Aplicaciones del modelo de regresión logística: cálculo de predicciones

Contents

1 Predicción de π_i

Sea el modelo de regresión de Poisson

$$P(Y_i = y_i) = \frac{e^{-\lambda_i}\lambda_i^{y_i}}{y_i!}, \quad y_i = 0, 1, 2, 3, \ldots$$

donde:

$$\lambda_i = exp(x_i^T \beta)$$

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

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:

$$\hat{\lambda}_p = exp(x_p^T \hat{\beta})$$

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

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

Por tanto

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

ya que

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

y

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

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

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

Si llamamos:

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

se tiene que

$$exp(L_p) \leq \lambda_p \leq exp(U_p)$$

donde se recuerda que

$$\lambda_p = exp(x_p^T\beta)$$

3 Ejemplos

```
d = read.csv("datos/Aircraft_Damage.csv")
```

Primero estimamos el modelo:

```
m1 = glm(damage ~ bomber + load + experience, data = d, family = poisson)
summary(m1)
```

```
##
## Call:
## glm(formula = damage ~ bomber + load + experience, family = poisson,
##     data = d)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6418  -1.0064  -0.0180   0.5581   1.9094
```

```
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.406023   0.877489  -0.463   0.6436
## bomber       0.568772   0.504372   1.128   0.2595
## load         0.165425   0.067541   2.449   0.0143 *
## experience  -0.013522   0.008281  -1.633   0.1025
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 53.883  on 29  degrees of freedom
## Residual deviance: 25.953  on 26  degrees of freedom
## AIC: 87.649
##
## Number of Fisher Scoring iterations: 5
```

Queremos calcular la predicción en bomber = A-44 (0), load = 6, experience = 75:

```
xp = c(1,0,6,75)
beta_e = coef(m1)
( lambda_p = exp(t(xp) %*% beta_e) )
```

```
##           [,1]
## [1,] 0.6520434
```

Para calcular el intervalo de confianza:

```
source("poisson_funciones.R")
H = hess_logL(coef(m1),model.matrix(m1))
xp = matrix(xp, ncol = 1)
(se = sqrt(- t(xp) %*% solve(H) %*% xp ))
```

```
##           [,1]
## [1,] 0.3168692
```

```
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]
## [1,] 0.3503942
```

```
# limite superior intrevalo confianza
exp(Up)
```

```
##          [,1]
## [1,] 1.213378
```

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

```
xp_df = data.frame(bomber = 0, load = 6, experience = 75)
(pred = predict(m1, newdata = xp_df, type = "response"))
```

```
##         1
## 0.6520434
```

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

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

```
## $fit
##         1
## 0.6520434
##
## $se.fit
##         1
## 0.2066122
##
## $residual.scale
## [1] 1
```

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

```
##         1
## 0.2470909
```

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

```
##        1
## 1.056996
```

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. Ese término se conoce como *link*. En R se puede predecir el *link* en lugar de probabilidades:

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

```
## $fit
##          1
## -0.4276442
##
## $se.fit
## [1] 0.3168688
##
## $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
## 0.3503945
```

```r
# limite superior intervalo confianza
exp(Up)
```

```
##        1
## 1.213377
```