**12**

**Phân tích hồi qui logistic**

Trong các chương trước về phân tích hồi qui tuyến tính và phân tích phương sai, chúng ta tìm mô hình và mối liên hệ giữa một biến phụ thuộc liên tục (continuous dependent variable) và một hay nhiều biến độc lập (independent variable) hoặc là liên tục hoặc là không liên tục. Nhưng trong nhiều trường hợp, biến phụ thuộc không phải là biến liên tục mà là biến mang tính đo lường nhị phân: có/không, mắc bệnh/không mắc bệnh, chết/sống, xảy ra/không xảy ra, v.v…, còn các biến độc lập có thể là liên tục hay không liên tục. Chúng ta cũng muốn tìm hiểu mối liên hệ giữa các biến độc lập và biến phụ thuộc.

**Ví dụ 1**. Trong một nghiên cứu do tác giả tiến hành để tìm hiểu mối liên hệ giữa nguy cơ gãy xương (fracture, viết tắt là fx) và mật độ xương cùng một số chỉ số sinh hóa khác, 139 bệnh nhân nam (hay nói đúng hơn là đối tượng nghiên cứu) tuổi từ 60 trở lên. Năm 1990, các số liệu sau đây được thu thập cho mỗi đối tượng: độ tuổi (age), tỉ trọng cơ thể (body mass index hay BMI), mật độ chất khoáng trong xương (bone mineral density hay BMD), chỉ số hủy xương ICTP, chỉ số tạo xương PINP. Các đối tượng nghiên cứu được theo dõi trong vòng 15 năm. Trong thời gian theo dõi, các bệnh nhân bị gãy xương hay không gãy xương được ghi nhận. Câu hỏi đặt ra ban đầu là có một mối liên hệ gì giữa BMD và nguy cơ gãy xương hay không. Số liệu của nghiên cứu này được trình bày trong phần cuối của chương này, và sẽ trình bày một phần dưới đây để bạn đọc nắm được vấn đề.

**Bảng 1. Một phần số liệu nghiên cứu về các yếu tố nguy cơ cho gãy xương**

id fx age bmi bmd ictp pinp

1 1 79 24.7252 0.818 9.170 37.383

2 1 89 25.9909 0.871 7.561 24.685

3 1 70 25.3934 1.358 5.347 40.620

4 1 88 23.2254 0.714 7.354 56.782

5 1 85 24.6097 0.748 6.760 58.358

6 0 68 25.0762 0.935 4.939 67.123

7 0 70 19.8839 1.040 4.321 26.399

8 0 69 25.0593 1.002 4.212 47.515

9 0 74 25.6544 0.987 5.605 26.132

10 0 79 19.9594 0.863 5.204 60.267

...

137 0 64 38.0762 1.086 5.043 32.835

138 1 80 23.3887 0.875 4.086 23.837

139 0 67 25.9455 0.983 4.328 71.334

Ở đây, vì biến phụ thuộc (gãy xương) không được đo lường theo tính liên tục (mà chỉ là *có* hay *không*), cho nên phương pháp phân tích hồi qui tuyến tính để phân tích mối liên hệ giữa biến phụ thuộc và biến độc lập. Một phương pháp phân tích được phát triển tương đối gần đây (vào thập niên 1970s) có tên là logistic regression analysis (hay phân tích hồi qui logistic) có thể áp dụng cho trường hợp trên.

Trong nghiên cứu này, sau 15 năm theo dõi, có 38 bệnh nhân bị gãy xương. Tính theo phần trăm, tỉ lệ gãy xương là 38 / 139 = 0.273 (hay 27.3%).

**12.1 Mô hình hồi qui logistic**

Cho một tần số biến cố *x* ghi nhận từ *n* đối tượng, chúng ta có thể tính xác suất của biến cố đó là:

*p* có thể xem là một chỉ số đo lường nguy cơ của một biến cố. Một cách thể hiện nguy cơ khác là *odds*, tạm dịch là *khả năng*. Khả năng của một biến cố được định nghĩa đơn giản bằng tỉ số xác suất biến cố xảy ra trên xác suất biến cố không xảy ra:

[1]

Hàm *logit* của *odds* được định nghĩa như sau:

[2]

Mối liên hệ giữa *p* và logit(*p*) là một mối liên hệ liên tục và theo dạng như sau:

|  |
| --- |
|  |
| **Biểu đồ 1**. Mối liên hệ giữa logit(*p*) và *p*, cho 1*<p<*0. |

Chú ý: biểu đồ trên được vẽ bằng các lệnh sau đây:

p <- seq(0, 1, length=100)

p <- p[2:(length(p)-1)]

logit <- function(t)

{

log(t / (1-t))

}

plot(logit(p) ~ p, type="l")

Cho một biến độc lập *x* (*x* có thể là liên tục hay không liên tục), mô hình hồi qui logistic phát biểu rằng:

[3]

Tương tự như mô hình hồi qui tuyến tính,  và  là hai thông số tuyến tính cần phải ước tính từ dữ liệu nghiên cứu. Nhưng ý nghĩa của thông số này, đặc biệt là thông số , rất khác với ý nghĩa mà ta đã quen với mô hình hồi qui tuyến tính. Để hiểu ý nghĩa của hai thông số này, chúng ta sẽ quay lại với ví dụ 1.

**Ví dụ 1 (tiếp theo)**. Vấn đề mà chúng ta muốn biết là mối liên hệ giữa mật độ xương bmd và nguy cơ gãy xương (fx). Để tiện cho việc minh họa, gọi bmd là *x,* vấn đề mà chúng ta cần biết có thể viết bằng ngôn ngữ mô hình như sau

[4]

Nói cách khác:

Mô hình hồi qui logistic vừa trình bày trên phát biểu rằng mối liên hệ giữa xác suất gãy xương (*p*) và mật độ xương bmd là một mối liên hệ theo hình chữ S. Mô hình trên còn cho thấy xác suất gãy xương *p* tùy thuộc vào giá trị của *x*. Do đó, mô hình trên có thể viết một cách chính xác hơn rằng *khả năng* gãy xương với điều kiện *x* là:

Khi *x = x*0, khả năng gãy xương là:

Khi *x = x*0 + 1 (tức tăng 1 đơn vị từ *x*0), khả năng gãy xương là:

Và, tỉ số của hai xác suất gãy xương:

[5]

Trong dịch tễ học, được gọi là *odds ratio*, tạm dịch là *tỉ số khả năng* hay *tỉ số odds*. Nói cách khác, hệ số  trong mô hình hồi qui logistic chính là tỉ số khả dĩ.

Phương pháp để ước tính thông số trong mô hình [3] khá phức tạp (dùng phương pháp maximum likelihood – tức phương pháp *Hợp lí cực đại*) và không nằm trong phạm vi của cuốn sách này, nên sẽ không trình bày ở đây (bạn đọc có thể tham khảo sách giáo khoa để biết thêm, nếu cần thiết). Tuy nhiên, có lẽ cần đề cập một cách ngắn gọn là phương pháp hợp lí cực đại cung cấp cho chúng ta một hệ phương trình như sau:

Trong đó, Trong đó, *yi* là biến phụ thuộc (gãy xương với giá trị 0 hay 1), và *xi* là biến độc lập (mật độ xương), và *n* là số mẫu. Để tìm ước sốvà , một trong những phép tính hay sử dụng là iterative weighted least square hay Newton-Raphson. R sử dụng phép tính Newton-Raphson để tìm hai ước số đó.

Sau khi đã có ước số và chúng ta có thể ước tính xác suất *p* cho bất cứ giá trị nào của *x* như sau (sau vài thao tác đại số):

Chú ý chúng ta dùng dấu mũ để chỉ số ước tính (predicted value), chứ không phải *p* là xác suất quan sát. Nếu mô hình mô tả dữ liệu tốt và đầy đủ, độ khác biệt giữa *p* và nhỏ; nếu mô hình không thích hợp hay không tốt, độ khác biệt đó có thể sẽ cao. Độ khác biệt giữa *p* và được gọi là *deviance*. Phương pháp tính deviance khá phức tạp, nhưng đó không phải là chủ đề ở đây, cho nên chúng ta chỉ nói qua khái niệm mà thôi. Khi chúng ta có nhiều mô hình để mô phỏng một hay nhiều mối liên hệ, deviance có thể được sử dụng để đánh giá sự thích hợp của một mô hình, hay để chọn một mô hình “tối ưu”.

**12.2 Phân tích hồi qui logistic bằng** R

**Ví dụ 1 (tiếp theo)**. Bây giờ, chúng ta quay lại với ví dụ 1, dùng số liệu trong Bảng 12.1 để ước tính hai thông số  và  bằng R. Trước hết chúng ta phải nhập toàn bộ số liệu vào một data frame, và cho một cái tên, chẳng hạn như fracture. Trong trường hợp của ví dụ này, dữ liệu được chứa trong directory c:\works\stats dưới tên fracture.txt, do đó, các lệnh sau đây cần thiết để nhập số liệu:

# báo cho R biết nơi chứa số liệu

> setwd("c:/works/stats")

# nhập số liệu và cho vào một data frame tên fracture

> fracture <- read.table("fracture.txt", header=TRUE, na.string=".")

# kiểm tra xem có bao nhiêu biến trong dữ liệu fracture

> names(fracture)

[1] "id" "fx" "age" "bmi" "bmd" "ictp" "pinp"

# Chọn những bệnh nhân có đầy đủ số liệu cho phân tích

fulldata <- na.omit(fracture)

attach(fulldata)

Hai biến mà chúng ta quan tâm trong ví dụ này là: fx (gãy xương) và bmd (mật độ xương). Chúng ta kiểm tra xem có bao nhiêu bệnh nhân gãy xương:

> table(fx)

fx

0 1

101 38

Kế đến, xem mật độ xương trong nhóm gãy xương và không gãy xương ra sao:

> tapply(bmd, fx, mean)

0 1

0.9444851 0.9016667

> boxplot(bmd ~ fx,

xlab="Fracture: 1=yes, 0=no)",

ylab="BMD")



Kết quả trên cho thấy, bmd trong nhóm bệnh nhân bị gãy xương thấp hơn so với nhóm không bị gãy xương (0.90 và 0.94). Và, kiểm định t sau đây cho thấy mức độ khác biệt này không có ý nghĩa thống kê (p = 0.15).

> t.test(bmd~fx)

Welch Two Sample t-test

data: bmd by fx

t = 1.4572, df = 53.952, p-value = 0.1508

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

-0.01609226 0.10172922

sample estimates:

mean in group 0 mean in group 1

0.9444851 0.9016667

Để ước tính thông số trong mô hình [4], hàm số glm (viết tắt từ *generalized linear model*) trong R có thể áp dụng, với “cú pháp” như sau:

> logistic <- glm(fx ~ bmd, family=”binomial”)

> summary(logistic)

Call:

glm(formula = fx ~ bmd, family = "binomial")

Deviance Residuals:

Min 1Q Median 3Q Max

-1.0287 -0.8242 -0.7020 1.3780 2.0709

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) 1.063 1.342 0.792 0.428

bmd -2.270 1.455 -1.560 0.119

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 157.81 on 136 degrees of freedom

Residual deviance: 155.27 on 135 degrees of freedom

AIC: 159.27

Number of Fisher Scoring iterations: 4

Chúng ta sẽ lần lượt xem qua các kết quả trên:

1. Trong lệnh logistic <- glm(fx~bmd, family=”binomial”) chúng ta yêu cầu R phân tích theo mô hình fx là một hàm số với bmd như mô hình [4]. Trong glm có nhiều luật phân phối, mà trong đó phân phối nhị phân (binomial) là một luật phân phối chuẩn cho hồi qui logistic. Do đó, family=”binomial” cần thiết cho R.
2. Deviance: phần thứ nhất của kết quả cho biết qua về deviance:

Deviance Residuals:

Min 1Q Median 3Q Max

-1.0287 -0.8242 -0.7020 1.3780 2.0709

Deviance như giải thích trên phản ánh độ khác biệt về giá trị quan sát và ước tính của logit(*p*)(hay giữa mô hình và dữ liệu, cũng tương tự như *mean square residual* trong phân tích hồi qui tuyến tính vậy). Đối với một mô hình đơn lẻ như ví dụ này thì giá trị của deviance không có ý nghĩa gì nhiều.

1. Phần kế tiếp cung cấp ước số của  (mà R đặt tên là intercept) và  (bmd) và sai số chuẩn (standard error).

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) 1.063 1.342 0.792 0.428

bmd -2.270 1.455 -1.560 0.119

Qua kết quả này, chúng ta có = 1.063 và  = -2.27. Ước số  là số âm cho thấy mối liên hệ giữa nguy cơ gãy xương và bmd là mối liên hệ nghịch đảo: xác suất gãy xương tăng khi giá trị của bmd giảm. Tuy nhiên, kiểm định z (tính bằng cách lấy ước số chia cho sai số chuẩn) cho chúng ta thấy ảnh hưởng của bmd không có ý nghĩa thống kê, vì trị số P=0.119.

Nhớ rằng tỉ số khả dĩ (odds ratio hay viết tắt là OR) chính là . Nói cách khác, khi bmd tăng 1 g/cm2 (đơn vị đo lường của bmd là g/cm2) thì tỉ số OR giảm 0.9067 hay 90.67%. Nhưng tăng 1 g/cm2 là mật độ rất cao trong xương và không thực tế. Cho nên một cách tính khác là tính trên độ lệch chuẩn (standard deviation) của bmd. Chúng ta sẽ tìm hiểu độ lệch chuẩn của bmd:

> sd(bmd)

[1] 0.1406543

Do đó, OR sẽ tính trên mỗi 0.14 g/cm2. Và OR cho mỗi độ lệch chuẩn, do đó, là:

*e*-2.27\*0.1406 = 0.7267

Tức là, khi bmd tăng một độ lệch chuẩn thì tỉ số khả dĩ gãy xương giảm khoảng 28%. Cũng có thể nói cách khác, là khi bmd *giảm* một độ lệch chuẩn thì tỉ số khả dĩ tăng *e*2.27\*0.1406 = 1.376 hay khoảng 38%.

Một cách khác để biết ảnh hưởng của bmd là ước tính xác suất gãy xương là qua phương trình:



Theo đó, khi bmd = 1.00, p = 0.23. Khi bmd = 0.86 (tức giảm 1 độ lệch chuẩn), p = 0.291. Tức là, nếu BMD giảm 1 độ lệch chuẩn thì xác suất gãy xương tăng 0.291/0.23 = 1.265 hay 26.5%.

1. Phần cuối của kết quả cung cấp deviance cho hai mô hình: mô hình không có biến độc lập (null deviance), và mô hình với biến độc lập, tức là bmd trong ví dụ (residual deviance).

Null deviance: 157.81 on 136 degrees of freedom

Residual deviance: 155.27 on 135 degrees of freedom

AIC: 159.27

Qua hai số này, chúng ta thấy bmd ảnh hưởng rất thấp đến việc tiên đoán gãy xương, chỉ làm giảm deviance từ 157.8 xuống còn 155.27, và mức độ giảm này không có ý nghĩa thống kê.

Ngoài ra, R còn cung cấp giá trị của AIC (Akaike Information Criterion) được tính từ deviance và bậc tự do. Chúng ta sẽ quay lại ý nghĩa của AIC trong phần sắp đến khi so sánh các mô hình.

**12.3 Ước tính xác suất bằng** R

Trong phân tích trên, chúng ta cho các kết quả vào đối tượng logistic. Trong đối tượng này có nhiều thông tin có ích, nhưng nếu muốn xem các thông tin này chúng ta phải dùng đến các lệnh như summary chẳng hạn. Trong phần này, chúng ta sẽ xem qua vài hàm trong R để lấy các thông tin liên quan đến việc tiên đoán xác suất.

* predict dùng để liệt kê các giá trị ước tính (predicted values) của mô hình cho từng bệnh nhân.

> predict(logistic)

1 2 3 4 5 6

2.37757 1.08569 -2.14111 1.49282 0.96537 -0.94125

7 8 9 10 11 12

-1.73368 -1.67564 -0.66528 -0.50704 -0.94185 -0.64874

...

Các số trên là log(*p* / (1 – *p*)), tức *log odds*, không có ý nghĩa thực tế bao nhiêu. Chúng ta muốn biết giá trị tiên đoán xác suất *p* tính từ phương trình . Để có giá trị này cho từng bệnh nhân, chúng ta cho thông số type=”response” vào hàm predict như sau:

> predict(logistic, type="response")

1 2 3 4 5 6 7

0.915 0.747 0.105 0.816 0.724 0.281 0.150

8 9 10 11 12 13 14

0.158 0.339 0.376 0.281 0.343 0.443 0.238

...

Trong kết quả trên (chỉ in một phần) ước tính xác suất gãy xương cho bệnh nhân 1 là 0.915, cho bệnh nhân 2 là 0.747, v.v…

* Chúng ta có thể xem xét các giá trị tiên đoán này với độ bmd bằng cách dùng hàm plot thông thường:

> plot(bmd, fitted(glm(fx ~ bmd, family="binomial")))

|  |
| --- |
|  |
| Xác suất tiên đoán gãy xương (trục tung) và độ bmd (trục hoành) qua mô hình hồi qui logistic. |

Biểu đồ trên có thể cải tiến bằng cách cho các khoảng cách giá trị bmd gần nhau hơn (như 0.50, 0.55, 0.60, …, 1.20 chẳng hạn), và dùng đường biểu diễn thay vì dùng dấu chấm. Các lệnh sau đây sẽ cải tiến biểu đồ.

> logistic <- glm(fx ~ bmd, family="binomial")

#cho fnbmd từ > 0.50,0.55,0.6,...,1.2

> fnbmd <- seq(0.5, 1.2, 0.05)

#cho vào một dataframe mới

> new.data <- data.frame(bmd = fnbmd)

> predicted <- predict(logistic, new.data, type="response")

> plot(predicted ~ fnbmd, type="l")

|  |
| --- |
|  |
| Xác suất tiên đoán gãy xương (trục tung) và độ bmd (trục hoành) qua mô hình hồi qui logistic. |

**12.4 Phân tích hồi qui logistic từ số liệu giản lược bằng** R

Trong quá trình phân tích số liệu vừa trình bày trên đây, chúng ta có số liệu cho từng bệnh nhân và các biến độc lập đều là biến liên tục. Nhưng trong nhiều trường hợp biến độc lập là bậc thứ (và bởi vì biến phụ thuộc chỉ có hai giá trị 0 và 1) cho nên trên lí thuyết chúng ta có thể tóm lược dữ liệu bằng các bảng tần số (frequency table).

**Ví dụ 2**. Trong một nghiên cứu về ảnh hưởng của thói quen hút thuốc lá, tình trạng béo phì, thở ngáy (trong khi ngủ) đến nguy cơ bệnh cao huyết áp, các nhà nghiên cứu tóm lược số liệu như sau (số liệu trích từ Altman, trang 353):

|  |
| --- |
| smoking obesity snoring ntotal nhyper  0 0 0 60 5  1 0 0 17 2  0 1 0 8 1  1 1 0 2 0  0 0 1 187 35  1 0 1 85 13  0 1 1 51 15  1 1 1 23 8  Tổng số 433 79 |
| **Bảng 2**. Tóm lược số liệu liên quan đến hút thuốc lá (smoking), béo phì (obesity), ngáy (snoring), và cao huyết áp. ntotal là tổng số bệnh nhân cho từng nhóm, và nhyper là số bệnh nhân trong tổng số bị bệnh cao huyết áp. Các biến số smoking, obesity, và snoring có giá trị 0=no và 1=yes. |

Trong nghiên cứu có 433 bệnh nhân, và trong số này 79 người (hay 18%) bị bệnh cao huyết áp. Tuy nhiên, tỉ lệ này dao động khá cao theo từng nhóm bệnh nhân. Chẳng hạn như trong nhóm không hút thuốc lá(smoking=0), không béo phì (obesity=0) và không ngáy (snoring=0), tỉ lệ cao huyết áp là 8.3% (5/60). Trong khi đó nhóm với 3 yếu tố nguy cơ trên (smoking=1, obesity=1, snoring=0) thì có hơn 1 phần 3 hay 35% (8/23) bị bệnh cao huyết áp.

Để phân tích mối liên hệ giữa 3 yếu tố nguy cơ đó và bệnh cao huyết áp, trước hết cần phải cho số liệu vào R theo đúng như bảng số liệu trên.

> noyes <- c(“no”, “yes”) #định nghĩa biến noyes có 2 giá trị

> smoking <- gl(2,1,8, noyes) #biến smoking

> obesity <- gl(2,2,8, noyes) #biến obesity

> snoring <- gl(2,4,8, noyes) #biến snoring

> ntotal <- c(60, 17, 8, 2, 187, 85, 51, 23)

> nhyper <- c(5, 2, 1, 0, 35, 13, 15, 8)

> data <- data.frame(smoking, obesity, snoring, ntotal, nhyper)

> data

smoking obesity snoring ntotal nhyper

1 no no no 60 5

2 yes no no 17 2

3 no yes no 8 1

4 yes yes no 2 0

5 no no yes 187 35

6 yes no yes 85 13

7 no yes yes 51 15

8 yes yes yes 23 8

Bây giờ chúng ta có thể sử dụng hàm glm để phân tích số liệu. Trước hết, chúng ta phải tạo thêm một biến số proportion như sau:

> proportion <- nhyper/ntotal

> logistic <- glm(proportion ~ smoking+obesity+snoring,

family=”binomial”,

weight=ntotal)

Chú ý trong hàm glm trên, chúng ta mô phỏng proportion như là một hàm số của smoking, obesity và snoring, vẫn với phân phối nhị phân (binomial), nhưng có thêm một thông số weight=ntotal. Thông số weight yêu cầu R sử dụng ntotal là một số tóm lược (thay vì một bệnh nhân). Bây giờ, chúng ta có thể xem qua kết quả phân tích:

> summary(logistic)

Call:

glm(formula = proportion ~ smoking + obesity + snoring, family = "binomial", weights = ntotal)

Deviance Residuals:

1 2 3 4 5 6 7 8

-0.04344 0.54145 -0.25476 -0.80051 0.19759 -0.46602 -0.21262 0.56231

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -2.37766 0.38018 -6.254 4e-10 \*\*\*

smokingyes -0.06777 0.27812 -0.244 0.8075

obesityyes 0.69531 0.28509 2.439 0.0147 \*

snoringyes 0.87194 0.39757 2.193 0.0283 \*

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 14.1259 on 7 degrees of freedom

Residual deviance: 1.6184 on 4 degrees of freedom

AIC: 34.537

Number of Fisher Scoring iterations: 4

Kết quả trên cho thấy biến smoking không có ý nghĩa thống kê; cho nên có lẽ chúng ta nên bỏ biến này ra ngoài mô hình và có một mô hình đơn giản hơn:

> logistic <- glm(proportion ~ obesity+snoring,

family=”binomial”, weight=ntotal)

> summary(logistic)

Call:

glm(formula = proportion ~ obesity + snoring, family = "binomial", weights = ntotal)

Deviance Residuals:

1 2 3 4 5 6 7

-0.01247 0.47756 -0.24050 -0.82050 0.30794 -0.62742 -0.14449

8

0.45770

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -2.3921 0.3757 -6.366 1.94e-10 \*\*\*

obesityyes 0.6954 0.2851 2.440 0.0147 \*

snoringyes 0.8655 0.3967 2.182 0.0291 \*

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 14.1259 on 7 degrees of freedom

Residual deviance: 1.6781 on 5 degrees of freedom

AIC: 32.597

Number of Fisher Scoring iterations: 4

Phân tích phương sai trên deviance sau đây cũng khẳng định obesity và snoring là hai biến có ảnh hưởng đến cao huyết áp:

> anova(logistic, test="Chisq")

Analysis of Deviance Table

Model: binomial, link: logit

Response: proportion

Terms added sequentially (first to last)

Df Deviance Resid. Df Resid. Dev P(>|Chi|)

NULL 7 14.1259

obesity 1 6.8260 6 7.2999 0.0090

snoring 1 5.6218 5 1.6781 0.0177

**12.5 Phân tích hồi qui logistic đa biến và chọn mô hình**

Một trong những vấn đề khó khăn và có khi khá nan giải trong việc phân tích hồi qui logistic đa biến là chọn một mô hình để có thể mô tả đầy đủ dữ liệu. Một nghiên cứu với một biến phụ thuộc *y* và 3 biến độc lập *x*1, *x*2 và *x*3, chúng ta có thể có những mô hình sau đây để tiên đoán *y*: *y* = *f*(*x*1), *y* = *f*(*x*2), *y* = *f*(*x*3), *y* = *f*(*x*1, *x*2), *y* = *f*(*x*1, *x*3), *y* = *f*(*x*2, *x*3), và *y* = *f*(*x*1, *x*2, *x*3), trong đó *f* là hàm số. Nói chung với *k* biến độc lập *x*1, *x*2, *x*3, …, *x*k, chúng ta có rất nhiều mô hình (2k) để tiên đoán *y.* Trong điều kiện có nhiều mô hình khả dĩ như thế, vấn đề đặt ra là mô hình nào được xem là tối ưu?

Câu hỏi trên đặt ra một câu hỏi cơ bản khác: thế nào là “tối ưu”? Nói một cách ngắn gọn một mô hình tối ưu phải đáp ứng ba tiêu chuẩn sau đây:

* Đơn giản
* Đầy đủ
* Có ý nghĩa thực tế

Tiêu chuẩn đơn giản đòi hỏi mô hình có ít biến số độc lập, vì nếu quá nhiều biến số thì vấn đề diễn dịch sẽ trở nên khó khăn, và có khi thiếu thực tế. Nói một cách ví von, nếu chúng ta bỏ ra 50.000 đồng để mua 500 trang sách tốt hơn là bỏ ra 60.000 ngàn để mua cùng số trang sách. Tương tự, một mô hình với 3 biến độc lập mà có khả năng mô tả dữ liệu tương đương với mô hình với 5 biến độc lập, thì mô hình đầu được chọn. Một mô hình đơn giản là một mô hình … tiết kiệm! (Tiếng Anh gọi là *parsimonious model*).

Tiêu chuẩn đầy đủ ở đây có nghĩa là mô hình đó phải mô tả dữ liệu một cách thỏa đáng, tức phải tiên đoán gần (hay càng gần càng tốt) với giá trị thực tế quan sát của biến phụ thuộc *y*. Nếu giá trị quan sát của *y* là 10, và nếu có một mô hình tiên đoán là 9 và một mô hình tiên đoán là 6 thì mô hình đầu phải được xem là đầy đủ hơn.

Tiêu chuẩn “có ý nghĩa thực tế”, như cách gọi, có nghĩa là mô hình đó phải được yểm trợ bằng lí thuyết hay có ý nghĩa sinh học (nếu là nghiên cứu sinh học), ý nghĩa lâm sàng (nếu là nghiên cứu lâm sàng), v.v... Có thể số điện thoại một cách nào đó có liên quan đến tỉ lệ gãy xương, nhưng tất nhiên một mô hình như thế hoàn toàn vô nghĩa. Đây là một tiêu chuẩn quan trọng, bởi vì nếu một phân tích thống kê dẫn đến một mô hình dù rất có ý nghĩa toán học mà không có ý nghĩa thực tế thì mô hình đó cũng chỉ là một trò chơi con số, không có giá trị khoa học thật sự.

Tiêu chuẩn thứ ba (có ý nghĩa thực tế) thuộc về lĩnh vực lí thuyết, và chúng ta sẽ không bàn ở đây. Chúng ta sẽ bàn qua tiêu chuẩn đơn giản và đầy đủ. Một thước đo quan trọng và có ích để chúng ta quyết định một mô hình đơn giản và đầy đủ là Akaike Information Criterion (AIC) mà chúng ta đã gặp trong phần đầu của chương này. Để hiểu AIC, chúng ta quay lại với ví dụ 1.

Xin nhắc lại trong ví dụ 1, chúng ta muốn tiên đoán gãy xương (biến fx) từ các biến độc lập sau đây: độ tuổi (age), tỉ số cơ thể (bmi), mật độ chất khoáng trong xương (bmd), và hai chỉ số hủy xương (ictp) và tạo xương (pinp).

(a) Chúng ta thử mô hình fx là hàm số của độ tuổi:

> attach(fulldata)

> summary(glm(fx ~ age, family=”binomial”, data=fulldata))

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -8.06447 2.72559 -2.959 0.00309 \*\*

age 0.09806 0.03766 2.604 0.00922 \*\*

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 157.81 on 136 degrees of freedom

Residual deviance: 150.74 on 135 degrees of freedom

AIC: 154.74

Chúng ta để ý thấy residual deviance = 150.74, và AIC = 154.74. Thật ra, AIC được ước tính từ công thức:

AIC = Residual Deviance + 2(số thông số)

Trong mô hình trên, chúng ta có 2 thông số (intercept và age), cho nên AIC = 150.74 + 4 = 154.74.

(b) Mô hình thứ hai mà chúng ta muốn so sánh là fx là hàm số của ictp:

> summary(glm(fx ~ ictp, family=”binomial”, data=fulldata))

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -3.9206 0.7726 -5.074 3.89e-07 \*\*\*

ictp 0.6066 0.1527 3.973 7.11e-05 \*\*\*

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.'

0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 157.81 on 136 degrees of freedom

Residual deviance: 139.15 on 135 degrees of freedom

AIC: 143.15

Cũng với hai thông số, nhưng mô hình này có giá trị residual deviance (139.15) nhỏ hơn mô hình với độ tuổi (150.74), và do đó AIC cũng thấp hơn (143.15 so với 154.74). Kết quả này cho thấy mô hình với ictp mô tả fx đầy đủ hơn là mô hình với độ tuổi. So sánh này cho thấy trong hai mô hình này, chúng ta sẽ chọn mô hình với ictp.

(c) Bây giờ chúng ta thử xem mô hình với ictp và age.

> summary(glm(fx ~ ictp + age, family=”binomial”, data=fulldata))

(Intercept) -8.25707 2.91403 -2.834 0.004603 \*\*

ictp 0.55461 0.15665 3.540 0.000399 \*\*\*

age 0.06398 0.04067 1.573 0.115701

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 157.81 on 136 degrees of freedom

Residual deviance: 136.61 on 134 degrees of freedom

AIC: 142.61

Mô hình này với 3 thông số (intercept, age và ictp), nhưng trị số AIC chỉ giảm xuống 142.61 (so với mô hình với ictp là 143.15), một độ giảm rất khiêm tốn, trong khi chúng ta phải “tiêu” thêm một thông số! Chúng ta có thể kết luận rằng age không cần thiết trong mô hình này. Thật vậy, trị số p cho age là 0.115, tức không có ý nghĩa thống kê.

Qua ba trường hợp trên, chúng ta có thể rút ra một nhận xét chung: một mô hình đơn giản và đầy đủ phải là mô hình có trị số AIC càng thấp càng tốt *và* các biến độc lập phải có ý nghĩa thống kê. Thành ra, vấn đề đi tìm một mô hình đơn giản và đầy đủ là thật sự đi tìm một (hay nhiều) mô hình với trị số AIC thấp nhất hay gần thấp nhất.

Tất nhiên, chúng ta có thể xem xét nhiều mô hình khác bằng cách thay thế hay tổng hợp các biến số độc lập với nhau. Nhưng một việc làm như thế rất phức tạp, đòi hỏi nhiều thời gian và có khi rườm rà. R có một hàm gọi là step có thể giúp chúng ta đi tìm một mô hình đơn giản và đầy đủ. Trong ví dụ trên, cách sử dụng hàm step sẽ được viết như sau:

> temp <- glm(fx ~ ., family=”binomial”, data=fulldata)

Trong lệnh trên, thông số “fx ~ .” có nghĩa là tìm tất cả các biến độc lập (kí hiệu “.”) để tiên đoán fx trong dataframe fulldata. Kết quả cho vào đối tượng temp. Để xem kết quả trong temp, chúng ta lệnh search như sau:

> search <- step(temp)

> search <- step(temp)

Start: AIC= 146.09

fx ~ id + age + bmi + bmd + ictp + pinp

Df Deviance AIC

- pinp 1 132.45 144.45

- id 1 132.47 144.47

- age 1 132.63 144.63

- bmi 1 133.41 145.41

- bmd 1 133.87 145.87

<none> 132.09 146.09

- ictp 1 148.90 160.90

Step: AIC= 144.45

fx ~ id + age + bmi + bmd + ictp

Df Deviance AIC

- id 1 132.81 142.81

- age 1 133.14 143.14

- bmi 1 133.66 143.66

- bmd 1 134.00 144.00

<none> 132.45 144.45

- ictp 1 149.05 159.05

Step: AIC= 142.81

fx ~ age + bmi + bmd + ictp

Df Deviance AIC

- age 1 133.32 141.32

- bmi 1 133.67 141.67

- bmd 1 134.33 142.33

<none> 132.81 142.81

- ictp 1 149.88 157.88

Step: AIC= 141.33

fx ~ bmi + bmd + ictp

Df Deviance AIC

- bmi 1 134.34 140.34

<none> 133.32 141.32

- bmd 1 135.65 141.65

- ictp 1 155.18 161.18

Step: AIC= 140.34

fx ~ bmd + ictp

Df Deviance AIC

<none> 134.34 140.34

- bmd 1 139.15 143.15

- ictp 1 155.27 159.27

Trong kết quả trên, R báo cáo cho chúng ta biết từng bước trong quá trình đi tìm mộ mô hình tối ưu. Khởi đầu là mô hình với tất cả 6 biến, và trị số AIC = 146.09. Bước thứ hai chỉ gồm 5 biến (loại bỏ pinp) và AIC=144.45, v.v... Kết quả có thể tóm lược trong bảng sau đây:

|  |  |
| --- | --- |
| Mô hình | AIC |
| fx ~ id + age + bmi + bmd + ictp + pinp | 146.09 |
| fx ~ id + age + bmi + bmd + ictp | 144.45 |
| fx ~ age + bmi + bmd + ictp | 142.81 |
| fx ~ bmi + bmd + ictp | 141.33 |
| fx ~ bmd + ictp | 140.34 |

Kết quả 5 bước tìm mô hình, R dừng lại với mô hình gồm 2 biến bmd và ictp vì có giá trị AIC thấp nhất. Thật ra, nếu không muốn in tất cả các bước đi tìm mô hình, chúng ta chỉ cần lệnh summary như sau:

> summary(search)

Call:

glm(formula = fx ~ bmd + ictp, family = "binomial", data = fulldata)

Deviance Residuals:

Min 1Q Median 3Q Max

-1.9126 -0.7317 -0.5559 0.4212 2.1242

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -1.0651 1.5029 -0.709 0.4785

bmd -3.4998 1.6638 -2.103 0.0354 \*

ictp 0.6876 0.1704 4.036 5.43e-05 \*\*\*

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 157.81 on 136 degrees of freedom

Residual deviance: 134.34 on 134 degrees of freedom

AIC: 140.34

Number of Fisher Scoring iterations: 4

Kết quả này đơn giản hơn kết quả của hàm search, vì summary chỉ trình bày mô hình sau cùng. Nói tóm lại, trong phân tích này, chúng ta kết luận rằng bmd (mật độ chất khoáng trong xương) và ictp (marker về chu trình hủy xương) là hai yếu tố có liên hệ hay ảnh hưởng đến nguy cơ gãy xương.

**12.6 Chọn mô hình hồi qui logistic bằng Bayesian Model Average (BMA)**

Trong chương 10, chúng ta đã xem qua cách chọn và xây dựng một mô hình hồi qui tuyến tính bằng ứng dụng phép tính BMA. Chúng ta cũng có thể ứng dụng BMA vào việc xây dựng một mô hình hồi qui logistic.

Tiếp tục ví dụ 1, chúng ta sẽ chuẩn bị dữ liệu cho phân tích BMA bằng cách chọn ra biến phụ thuộc (trong trường hợp này là fx) và một ma trận gồm các biến độc lập. Tiếp theo đó, chúng ta sử dụng hàm bic.glm để tìm các biến có ảnh hưởng đến fx.

> attach(fulldata)

> names(fulldata)

[1] "id" "fx" "age" "bmi" "bmd" "ictp" "pinp"

# Chọn cột 3 đến 7 (từ age đến pinp) làm ma trận biến độc lập

> xvars <- fulldata[,3:7]

# Chọn fx làm biến phụ thuộc

> y <- fx

# Gọi hàm bic.glm với các thông số như sau:

> bma.search <- bic.glm(xvars, y, strict=F, OR=20,

glm.family="binomial")

# Tóm lược kết quả phân tích:

> summary(bma.search)

Call: bic.glm.data.frame(x = xvars, y = y, glm.family="binomial", strict=F,OR=20)

9 models were selected

Best 5 models (cumulative posterior probability = 0.8836 ):

**p!=0 EV SD model 1 model 2**

Intercept 100 -2.85 2.865 -3.920 -1.065

age 15.3 0.008 0.026 . .

bmi 21.7 -0.023 0.054 . .

bmd 39.7 -1.341 1.976 . -3.499

ictp 100.0 0.645 0.169 0.606 0.687

pinp 5.7 -0.0003 0.004 . .

nVar 1 2

BIC -525.04 -524.94

post prob 0.307 0.291

**model 3 model4 model 5**

Intercept -1.201 -8.257 -0.072

age . 0.063 .

bmi -0.116 . -0.070

bmd . . -2.696

ictp 0.680 0.554 0.714

pinp . . .

nVar 2 2 3

BIC -523.63 -522.67 -521.03

post prob 0.151 0.094 0.041

Kết quả phân tích trên đây cho thấy xác suất mà ictp là liên quan đến gãy xương là 100%, trong khi đó, xác suất cho bmd chỉ khoảng 40%. Nhưng quan trọng hơn, mô hình “tối ưu” nhất là mô hình với ictp, và xác suất cho mô hình này là 0.307. Mô hình tối ưu thứ hai gồm có ictp và bmd (cũng là mô hình dựa vào tiêu chuẩn AIC như mô tả phần trên), nhưng xác suất cho mô hình này tương đối thấp hơn (0.291). Ba mô hình khác cũng có thể là “ứng viên” để mô tả xác suất gãy xương đầy đủ. Rõ ràng, qua phân tích BMA, chúng ta có nhiều lựa chọn mô hình hơn, và ý thức được sự bất định của một mô hình thống kê.

> imageplot.bma(bma.search)



Biểu đồ trên thể hiện kết quả bằng số trong phần trước. Qua biểu đồ này chúng ta thấy ictp là yếu tố có ảnh hưởng đến nguy cơ gãy xương có tính nhất quán cao nhất (xuất hiện trong 100% mô hình). Yếu tố quan trọng thư hai có lẽ là bmd hay bmi. Các yếu tố như age và pinp tuy có khả năng ảnh hưởng đến nguy cơ gãy xương, nhưng các yếu tố này không có độ nhất quán cao như các yếu tố vừa kể trên.

Xây dựng mô hình thống kê là một nghệ thuật. Vì tính nghệ thuật của việc làm, nhà nghiên cứu phải cân nhắc rất nhiều yếu tố để đi đến một mô hình đẹp. Bởi vì mô hình là nhằm mục đích mô tả thực tế, một mô hình đẹp là mô hình mô tả sát với thực tế. Tuy nhiên nếu một mô hình phản ánh 100% thực tế thì đó không còn là “mô hình” nữa, hay quá phức tạp không thể ứng dụng được. Ngược lại một mô hình chỉ mô tả thực tế khoảng 1% thì cũng không thể sử dụng được. Xây dựng mô hình phải làm sao tìm điểm cân bằng cho hai thái cực đó. Đó là một yêu cầu rất cao, cho nên xây dựng mô hình không chỉ tùy thuộc vào các phép tính thống kê, toán học, mà còn phải xem xét đến các yếu tố thực tế để bảo đảm cho sự hữu ích của mô hình. Nói như nhà thống kê học George Box: “*Mô hình nào cũng sai so với thực tế, nhưng trong số các mô hình sai đó, có một vài mô hình có ích*”.

**Phụ chương: Mô hình hồi qui Poission**

Số liệu sau đây phản ảnh số ca mắc bệnh ung thư da ở phụ nữ thuộc “thành phố song sinh” Minneapolis – St Paul:

|  |  |  |  |
| --- | --- | --- | --- |
| **Độ tuổi** | **Dân số** | **Số trường hợp ung thư da** | **Tỉ lệ phát sinh trên 1000 dân số** |
| 15-24 | 1 | 172,675 | 0.0058 |
| 25-34 | 16 | 146,207 | 0.1094 |
| 35-44 | 30 | 121,374 | 0.2472 |
| 45-54 | 71 | 111,353 | 0.6376 |
| 55-64 | 102 | 83,004 | 1.2289 |
| 65-74 | 130 | 55,932 | 2.3243 |
| 75-84 | 133 | 29,007 | 4.5851 |
| 85+ | 40 | 7,538 | 5.3064 |

Dựa vào hai số liệu trên, chúng ta có thể ước tính tỉ lệ phát sinh (incidence rate) tính trên 1000 dân số, và tỉ lệ này được trình bày trong cột cuối cùng của bảng số liệu. Chúng ta dễ dàng thấy tỉ lệ phát sinh ung thư da tăng theo độ tuổi.

Chúng ta cần một mô hình để mô tả xu hướng trên, hay mối liên hệ giữa độ tuổi và tỉ lệ phát sinh. Ở đây, có một khó khăn là dân số khác nhau giữa các độ tuổi, mà tỉ lệ thì tùy thuộc vào độ tuổi. Cho nên chúng ta cần một mô hình có thể điều chỉnh cho hai yếu tố này. Một mô hình thích hợp cho trường hợp này là mô hình hồi qui Poisson.

Gọi *N*i là dân số của độ tuổi *i*th, và gọi  là số trường hợp ung thư da, chúng ta có thể xem tỉ số như là một tỉ lệ phát sinh bệnh. Bây giờ, để tiện cho việc tính toán, chúng ta hoán chuyển tỉ số này sang đơn vị logarithm:

[1]

Mô hình hồi qui Poisson phát biểu rằng log của tỉ lệ phát sinh, tức là, là một hàm số của độ tuổi (và chúng ta kí hiệu độ tuổi bằng *x*). Nói cách khác, mô hình này phát biểu rằng:

[2]

Thay thế [1] vào vế trái của [2], chúng ta có:

hay nói cách khác:

[3]

Mô hình [3] chính là mô hình hồi qui Poisson. Cách diễn đạt mô hình theo công thức [3] cho phép chúng ta xây dựng một hàm số khả dĩ (likelihood function) để ước tính các thông số  và . Trong mô hình trên, được gọi là *offset,* tức là “điểm nhấn”.

Chúng ta có thể sử dụng R để ước tính các thông số trong mô hình [3] như sau:

age = c(19.5,29.5,39.5,49.5,59.5,69.5,79.5, 89.4)

cases = c(1, 16, 30, 71, 102, 130, 133, 40)

pop = c(172675, 123065, 96216, 92051, 72159, 54722, 32185, 8328)

dataset = data.frame(age, cases, pop)

fit = glm(cases ~ age + offset(log(pop)), family=poisson,data=dataset)

summary(fit)

Kết quả của phân tích là:

|  |
| --- |
| Deviance Residuals:  Min 1Q Median 3Q Max  -4.87198 -1.67519 -0.07185 1.20816 1.99291  Coefficients:  Estimate Std. Error z value Pr(>|z|)  (Intercept) -10.551604 0.168780 -62.52 <2e-16 \*\*\*  age 0.063629 0.002475 25.71 <2e-16 \*\*\*  ---  Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  (Dispersion parameter for poisson family taken to be 1)  Null deviance: 845.789 on 7 degrees of freedom  Residual deviance: 44.288 on 6 degrees of freedom  AIC: 91.688  Number of Fisher Scoring iterations: 5 |

Chúng ta chỉ chú ý đến thông số intercept và age. Thay thế các thông số này vào mô hình [3] chúng ta có:

trong đó, *x* là độ tuổi.

Cũng có thể viết mô hình trên thành tỉ lệ phát sinh (thay vì log của tỉ lệ) như sau:

Nếu tuổi là 19.5 (*x* = 19.5), chúng ta có thể ước tính tỉ lệ phát sinh là:

Nếu tuổi là 29.5, tỉ lệ phát sinh là:

Tỉ số của hai tỉ lệ phát sinh này là: 0.000171 / 0.0000904 = 1.89. Nói cách khác, tỉ lệ phát sinh tăng gấp 1.89 lần cho mỗi 10 tuổi tăng. Con số 1.89 này còn có tên là *relative risk* hay *tỉ số nguy cơ tương đối.*

Thật ra, chúng ta có thể ước tính ngay từ ước số của R:

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -10.551604 0.168780 -62.52 <2e-16 \*\*\*

age 0.063629 0.002475 25.71 <2e-16 \*\*\*

Trong phần trên, 0.0636 có nghĩa là cứ mỗi một tuổi tăng thì tỉ số nguy cơ tương đối tăng *e*0.0636 = 1.065 (hay tăng 6.5%). Do đó, khi tăng 10 tuổi thì nguy cơ tương đối mắc bệnh tăng *e*0.0636 x 10 = 1.89, hay 89%.