型等。
- 要让线性回归也适用于因变量非正态连续情形，就需要推广到广义线性模型。
- Logistic 回归、softmax 回归、泊松回归、Probit 回归、二项回归、负二项回归、最大熵模型等都是广义线性模型的特例。

- 广义线性模型，相当于是复合函数。先做线性回归，再接一个变换：

$$X\beta = u \sim \text{正态分布}$$

$$\downarrow$$

$$g(u) = y$$

- 经过变换后到达非正态分布的因变量数据。

- 一般更习惯反过来写：即对因变量 y 做一个变换，就是正态分布，从而就可以做线性回归：

$$\sigma(y) = X\beta$$

- $\sigma(\cdot)$ 称为连接函数。

常见的连接函数和误差函数

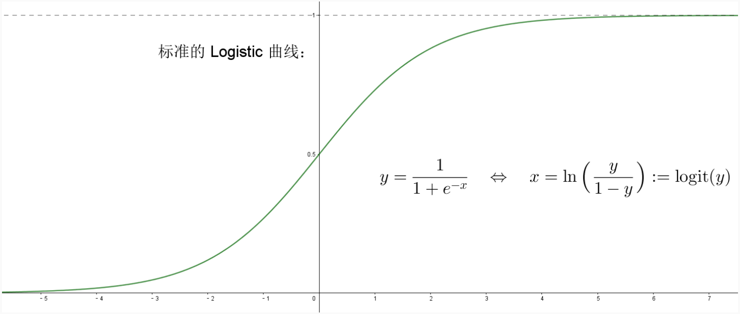
回归模型	变换	连接函数	逆连接函数	误差
线性回归	恒等	$\mu_Y = x^T\beta$	$\mu_Y = x^T\beta$	正态分布
Logistic 回归	Logit	$\mathrm{Logit}\,\mu_Y = x^T\beta$	$\mu_Y = \dfrac{\exp(x^T\beta)}{1+\exp(x^T\beta)}$	二项分布
泊松回归	对数	$\ln\mu_Y = x^T\beta$	$\mu_Y = \exp\left(x^T\beta\right)$	泊松分布
负二项回归	对数	$\ln\mu_Y = x^T\beta$	$\mu_Y = \exp\left(x^T\beta\right)$	负二项分布
Gamma 回归	逆	$\dfrac{1}{\mu_Y} = x^T\beta$	$\mu_Y = \dfrac{1}{x^T\beta}$	Gamma 分布

- 注：因变量数据只要服从指数族分布：正态分布、伯努利分布、泊松分布、指数分布、Gamma 分布、卡方分布、Beta 分布、狄里克雷分布、Categorical 分布、Wishart 分布、逆 Wishart 分布等，就可以使用对应的广义线性模型。

- Logistic 回归是分类模型，适合因变量是分类数据（例如：患病与不患病；违约与不违约）。
- 对于二分类因变量，$y=1$ 表示事件发生；$y=0$ 表示事件不发生。事件发生的条件概率 $\Pr(y=1|X)$ 与 x_i 之间是非线性关系，通常是单调的，即随着 X 的增加/减少，$\Pr(y=1|X)$ 也增加/减少。

1. Logistic 回归原理

- Logistic 回归可看作先做线性回归，再接一个逆连接函数：sigmoid 函数



- sigmoid 函数值域在 $(0,1)$ 之间，而且是一个渐变过程，正好适合描述概率 $\Pr(y=1|X)$.
- 概率值 $\Pr(y=1|X)$ 有了，再根据阈值 0.5 做判断：大于 0.5, 则预测 $\hat{y}=1$; 小于 0.5, 则预测 $\hat{y}=0$.

- 于是，二项 Logistic 回归模型可表示为

$$\operatorname{logit}(p) := \ln\left(\frac{p}{1-p}\right) = \ln\left(\frac{\Pr(y=1|X)}{\Pr(y=0|X)}\right) = X\beta$$

2. Logistic **回归系数的解释**

- 例如，影响是否患病的因素有性别和肿瘤体积，通过 Logistic 回归建模，得到

$$\text{Odds} = \frac{p}{1-p} = e^{\beta_0 + \beta_1 \text{Gender} + \beta_2 \text{Volume}} = e^{\beta_0} \cdot e^{\beta_1 \text{Gender}} \cdot e^{\beta_2 \text{Volume}}$$

- 对于分类变量 Gender，男用 1 表示，女用 0 表示，代入可得性别变量的发生比率为：

$$\frac{\text{Odds}_1}{\text{Odds}_0} = e^{\beta_1}$$

这表示男性患病的发生比约为女性患病发生比的 e^{β_1} 倍。

- 对于连续变量 Volume，若肿瘤体积从 v_0 增加 1 个单位，则

$$\frac{\text{Odds}_{v_0+1}}{\text{Odds}_{v0}} = e^{\beta_2}$$

这表示在其它变量不变的情况下，肿瘤体积每增加 1 个单位，将会使患病发生比变化 e^{β_2} 倍，注意该倍数是相对于原来 v_0 而言的。

3. Logisic 回归的损失函数

- 二分类 Logistic 回归用的是交叉熵损失：

$$J(\beta) = -\frac{1}{n}\sum_{i=1}^{n}\left[y^{(i)}\ln\hat{p}_i + (1-y^{(i)})\ln(1-\hat{p}_i)\right]$$

其中，y_i 为样本真实类别，$\hat{p}_i = \Pr(y=1|X)$ 为预测为正例的概率。

```
y_true = [   0,         0,         1,          1]
y_pred = [[.9, .1], [.8, .2], [.3, .7], [.01, .99]]
```

```
y = c(0, 0, 1, 1)                # 真实类别
p = c(0.1, 0.2, 0.7, 0.99)    # 预测为正例的概率
- mean(y * log(p) + (1-y) * log(1-p))    # 交叉熵损失
#> [1] 0.174
```

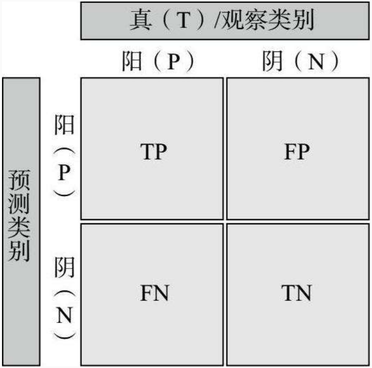
4. 阈值调参

- 得到预测概率后，通常是以 0.5 为阈值，将 y 预测为正类或负类，这只适合于均衡分类与代价不敏感的问题。
- 对于不均衡分类或代价敏感的问题，以 0.5 为阈值往往不是最优选择，需要根据具体需要，选择最优的阈值。

5. 二分类模型度量指标

- 准确率（Accuracy）: $\frac{n_{correct}}{n_{total}}$
- 混淆矩阵（Confusion Matrix）
- 精确率（Precision）
- 召回率（Recall）
- ROC 曲线
- AUC 值

混淆矩阵（Confusion Matrix）



- TP（True Positive，真正）：将正类预测为正类数
- TN（True Negative，真负）：将负类预测为负类数
- FP（False Positive，假正）：将负类预测为正类数误报（Type I error）
- FN（False Negative，假负）：将正类预测为负类数–> 漏报 (Type II error)

- **查准率（**Precision**）：**表示被分为正例的示例中实际为正例的比例

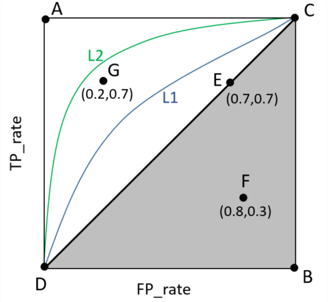
$$Perecision = \frac{TP}{TP + FP}$$

- **召回率 (**Recall**)：**是覆盖面的度量，度量有多少个正例被正确地分为正例

$$Recall = \frac{TP}{TP + FN}$$

ROC 曲线和 AUC 值

- ROC 曲线在不同分类阈值上对比真正率（TP_rate）与假正率（FP_rate）的曲线，ROC 曲线下面的面积叫做 AUC：



$$TP_{rate} = \frac{TP}{TP + FN}, \qquad FP_{rate} = \frac{FP}{FP + TN}$$

- ROC 曲线越接近左上角（AUC 面积越大），表示分类性能越好；
- ROC 曲线的优点是，对类分布/不平衡数据不敏感；正例负例的占比变化，ROC 曲线不变。

注：对于极度不平衡数据，建议用 PR（Precision-Recall）曲线。

研究 GRE（研究生入学成绩）、GPA（平均成绩）和本科院校的等级如何影响研究生院的录取。响应变量 admit0（不录取）/1（录取），是一个二值变量。

```
library(tidyverse)
df = read_csv("datas/binary.csv") %>%
  mutate(rank = factor(rank))
df
#> # A tibble: 400 x 4
#>   admit   gre   gpa rank
#>   <dbl> <dbl> <dbl> <fct>
#> 1     0   380  3.61 3
#> 2     1   660  3.67 3
#> 3     1   800  4    1
#> 4     1   640  3.19 4
#> # ... with 396 more rows
```

- 拟合 Logistic 回归模型

```
log_reg = glm(admit ~ gre + gpa + rank, data = df,
              family = "binomial")
library(broom)
glance(log_reg)
#> # A tibble: 1 x 8
#>   null.deviance df.null logLik   AIC   BIC deviance df.res
#>           <dbl>   <int>  <dbl> <dbl> <dbl>    <dbl>
#> 1          500.     399  -229.  471.  494.     459.
```

```
tidy(log_reg)
#> # A tibble: 6 x 5
#>   term        estimate std.error statistic  p.value
#>   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
#> 1 (Intercept) -3.99      1.14        -3.50 0.000465
#> 2 gre          0.00226   0.00109      2.07 0.0385
#> 3 gpa          0.804     0.332        2.42 0.0154
#> 4 rank2       -0.675     0.316       -2.13 0.0328
#> 5 rank3       -1.34      0.345       -3.88 0.000104
#> 6 rank4       -1.55      0.418       -3.71 0.000205
```

- 写出回归方程：

$$
\begin{aligned}
\ln\frac{p}{1-p} = &-3.99 + 0.00226 * gre + 0.804 * gpa \\
&- 0.675 * rank2 - 1.34 * rank3 - 1.55 * rank4
\end{aligned}
$$

其中，$p = \mathrm{Pr}(\text{admit} = 0|X)$.

自行练习：解释回归系数。

- 指数化回归系数和置信区间，让结果更好解释：

```
exp(cbind(OR = coef(log_reg), confint(log_reg)))
#>                 OR   2.5 % 97.5 %
#> (Intercept) 0.0185 0.00189  0.167
#> gre         1.0023 1.00014  1.004
#> gpa         2.2345 1.17386  4.324
#> rank2       0.5089 0.27229  0.945
#> rank3       0.2618 0.13164  0.512
#> rank4       0.2119 0.09072  0.471
```

自行练习：继续解释回归系数。

- Wald 检验变量 `rank` 的整体效应是否显著:

```
library(lmtest)
waldtest(log_reg, . ~ . - rank)  # 对比不带 rank 的模型
#> Wald test
#>
#> Model 1: admit ~ gre + gpa + rank
#> Model 2: admit ~ gre + gpa
#>   Res.Df Df    F  Pr(>F)
#> 1    394
#> 2    397 -3 6.97 0.00014 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 '
```

- 模型预测新数据

```
newdat = tibble(gre = mean(df$gre), gpa = mean(df$gpa),
                rank = factor(1:4))
rslt = bind_cols(newdat,
          predict(log_reg, newdata = newdat,
                  type = "response", se = TRUE)) %>%
  mutate(LL = plogis(fit - (1.96 * se.fit)),
         UL = plogis(fit + (1.96 * se.fit)))
```

```
rslt
#> # A tibble: 4 x 8
#>     gre   gpa rank    fit se.fit residual.scale    LL    U
#>   <dbl> <dbl> <fct> <dbl>  <dbl>          <dbl> <dbl> <dbl
#> 1  588.  3.39 1     0.517 0.0663              1 0.595 0.65
#> 2  588.  3.39 2     0.352 0.0398              1 0.568 0.60
#> 3  588.  3.39 3     0.219 0.0383              1 0.536 0.57
#> 4  588.  3.39 4     0.185 0.0486              1 0.522 0.57
```

1. **多项** Logistic **回归**

- 对于 y 是多分类情形，是 K 类可能的结果，任选一类，比如第 K 类，结果作为” 正例”，将其分别与其它 $K-1$ 类（作为” 负例”），做 $K-1$ 次二分类 logistic 回归：

$$\ln \frac{\Pr(y=1)}{\Pr(y=K)} = X\beta^{(1)}$$

$$\ln \frac{\Pr(y=2)}{\Pr(y=K)} = X\beta^{(2)}$$

$$\cdots\cdots$$

$$\ln \frac{\Pr(y=K-1)}{\Pr(y=K)} = X\beta^{(K-1)}$$

- 两边取 $\exp(\cdot)$, 再保证概率之和为 1, 则有

$$\Pr(y=K) = 1 - \sum_{k=1}^{K-1} \Pr(y=k) = 1 - \sum_{k=1}^{K-1} \Pr(y=K) e^{X\beta^{(k)}}$$

$$\Longrightarrow \qquad \Pr(y=K) = \frac{1}{1+\sum_{k=1}^{K-1} e^{X\beta^{(k)}}}$$

- 进而就能计算出其它概率，从而得到最终多项 Logistic 回归模型：

$$\begin{cases} \Pr(y=1) = \frac{e^{X\beta^{(1)}}}{1+\sum_{k=1}^{K-1} e^{X\beta^{(k)}}} \\ \qquad\qquad \cdots\cdots \\ \Pr(y=K-1) = \frac{e^{X\beta^{(K-1)}}}{1+\sum_{k=1}^{K-1} e^{X\beta^{(k)}}} \\ \Pr(y=K) = \frac{1}{1+\sum_{k=1}^{K-1} e^{X\beta^{(k)}}} \end{cases}$$

- 二分类 Logistic 回归自然推广到多分类就是 Softmax 回归，它是更均衡的做法。
- 将前面多项 Logistic 回归推导公式改写：

$$\ln \Pr(y=1) = X\beta^{(1)} - \ln z$$

$$\ln \Pr(y=2) = X\beta^{(2)}X - \ln z$$

$$\cdots\cdots$$

$$\ln \Pr(y=K) = X\beta^{(K)} - \ln z$$

- 同样需要保证：

$$\sum_{k=1}^{K} \Pr(y=k) = 1$$

- 这可以推出

$$z = \sum_{k=1}^{K} e^{X\beta^{(k)}}$$

- 进而，可以得到 Softmax 回归模型：

$$\Pr(y=k) = \frac{e^{X\beta^{(k)}}}{\sum_{k=1}^{K} e^{X\beta^{(k)}}}, \qquad k = 1, \cdots, K$$

- 于是，Softmax 回归即线性回归再接一个 softmax 函数：

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}}, \qquad i = 1, \cdots, K$$

- softmax 函数广泛应用于深度学习，是从连续值到多分类值的激活函数，深度学习中的分类任务，最后都需要接一个这样是 softmax 层。

- softmax 回归的损失函数是对数似然损失：

$$J(\beta) = -\frac{1}{n}\sum_{i=1}^{n} y_i \ln \hat{y}_i$$

其中，y_i 为第 i 个样本虚拟变量表示的真实类别，$\hat{y}_i$ 为第 i 个样本预测属于每个类别的概率向量。

注 1. 二分类 Logistic 回归的交叉熵损失，如果用虚拟变量表示，就是多分类对数似然损失的特例。

注 2. 二项 Logistic 回归与 Softmax 回归的最优参数，都是通过对损失函数应用梯度下降法计算的，选用 sigmoid 函数与 softmax 函数的另一优势就是它们的导数形式特别简洁易算。

当因变量是有序分类变量时，适合用有序 Logistic 回归。

不同于普通 Logistic 回归考虑单个事件的概率，有序 Logistic 回归考虑的是累积到事件 k 的概率（γ）：

$$\Pr(Y \leq k) = p_1 + \cdots + p_k, \qquad k = 1, \cdots, K$$

连接函数	形式	适用情形
Logit/Probit	$\ln(\frac{\gamma}{1-\gamma})$ 或 $\Phi^{-1}(\gamma)$	类别均匀分布/分析显性正态分布潜变量
余 log-log	$\ln(-\ln(1-\gamma))$	高类别概率更大
负 log-log	$-\ln(-\ln(\gamma))$	低类别概率更大
逆柯西	$\tan(\pi(\gamma-0.5))$	结果有较多极端值

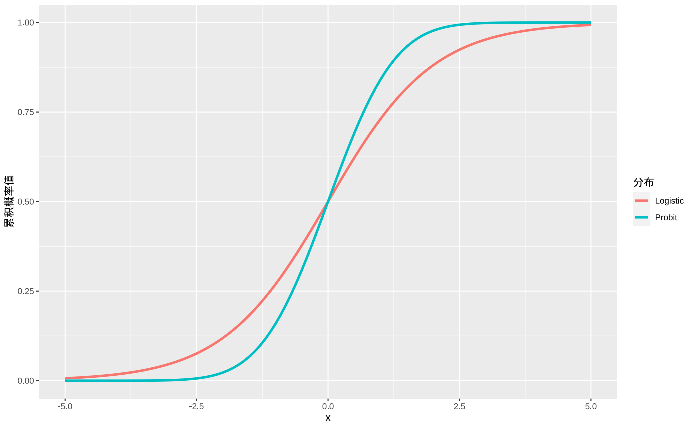
4. Probit 回归

Probit 回归的连接函数是累积正态分布的逆，即

$$\Phi^{-1}(y) = X\beta$$

Probit 回归与 Logistic 回归非常相似，区别是二者对误差分布的假设不同。

```
tibble(x = seq(-5, 5, 0.1), Probit = pnorm(x),
       Logistic = plogis(x)) %>%
 pivot_longer(-x, names_to = "distr", values_to = "val") %>%
 ggplot(aes(x, val, color = distr)) +
 geom_line(size = 1.2) +
 labs(y = " 累积概率值", color = " 分布")
```



理论上，这两种回归基本可以互换使用，但是 Probit 回归系数没有 Logistic 回归这样明确的可解释性，所以 Logistic 回归更加流行。

本篇主要参阅 (张敬信, 2022), Companion to BER 642: Advanced Regression Methods，模板感谢 (黄湘云, 2021), (谢益辉, 2021).

参考文献

张敬信 (2022). *R* 语言编程：基于 *tidyverse*. 人民邮电出版社, 北京.

谢益辉 (2021). *rmarkdown: Dynamic Documents for R*.

黄湘云 (2021). *Github: R-Markdown-Template*.