Estimadores y su distribución. Inferencia

Contents

1 Inferencia de parámetros individuales

1.1 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

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

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:

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

1.2 Contrastes de hipótesis individuales

Para resolver contrastes del tipo:

$$H_0 : \beta_j = 0 H_1 : \beta_j \neq 0$$

se utiliza la distribución asintóntica 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:

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

```
## 'data.frame':    164 obs. of  6 variables:
##  $ InMichelin     : int  0 0 0 1 0 0 1 1 1 0 ...
##  $ Restaurant.Name: chr  "14 Wall Street" "212" "26 Seats" "44" ...
##  $ Food           : int  19 17 23 19 23 18 24 23 27 20 ...
##  $ Decor          : int  20 17 17 23 12 17 21 22 27 17 ...
##  $ Service        : int  19 16 21 16 19 17 22 21 27 19 ...
##  $ Price          : int  50 43 35 52 24 36 51 61 179 42 ...
```

```
#d$InMichelin = factor(d$InMichelin)
```

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

```
# cargamos las funciones que vamos a utilizar
source("funciones/logit_funciones.R")

# punto de partida
m = lm(InMichelin ~ Food + Decor + Service + Price, data = d)
beta_i = coef(m)

# matriz X
X = cbind(rep(1,nrow(d)), d[,3:6])

# estimacion con el algoritmo optim
mle = optim(par = beta_i, fn = logit_logL_optim, gr = NULL,
  y = d$InMichelin, X = X,
  method = "BFGS", hessian = F,
  control = list(trace = 1, REPORT = 1, maxit = 200))
```

```
## initial  value 108.793234
## iter   2 value 108.179208
## iter   3 value 95.583943
## iter   4 value 94.546080
## iter   5 value 93.925129
## iter   6 value 76.862843
## iter   7 value 74.881601
## iter   8 value 74.297605
## iter   9 value 74.231461
## iter  10 value 74.210848
```

```
## iter  11 value 74.202494
## iter  12 value 74.199408
## iter  13 value 74.199353
## iter  14 value 74.199254
## iter  15 value 74.199238
## iter  16 value 74.199235
## iter  17 value 74.198614
## iter  18 value 74.198479
## iter  19 value 74.198474
## iter  19 value 74.198474
## iter  19 value 74.198474
## final  value 74.198474
## converged
```

```r
(beta_e = mle$par)
```

```
##  (Intercept)         Food        Decor      Service        Price
## -11.19660070   0.40483799   0.09992470  -0.19249028   0.09175322
```

```r
# matriz de información de Fisher
I = -solve(logit_hess(beta_e,X))
# standard error de los parámetros estimados
(beta_se = sqrt(diag(I)))
```

```
## rep(1, nrow(d))            Food           Decor         Service           Price
##      2.30897047      0.13146216      0.08919429      0.12357290      0.03175475
```

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

```
## (Intercept)        Food       Decor     Service       Price
##   -4.849174    3.079502    1.120304   -1.557706    2.889433
```

```r
# pvalores
2*(1 - pnorm(abs(z)))
```

```
##  (Intercept)         Food        Decor      Service        Price
## 1.239763e-06 2.073469e-03 2.625843e-01 1.193029e-01 3.859378e-03
```

Esta información se obtiene cuando utilizamos la funcion glm():

```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
```

1.3 Intervalos de confianza

Partimos de nuevo 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) -15.72209967 -6.67110173
## Food          0.14717690  0.66249909
## Decor        -0.07489290  0.27474231
## Service      -0.43468871  0.04970816
## Price         0.02951504  0.15399139
```

```
confint(m1, level = 1-alfa)
```

```
## Waiting for profiling to be done...
```

```
##                   2.5 %      97.5 %
## (Intercept) -16.05404706 -6.93894624
## Food          0.14737740  0.66963794
## Decor        -0.07378966  0.27855161
## Service      -0.44107191  0.04758419
## Price         0.03153229  0.15709677
```