Aplicaciones del modelo de regresión logística: cálculo de predicciones

Contents

1 Predicción de π_i

Sea el modelo de regresión logística

$$P(Y_i = y_i) = \pi_i^{y_i}(1-\pi_i)^{1-y_i}, \quad y_i = 0, 1, \quad i = 1, 2, \ldots, n$$

donde:

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

$$x_i = \begin{bmatrix} 1 \\ x_{1i} \\ x_{2i} \\ \cdots \\ x_{ki} \end{bmatrix}, \quad \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \cdots \\ \beta_k \end{bmatrix}$$

Estamos interesados en el valor de la respuesta para los regresores $x_p^T = [1\ x_{1p}\ x_{2p}\ \cdots\ x_{kp}]$. El valor predicho de π_i en x_p es:

$$\hat{\pi}_p = \frac{exp(x_p^T\hat{\beta})}{1 + exp(x_p^T\hat{\beta})}$$

donde $\hat{\beta}$ es el vector de parámetros estimados:

$$\hat{\beta} = \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \hat{\beta}_2 \\ \cdots \\ \hat{\beta}_k \end{bmatrix}$$

2 Intervalo de confianza para π_p

Se tiene que

$$\hat{\beta} \sim N(\beta, (X^T W X)^{-1})$$

Por tanto

$$x_p^T \hat{\beta} \sim N(x_p^T \beta, x_p^T (X^T W X)^{-1} x_p)$$

ya que

$$E[x_p^T \hat{\beta}] = x_p^T E[\hat{\beta}] = x_p^T \beta$$

y

$$Var[x_p^T \hat{\beta}] = x_p^T Var[\hat{\beta}] x_p = x_p^T (X^T W X)^{-1} x_p$$

Por tanto, el intervalo de confianza para $x_p^T \beta$ es

$$x_p^T \hat{\beta} - z_{\alpha/2} \sqrt{x_p^T (X^T W X)^{-1} x_p} \leq x_p^T \beta \leq x_p^T \hat{\beta} + z_{\alpha/2} \sqrt{x_p^T (X^T W X)^{-1} x_p}$$

Si llamamos:

$$L_p = x_p^T \hat{\beta} - z_{\alpha/2} \sqrt{x_p^T (X^T W X)^{-1} x_p} U_p = x_p^T \hat{\beta} + z_{\alpha/2} \sqrt{x_p^T (X^T W X)^{-1} x_p}$$

se tiene que

$$L_p \leq x_p^T \beta \leq U_p$$

$$exp(L_p) \leq exp(x_p^T \beta) \leq exp(U_p)$$

Sumando uno a cada término:

$$1 + exp(L_p) \leq 1 + exp(x_p^T \beta) \leq 1 + exp(U_p)$$

y dividiendo ambas expresiones término a término se obtiene:

$$\frac{exp(L_p)}{1 + exp(L_p)} \leq \pi_p \leq \frac{exp(U_p)}{1 + exp(U_p)}$$

donde se recuerda que

$$\pi_p = \frac{exp(x_p^T \beta)}{1 + exp(x_p^T \beta)}$$

3 Ejemplos

```r
d = read.csv("datos/MichelinNY.csv")
```

Primero estimamos el modelo:

```r
m1 = glm(InMichelin ~ Food + Decor + Service + Price, data = d, family = binomial)
summary(m1)
```

```
##
## Call:
## glm(formula = InMichelin ~ Food + Decor + Service + Price, family = binomial,
##     data = d)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -11.19745    2.30896  -4.850 1.24e-06 ***
## Food          0.40485    0.13146   3.080  0.00207 **
## Decor         0.09997    0.08919   1.121  0.26235
## Service      -0.19242    0.12357  -1.557  0.11942
## Price         0.09172    0.03175   2.889  0.00387 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 225.79  on 163  degrees of freedom
## Residual deviance: 148.40  on 159  degrees of freedom
## AIC: 158.4
##
## Number of Fisher Scoring iterations: 6
```

Queremos calcular la predicción en Food = 22, Decor = 25, Service = 24, Price = 75:

```r
xp = c(1,22,25,24,75)
beta_e = coef(m1)
( pi_p = exp(t(xp) %*% beta_e)/(1 + exp(t(xp) %*% beta_e)) )
```

```
##           [,1]
## [1,] 0.9219606
```

Para calcular el intervalo de confianza:

```r
source("funciones/logit_funciones.R")
H = logit_hess(coef(m1),model.matrix(m1))
xp = matrix(xp, ncol = 1)
(se = sqrt(- t(xp) %*% solve(H) %*% xp ))
```

```
##           [,1]
## [1,] 0.6718705
```

```r
alfa = 0.05
Lp = t(xp) %*% beta_e - qnorm(1-alfa/2)*se
Up = t(xp) %*% beta_e + qnorm(1-alfa/2)*se
# limite inferior intrevalo confianza
exp(Lp)/(1+exp(Lp))
```

```
##           [,1]
## [1,] 0.7599576
```

```r
# limite superior intrevalo confianza
exp(Up)/(1+exp(Up))
```

```
##           [,1]
## [1,] 0.9778199
```

Con R, podemos predecir las probabilidades $\hat{\pi}_p$:

```r
xp_df = data.frame(Food = 22, Decor = 25, Service = 24, Price = 75)
(pred = predict(m1, newdata = xp_df, type = "response"))
```

```
##         1
## 0.9219606
```

Para calcular el intervalo de confianza activamos la opción se.fit:

```r
(pred = predict(m1, newdata = xp_df, type = "response", se.fit = T))
```

```
## $fit
##         1
## 0.9219606
##
## $se.fit
##          1
## 0.04834054
##
## $residual.scale
## [1] 1
```

```r
alfa = 0.05
# limite inferior intervalo confianza
pred$fit - qnorm(1-alfa/2)*pred$se.fit
```

```
##         1
## 0.8272149
```

```r
# limite superior intervalo confianza
pred$fit + qnorm(1-alfa/2)*pred$se.fit
```

```
##        1
## 1.016706
```

Sin embargo, estos valores no coinciden con los calculados anteriormente. Es más, obtenemos un valor de probabilidad por encima de 1, lo que no es posible. Esto es debido a que las probabilidades no tienen distribución normal, y en el cálculo de estos intervalos estamos asumiento que las probabilidades estimadas tienen esa distribución.

Lo que hemos encontrado de manera teórica es que $x_p^T\hat{\beta}$ tiene distribución normal, no que π_p tiene distribución normal. El término $x_p^T\hat{\beta}$ se conoce como *link*. En R se puede predecir el *link* en lugar de probabilidades:

```r
(pred = predict(m1, newdata = xp_df, type = "link", se.fit = T))
```

```
## $fit
##        1
## 2.469289
##
## $se.fit
## [1] 0.6718702
##
```

```
## $residual.scale
## [1] 1
```

Por tanto, el intervalo de confianza sería:

```
alfa = 0.05
Lp = pred$fit - qnorm(1-alfa/2)*pred$se.fit
Up = pred$fit + qnorm(1-alfa/2)*pred$se.fit
# limite inferior intervalo confianza
exp(Lp)/(1+exp(Lp))
```

```
##         1
## 0.7599577
```

```
# limite superior intervalo confianza
exp(Up)/(1+exp(Up))
```

```
##         1
## 0.9778199
```

Hemos hecho la predicción de $\hat{\pi}_p$, probabilidades, pero queremos predecir si un Restaurante con Food = 22, Decor = 19, Service = 24, Price = 55 va a estar en la Guía Michelín o no. Para eso, adoptamos el criterio:

- si $\hat{P}(Y_p = 1) = \hat{\pi}_p > 0.5$, entonces consideramos que el restaurante está en la Guía Michelin.
- si $\hat{P}(Y_p = 1) = \hat{\pi}_p < 0.5$, entonces consideramos que el restaurante no está en la Guía Michelin.

En este caso, como $\hat{\pi}_p = 0.92$, la predicción es que ese restaurante va a estar incluido en la Guía Michelín. Además, el intervalo de confianza está muy por encima de 0.5, luego tenemos mucha confianza en esa decisión.

El límite de 0.5 que hemos utilizado es totalmente arbitrario y se puede modificar según el criterio del autor del análisis.

4 Intervalo de confianza para la predicción utilizando bootstrap

```
set.seed(567)
B = 500
n = nrow(d)
yp = rep(0,B)
for (b in 1:B){
  # remuestrear datos
  posb = sample(1:n, replace = T)
  db = d[posb,]

  # estimar modelo b
  mb = glm(InMichelin ~ Food + Decor + Service + Price,
           data = db,family = binomial)

  # se guarda la prediccion
  yp[b] = predict(mb, newdata = xp_df, type = "response")
}
# prediccion puntual
mean(yp)
```

```
## [1] 0.916302
```

```
# valor del standard error calculado con bootstrap (de la respuesta)
sd(yp)
```

```
## [1] 0.07256359
```

```
# intervalo de prediccion
quantile(yp, probs = c(alfa/2,1-alfa/2))
```

```
##      2.5%     97.5%
## 0.7451559 0.9972382
```