

Estimadores y su distribución. Inferencia

Contents

1 Distribución asintótica de los estimadores	1
2 Contrastes de hipótesis individuales	2
3 Intervalos de confianza	3
4 Bootstrap	3

1 Distribución asintótica de los estimadores

Para muestras grandes, los estimadores de máxima verosimilitud tienen distribución asintótica normal. En concreto, se tiene que

$$\hat{\beta} \sim N(\beta, I(\hat{\beta}))$$

donde $I(\beta)$ se denomina Matriz de Información de Fisher observada:

$$I(\beta) = -H_{logL}^{-1}(\beta)$$

es decir, la inversa del hessiano de la función de verosimilitud (con signo negativo):

$$H_{logL}(\beta) = -X^T W X$$

En el caso de la propiedad anterior, la matriz $I(\beta)$ está evaluada en el valor que maximiza la verosimilitud. Por tanto, cada estimador de manera individual se distribuye como:

$$\hat{\beta}_j \sim N(\beta_j, Var(\hat{\beta}_j))$$

donde

$$Var(\hat{\beta}_j) = I_{(j+1,j+1)}(\hat{\beta}), \quad j = 0, 1, \dots, k$$

es decir, los elementos de la diagonal de la matriz $I(\hat{\beta})$. Al igual que en regresión lineal, el *standard error* de los estimadores es:

$$se(\hat{\beta}_j) = \sqrt{Var(\hat{\beta}_j)}$$

2 Contrastes de hipótesis individuales

Para resolver contrastes del tipo:

$$H_0 : \beta_j = 0$$

$$H_1 : \beta_j \neq 0$$

se utiliza la distribución asintótica mostrada anteriormente. Por tanto, si la hipótesis nula es cierta se tiene:

$$\frac{\hat{\beta}_j}{se(\hat{\beta}_j)} \sim N(0, 1)$$

A este método se lo conoce como método de Wald, o estadístico de Wald.

Con R:

Estimamos los parámetros del modelo (se van a utilizar las funciones de R del archivo poisson_funciones.R):

```
# cargamos las funciones que vamos a utilizar
source("funciones/poisson_funciones.R")
d = read.csv("datos/Aircraft_Damage.csv")
d$bomber = factor(d$bomber, labels = c("A4", "A6"))

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

##
## Call:
## glm(formula = damage ~ bomber + load + experience, family = poisson,
##      data = d)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.406023  0.877489 -0.463   0.6436
## bomberA6     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

beta_e = coef(m)
# matriz de información de Fisher
I = -solve(poisson_hess(beta_e, model.matrix(m)))
# standard error de los parámetros estimados
(beta_se = sqrt(diag(I)))

## (Intercept)    bomberA6        load    experience
## 0.877489518  0.504372854  0.067541103  0.008280828
```

```

# valor del estadístico del contraste
(z = beta_e/beta_se)

## (Intercept)    bomberA6      load  experience
##   -0.4627094   1.1276825   2.4492552  -1.6329669
# pvalores
2*(1 - pnorm(abs(z)))

## (Intercept)    bomberA6      load  experience
##   0.6435726   0.2594540   0.0143152   0.1024760

```

3 Intervalos de confianza

Partimos del estadístico de Wald:

$$\frac{\hat{\beta}_j - \beta_j}{se(\hat{\beta}_j)} \sim N(0, 1)$$

Por tanto, el intervalo será:

$$\hat{\beta}_j - z_{\alpha/2} se(\hat{\beta}_j) \leq \beta_j \leq \hat{\beta}_j + z_{\alpha/2} se(\hat{\beta}_j)$$

```

alfa = 0.05
data.frame(LI = beta_e - qnorm(1-alfa/2)*beta_se,
LS = beta_e + qnorm(1-alfa/2)*beta_se)

##           LI          LS
## (Intercept) -2.12587054 1.313825160
## bomberA6    -0.41978021 1.557325049
## load        0.03304727 0.297803527
## experience -0.02975244 0.002707807
confint(m, level = 1-alfa)

## Waiting for profiling to be done...
##           2.5 %     97.5 %
## (Intercept) -2.19824432 1.253906807
## bomberA6    -0.42621497 1.567016666
## load        0.03526245 0.301488598
## experience -0.02998744 0.002670296

```

4 Bootstrap

```

set.seed(99)
B = 500
n = nrow(d)
beta_B = matrix(0, nrow = B, ncol = 4)
for (b in 1:B){
  pos_b = sample(1:n, n, replace = T)
  d_b = d[pos_b,]
  m_b = glm(damage ~ bomber + load + experience, data = d_b, family = poisson)

```

```
    beta_B[b,] = coef(m_b)
}
```

- Standard errors calculados con bootstrap:

```
apply(beta_B, 2, sd)
```

```
## [1] 1.26643083 0.60749379 0.09540763 0.01208960
```

- Intervalos de confianza calculados con bootstrap:

```
alfa = 0.05
```

```
apply(beta_B, 2, quantile, probs = c(alfa/2, 1 - alfa/2))
```

```
## [,1]      [,2]      [,3]      [,4]
## 2.5%   -3.496665 -1.061774 -0.01456464 -0.03164706
## 97.5%   1.376005  1.452667  0.40581495  0.01325222
```