Mínimos cuadrados

Guillermo Ayala Gallego

Mínimos cuadrados

Guillermo Ayala Gallego

2024-03-25

Introducción

- Disponemos de los datos \((\mathbf x_i,y_i)\) con \(i=1, \ldots, n\) siendo \(\mathbf y = (y_1,\ldots,y_n)^T\), las respuestas observadas; \(\mathbf x_i = (x_{i1},\ldots,x_{ip})^T\), los predictores correspondientes a la \(i\)-ésima observación.
- La matriz de modelo que recoge los valores de los predictores: \[\mathbf X = \begin{bmatrix} x_{11} & \ldots & x_{1p} \\ \vdots & \vdots & \vdots \\ x_{n1} & \ldots & x_{np} \end{bmatrix} \]
- Asumimos \[\mathbf \mu = \mathbf X \mathbf \beta. \]
- ¿Cómo estimamos \(\mathbf \beta\)?

Planteamiento

Estimadores mínimo cuadráticos

- Una opción clásica son los mínimos cuadrados en donde pretendemos que \[\| \mathbf y - \hat{ \mathbf \mu} \|^2 \]
- Lo que estamos haciendo es sustituir el vector de medias desconocido por los valores observados.
- Consideramos la función \[L(\mathbf \beta) = \sum_{i=1}^n (y_i - \mu_i)^2 = \sum_{i=1}^n \bigg (y_i - \sum_{j=1}^p \beta_j x_{ij}\bigg)^2. \]
- Los estimadores mínimo cuadráticos minimizan la función \(L(\mathbf \beta)\). \[\hat{ \mathbf \beta} = (\mathbf X^T \mathbf X)^{-1} \mathbf X^T \mathbf y. \]

Predicciones y la matriz \(H\)

- Las medias estimadas las obtenemos con \[\hat{ \mathbf \mu} = \mathbf X (\mathbf X^T \mathbf X)^{-1} \mathbf X^T \mathbf y. \]
- Si denotamos \[\mathbf H = \mathbf X (\mathbf X^T \mathbf X)^{-1} \mathbf X^T, \] entonces \[\hat{ \mathbf \mu} = \mathbf H \mathbf y. \] \[E [\hat{ \mathbf \beta}] = \mathbf \beta. \] \[var [\hat{ \mathbf \beta}] = \sigma^2 (\mathbf X^T \mathbf X)^{-1}, \]

Residuos

- Tenemos \[C(\mathbf X)^{\perp} = null(\mathbf X^T). \]
- El espacio \(C(\mathbf X)^{\perp}\) básicamente se le puede llamar **espacio del error**.
  - Los residuos son \[\mathbf e = \mathbf y - \mathbf X \hat{ \mathbf \beta}. \]
- Se tiene \[\mathbf e \in null(\mathbf X^T) = C(\mathbf X)^{\perp}. \]

Estimando la variación

Modelo

- El modelo lineal que consideramos incorporando el error aleatorio es \[\mathbf Y = \mathbf X \mathbf \beta + \mathbf \epsilon, \] con \(var(\mathbf \epsilon) = \sigma^2 \mathbf I\).

Estimador insesgado de la varianza

- Se tiene
  \[E \bigg [\frac{ \mathbf Y^T (\mathbf I - \mathbf P_{ \mathbf X}) \mathbf Y}{n-p} \bigg] = E \sum_{i=1}^n \frac{ (Y_i - \hat{\mu}_i)^2}{n-p} = \sigma^2. \]

Estimador insesgado de la varianza\(\ldots\)

- En el caso en que la matriz modelo no tuviera rango completo \(rank(X) = r < p\) entonces la misma prueba nos valdría sustituyendo \(p\) por \(r\).
- El estimador seguiría siendo insesgado.
- El estimador \(S^2\) recibe el nombre de **cuadrado medio del error** o **cuadrado medio residual**.
- Su raiz cuadrada, \(S\) es el **error estándar residual**.
- \(S^2\) es un estimador insesgado de \(\sigma^2\) mientras que \(S\) no es un estimador insesgado de \(\sigma\).

Coeficiente de determinación

- Se define como \[\begin{multline} \label{eq:160219gg} R^2 = \frac{SS(Regression)}{SS(Total)} = \frac{SS(Total)-SS(Residual)}{SS(Total)} = \\ \frac{\sum_{i=1}^n (y_i -\bar{y})^2 - \sum_{i=1}^n (y_i -\hat{\mu}_i)^2}{\sum_{i=1}^n (y_i -\bar{y})^2}. \end{multline}\]
- El coeficiente de correlación muestral se define como \[corr(\mathbf y, \hat{ \mathbf \mu}) = \frac{\sum_{i=1}^n (y_i -\bar{y}) (\hat{\mu}_i - \bar{\hat{\mu}})}{ \sqrt{\sum_{i=1}^n (y_i -\bar{y})^2 \sum_{i=1}^n (\hat{\mu}_i - \bar{\hat{\mu}})^2}}. \]

- Se verifica \[corr(\mathbf y, \hat{ \mathbf \mu}) = +\sqrt{R^2}. \]
- La raiz cuadrada positiva de \(R^2\) recibe el nombre de **correlación múltiple**.
- \(R^2\) ajustada es \[R^2_{adjusted} = 1 - \frac{SS(Residual)/(n-p)}{SS(Total)/(n-1)} = 1 - \frac{n-1}{n-p} (1 - R^2). \]

Residuo estandarizado

- Se define como
\[r_i = \frac{y_i - \hat{\mu}_i}{s \sqrt{1- h_{ii}}}. \] siendo \(s = \sqrt{\sum_{i=1}^n \frac{e_i^2}{n-p}}\).
- El **residuo studentizado** se define como el residuo estandarizado lo que ocurre es que el ajuste se realiza sin considerar el propio punto.

tamidata2::gse25171

Con `time` y `Pi`

- Analizamos los datos correspondientes a la sonda `261892_at`.
- Consideramos las variables fenotípicas `time` y `Pi`.

```
pacman::p_load(Biobase)
data(gse25171,package="tamidata2")
head(pData(gse25171),n=2)
```

```
                 time time2        Pi replication
GSM618324.CEL.gz    0 Short Treatment           1
GSM618325.CEL.gz    0 Short   Control           2
```

```
sel0 = which("261892_at"==fData(gse25171)[,"PROBEID"])
df0 = data.frame(pData(gse25171)[,c("time","Pi")],
```

```
3                   expression=exprs(gse25171)[sel0,])
```

- Ajustamos un modelo con dos variables predictoras.

```
1 fit4 = lm(expression~ time + Pi,data=df0)
```

- La matriz modelo es

```
1 head(model.matrix(fit4),n=5)
```

```
                  (Intercept) time PiTreatment
GSM618324.CEL.gz            1    0           1
GSM618325.CEL.gz            1    0           0
GSM618326.CEL.gz            1    1           1
GSM618327.CEL.gz            1    1           0
GSM618328.CEL.gz            1    6           1
```

- Los coeficientes los tenemos con

```
1 coef(fit4)
```

```
(Intercept)        time PiTreatment
  8.6293898  -0.1330703  -1.3219071
```

- Los residuos los tenemos con

```
1 resid(fit4)
```

```
GSM618324.CEL.gz GSM618325.CEL.gz GSM618326.CEL.gz GSM618327.CEL.gz
      1.01552765       0.21676116      -0.32266386      -0.07784405
GSM618328.CEL.gz GSM618329.CEL.gz GSM618330.CEL.gz GSM618331.CEL.gz
     -0.59106601       0.31824015       0.22638066      -0.30489608
GSM618332.CEL.gz GSM618333.CEL.gz GSM618334.CEL.gz GSM618335.CEL.gz
      0.49588789      -0.15067778      -0.63263328       0.15231308
GSM618336.CEL.gz GSM618337.CEL.gz GSM618338.CEL.gz GSM618339.CEL.gz
     -0.93990720       0.45816343       0.21916840      -0.92275255
GSM618340.CEL.gz GSM618341.CEL.gz GSM618342.CEL.gz GSM618343.CEL.gz
      0.24175878       0.15101414      -0.46350659       0.52297219
GSM618344.CEL.gz GSM618345.CEL.gz GSM618346.CEL.gz GSM618347.CEL.gz
     -0.39355515      -0.42938559       1.14460870       0.06609189
```

- El error estándar residual lo tenemos con

```
1 summary(fit4)$sigma
```

```
[1] 0.5644856
```

Con `time2` y `Pi`

- Analizamos los datos correspondientes a la sonda `261892_at`.
- Consideramos las variables fenotípicas `time2` y `Pi`.

```r
pacman::p_load(Biobase)
data(gse25171,package="tamidata2")
head(pData(gse25171),n=2)
```

```
                time time2        Pi replication
GSM618324.CEL.gz    0 Short Treatment           1
GSM618325.CEL.gz    0 Short   Control           2
```

```r
sel0 = which("261892_at"==fData(gse25171)[,"PROBEID"])
df0 = data.frame(pData(gse25171)[,c("time2","Pi")],
                 expression=exprs(gse25171)[sel0,])
```

- Ajustamos un modelo con dos variables predictoras. Ambas son categóricas y consideramos un modelo de análisis de la varianza con dos vías.

```r
fit5 = aov(expression~ time2 + Pi,data=df0)
```

- La matriz modelo es

```r
head(model.matrix(fit5),n=5)
```

```
                (Intercept) time2Medium PiTreatment
GSM618324.CEL.gz           1           0           1
GSM618325.CEL.gz           1           0           0
GSM618326.CEL.gz           1           0           1
GSM618327.CEL.gz           1           0           0
GSM618328.CEL.gz           1           1           1
```

- La tabla de análisis de la varianza la tenemos con

```r
summary(aov(fit5))
```

```
            Df Sum Sq Mean Sq F value   Pr(>F)
time2        1  26.99  26.992   29.36 2.24e-05 ***
Pi           1  10.48  10.485   11.41  0.00285 **
Residuals   21  19.30   0.919
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- ¿Y un modelo con interacción?

```r
fit6 = aov(expression~ time2 * Pi,data=df0)
```

- Vemos la tabla anova.

```r
summary(fit6)
```

```
            Df Sum Sq Mean Sq F value   Pr(>F)
time2        1 26.992  26.992   28.02 3.51e-05 ***
Pi           1 10.485  10.485   10.88  0.00358 **
time2:Pi     1  0.038   0.038    0.04  0.84370
Residuals   20 19.264   0.963
---
```

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

- No parece que tengamos interacción.

Muchos modelos

Un modelo por fila con `GSEAlm`

- Hemos elegido trabajar con una sonda elegida de un modo arbitrario.
- Tenemos interés en todas las sondas.
- Hemos de ajustar un modelo lineal para cada una de estas sondas.
- Los predictores serán variables fenotípicas compartidas por todas las sondas.
- Los distintos modelos comparten los predictores y difieren en la variable respuesta.

```
pacman::p_load(GSEAlm)
data(gse25171,package="tamidata2")
fits1 = GSEAlm::lmPerGene(gse25171,~ time + Pi)
```

- La matriz de modelo común a todos los ajustes la tenemos con

```
head(fits1$x)
```

```
                 (Intercept) time PiTreatment
GSM618324.CEL.gz           1    0           1
GSM618325.CEL.gz           1    0           0
GSM618326.CEL.gz           1    1           1
GSM618327.CEL.gz           1    1           0
GSM618328.CEL.gz           1    6           1
GSM618329.CEL.gz           1    6           0
```

- Los coeficientes de cada ajuste con

```
fits1$coefficients
```

- Para la primera sonda tenemos los coeficientes

```
fits1$coefficients[,1]
```

```
(Intercept)        time PiTreatment
5.080830082 0.003844458 0.010492799
```

- Las varianzas del error estimadas son

```
fits1$sigmaSqr
```

Y los estadísticos correspondientes a los tests de coeficientes nulos con

```
fits1$tstat
```

Un modelo por fila con `limma`

```
pacman::p_load(limma)
```

- Ajustamos los modelos.

```
design = model.matrix(~ pData(gse25171)[,"time"] +  pData(gse25171)[,"Pi"])
fits2 = limma::lmFit(gse25171,design)
```

- Los coeficientes los tenemos con

```
head(fits2$coefficients,n=2)
```

```
          (Intercept) pData(gse25171)[, "time"]
244901_at    5.080830               0.003844458
244902_at    4.843889              -0.005471131
          pData(gse25171)[, "Pi"]Treatment
244901_at                     0.0104927991
244902_at                    -0.0009233809
```

- Los coeficientes del ajuste para la primera sonda son

```
fits2$coefficients[1,]
```

```
                    (Intercept)        pData(gse25171)[, "time"]
                    5.080830082                     0.003844458
pData(gse25171)[, "Pi"]Treatment
                    0.010492799
```