Modelos lineales generalizados

Guillermo Ayala Gallego

2025-05-21

Table of contents

Modelo lineal generalizado **1**
Componentes de un modelo lineal generalizado . . . 1
La componente aleatoria . . . 2
Distribución binomial . . . 3
Distribución Poisson . . . 3
La componente sistemática . . . 4
Función de enlace . . . 4
Estimación de los coeficientes . . . 4
Distribución de los estimadores máximo verosímiles . . . 5
Covarianzas de $\hat{}$ . . . 5
Comparando modelos anidados . . . 5

Regresión logística **6**
Datos . . . 6

Modelo loglineales de Poisson **10**
Datos . . . 10
Modelo loglineal de Poisson . . . 10
Sobredispersión . . . 12
GLM binomiales negativos . . . 12

Modelo lineal generalizado

Componentes de un modelo lineal generalizado

- **Componente aleatoria**
- **Componente sistemática**

- **Función de enlace**

La componente aleatoria

- Consiste de una variable aleatoria Y con observaciones independientes $(Y_1, ..., Y_n)$.
- La respuesta Y sigue una distribución en la **familia de dispersión exponencial**

$$f(y_i; \theta_i, \phi) = e^{\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i, \phi)}.$$

- El parámetro θ_i el **parámetro natural**.
- El parámetro ϕ es el **parámetro de dispersión**.
- Familia exponencial natural ss un caso particular de la **familia de dispersión exponencial**.
- Ocurre cuando

$$a(\phi) = 1,$$
$$c(y_i, \phi) = c(y_i).$$

- La densidad tiene la forma

$$f(y_i; \theta_i) = h(y_i) \exp\left[y_i\theta_i - b(\theta_i)\right].$$

- Verosimilitud de una sola observación

$$f(y_i; \theta_i, \phi) = e^{\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i, \phi)}.$$

- Logverosimilitud de una sola observación

$$\ell_i = \frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i, \phi).$$

- Logverosimitud

$$\ell = \sum_{i=1}^{n} \ell_i.$$

Distribución binomial

- Y_i es la **proporción muestral de éxitos**.
- $n_i Y_i \sim Bi(n_i, \pi_i)$.
- $\mu_i = EY_i = \pi_i$.
- Consideramos
$$\theta_i = \log \frac{\pi_i}{1-\pi_i}$$
esto es, definimos θ_i como el logit de la probabilidad de éxito.
- La transformación inversa es
$$\pi_i = \frac{e^{\theta_i}}{1+e^{\theta_i}},$$
y que $\log(1-\pi_i) = -\log\left(1+e^{\theta_i}\right)$.
-
$$f(y_i;\pi_i,n_i) = \binom{n_i}{n_i y_i} \pi_i^{n_i y_i}(1-\pi_i)^{n_i - n_i y_i} =$$
$$\exp\left[\frac{y_i\theta_i - \log[1+\exp(\theta_i)]}{1/n_i} + \log\binom{n_i}{n_i y_i}\right],$$
siendo
$$b(\theta_i) = \log[1+\exp(\theta_i)],\ a(\phi) = 1/n_i,\ c(y_i,\phi) = \log\binom{n_i}{n_i y_i}.$$
- El parámetro natural es
$$\theta_i = \log\frac{\pi_i}{1-\pi_i},$$
el logit de π_i.

Distribución Poisson

- La función de probabilidad es
$$f(y_i;\mu_i) = \frac{e^{-\mu_i}\mu_i^{y_i}}{y_i!} = \exp(y_i\log\mu_i - \mu_i - \log(y_i!)).$$
- Tomamos
$$\theta_i = \log\mu_i,$$
$$b(\theta_i) = \exp\theta_i,$$
$$a(\phi) = 1,$$
$$c(y_i,\phi) = -\log(y_i!).$$

La componente sistemática

- La componente sistemática de un modelo lineal generalizado es el vector $(\eta_1, \dots, \eta_n)$.
- Cada uno de los η_i es la combinación lineal de los predictores correspondientes a la i-ésima observación, es decir,
$$\eta_i = \sum_{j=1}^{p} \beta_j x_{ij},$$
con $i = 1, \dots, n$ y x_{ij} es j-ésimo predictor en el i-ésimo individuo.
- La combinación lineal $\sum_j \beta_j x_{ij}$ es el **predictor lineal**.
- Se suele asumir que $x_{i1} = 1$.

Función de enlace

- Mediante la **función de enlace** g relacionamos las componentes aleatoria y sistemática
$$g(\mu_i) = \eta_i = \sum_{j=1}^{p} \beta_j x_{ij}.$$
  - La función de enlace que nos transforma la media μ_i en el parámetro natural recibe el nombre de **enlace canónico**
$$\theta_i = \sum_{j=1}^{p} \beta_j x_{ij}.$$

Estimación de los coeficientes

- La función de verosimilitud es
$$L(\,) = \prod_{i=1}^{n} \exp\left\{ \frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i, \phi) \right\}$$
y la función de logverosimilitud es
$$\ell(\,) = \sum_{i=1}^{n} \frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + \sum_{i=1}^{n} c(y_i, \phi).$$
- Las ecuaciones de estimación son
$$\frac{\partial \ell(\,)}{\partial \beta_j} = \sum_{i=1}^{n} \frac{\partial \ell_i(\,)}{\partial \beta_j}$$
y tienen la expresión

$$\frac{\partial \ell(\ )}{\partial \beta_j} = \sum_{i=1}^{n} \frac{(y_i - \mu_i) x_{ij}}{var(Y_i)} \frac{\partial \mu_i}{\partial \eta_i} = 0,$$

con $j = 1, \ldots, p$ siendo

$$\eta_i = \sum_{j=1}^{p} \beta_j x_{ij} = g(\mu_i)$$

para la función de enlace g.

- Estas ecuaciones han de ser resueltas de un modo iterativo.
- La distribución de la variable Y_i aparece en las ecuaciones de verosimilitud solamente a través de su media μ_i y su varianza $var(Y_i)$.

Distribución de los estimadores máximo verosímiles

- La distribución asintótica de los coeficientes $\hat{}$ viene dada por

  $$\hat{} \sim N_p(\ , (\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1}),$$

  siendo $\mathbf{W}$ una matriz diagonal con las entradas

  $$w_i = \left(\frac{\partial \mu_i}{\partial \eta_i} \right)^2 \frac{1}{var(Y_i)}.$$

Covarianzas de $\hat{}$

- La matriz de covarianzas asintótica de $\hat{}$ viene dada por

  $$\widehat{var(\ \hat{}\ )} = (\mathbf{X}^T \hat{\mathbf{W}} \mathbf{X})^{-1}$$

  donde $\hat{\mathbf{W}}$ es la matriz $\mathbf{W}$ evaluada en $\hat{}$.

Comparando modelos anidados

- Podemos comparar modelos anidados.
- $\mathbf{X}_0$ y $\mathbf{X}_1$ matrices de modelo tales que $C(\mathbf{X}_0) \subset C(\mathbf{X}_1)$.
- El contraste de hipótesis que nos interesa es

  $$H_0 : \mathbf{g}(\ ) = \mathbf{X}_{0\ 0},$$
  $$H_1 : \mathbf{g}(\ ) = \mathbf{X}_{1\ 1}$$

  siendo $\mathbf{g}(\ ) = (g(\mu_1), \ldots, g(\mu_p))^T$.

- El test del cociente de verosimilitudes se basa en comparar los máximos que podemos alcanzar en la logverosimilitud bajo cada una de las hipótesis, $\ell(\hat{}_0)$ y $\ell(\hat{}_1)$, y utilizar el resultado que afirma
$$-2(\ell(\hat{}_0) - \ell(\hat{}_1)) \sim \chi^2_{p_1-p_0},$$
siendo p_0 y p_1 las dimensiones de los espacios columna $C(\mathbf{X}_0)$ y $C(\mathbf{X}_1)$.

Regresión logística

Datos

- Son datos anonimizados.

```
library(tamidata3)
finput = system.file("extdata","SNPs_RM.csv",package="tamidata3")
df = read.table(file = finput,header = TRUE,sep=",")
```

- Realizamos alguna modificación.

```
df$clinica.diabetes.mellitus.type2 =  (df$clinica.diabetes.mellitus.type2 == "YES")
df$clinica.gender = factor(df$clinica.gender)
```

- Podemos ver las primeras filas de los datos.

```
head(df,n=2)
```

```
  SNP1 SNP2 SNP3 SNP4 SNP5 clinica.diabetes.mellitus.type2 clinica.age
1 <NA>   AG   AG   GT   CT                           FALSE          40
2   AG   AA   AG   GT   TT                           FALSE          46
  clinica.gender clinica.weight.kg clinica.smoking.status
1           Male                80             non-smoker
2         Female                72              ex-smoker
```

- Los SNPs aparecen en las primeras columnas.
- Hemos de transformar estas variables para convertirlas en posibles predictores de acuerdo con distintos modelos genéticos.

```r
#' Transformation of the SNP to a genetic model
#' @description
#' Transformation of the SNP to a genetic model
#' @param x SNPs
#' @param type Model to be used
#' @param sep Separator
#' @export
snp2model = function(x,type=c("codominant","dominant","recessive"),
                     sep=""){
  x1 = substr(x,1,1)
  x2 = substr(x,2,2)
  a = table(c(x1,x2))
  recessive  = names(which.min(a))
  dominant = names(which.max(a))
  x1 = (x1 == recessive)*1
  x2 = (x2 == recessive)*1
  if(type == "codominant") rs =  factor(x1+x2)
  if(type == "dominant") rs = factor(x1+x2 == 0)
  if(type == "recessive") rs = factor(x1+x2 == 2)
  rs
}
```

- Vamos a evaluar cada uno de los modelos genéticos buscando asociación con la respuesta.

```r
snpscol = 1:5
df1 = df
df1[,snpscol] = apply(df1[,snpscol],2,snp2model,type="codominant")
df1$SNP1 = factor(df1$SNP1)
df1$SNP2 = factor(df1$SNP2)
df1$SNP3 = factor(df1$SNP3)
df1$SNP4 = factor(df1$SNP4)
df1$SNP5 = factor(df1$SNP5)
head(df1)
```

```
  SNP1 SNP2 SNP3 SNP4 SNP5 clinica.diabetes.mellitus.type2 clinica.age
1 <NA>    1    1    1    1                            FALSE          40
2    1    0    1    1    0                            FALSE          46
3 <NA> <NA> <NA>    2    1                             TRUE          78
4 <NA>    2 <NA>    1    0                            FALSE          66
5    0    0 <NA>    2    0                            FALSE          84
6    0    1    1    2    1                            FALSE          32
  clinica.gender clinica.weight.kg clinica.smoking.status
```

```
1           Male                 80              non-smoker
2         Female                 72               ex-smoker
3         Female                 NA              non-smoker
4           Male                 86               ex-smoker
5         Female                 68              non-smoker
6           Male                 69               ex-smoker
```

Ajustamos el modelo.

```
fit = glm(clinica.diabetes.mellitus.type2 =="YES" ~ ., family="binomial",data=df1)
```

```
Warning: glm.fit: algorithm did not converge
```

- Vemos un resumen del ajuste.

```
summary(fit)
```

```
Call:
glm(formula = clinica.diabetes.mellitus.type2 == "YES" ~ ., family = "binomial",
    data = df1)

Coefficients:
                                  Estimate Std. Error z value Pr(>|z|)
(Intercept)                     -2.657e+01  7.873e+04       0        1
SNP11                            5.154e-14  2.309e+04       0        1
SNP12                           -4.241e-15  6.269e+04       0        1
SNP21                           -4.044e-14  2.282e+04       0        1
SNP22                           -3.688e-14  3.141e+04       0        1
SNP31                            4.390e-14  2.465e+04       0        1
SNP32                            3.733e-15  2.825e+04       0        1
SNP41                            4.092e-14  2.637e+04       0        1
SNP42                            1.824e-14  3.225e+04       0        1
SNP51                           -4.043e-14  2.442e+04       0        1
SNP52                           -2.938e-14  5.340e+04       0        1
clinica.age                     -6.287e-16  5.806e+02       0        1
clinica.genderMale              -6.587e-14  2.634e+04       0        1
clinica.weight.kg                1.012e-15  9.675e+02       0        1
clinica.smoking.statusnon-smoker -8.103e-14  2.583e+04       0        1
clinica.smoking.statussmoker     -7.005e-14  3.034e+04       0        1
```

```
(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 0.0000e+00  on 1176  degrees of freedom
Residual deviance: 6.8285e-09  on 1161  degrees of freedom
  (325 observations deleted due to missingness)
AIC: 32

Number of Fisher Scoring iterations: 25
```

- Utilizamos el modelo recesivo.

```
snpscol = 1:5
df1 = df
df1[,snpscol] = apply(df1[,snpscol],2,snp2model,type="recessive")
fit = glm(clinica.diabetes.mellitus.type2 =="YES" ~ ., family="binomial",data=df1)
```

```
Warning: glm.fit: algorithm did not converge
```

```
summary(fit)
```

```
Call:
glm(formula = clinica.diabetes.mellitus.type2 == "YES" ~ ., family = "binomial",
    data = df1)

Coefficients:
                                  Estimate Std. Error z value Pr(>|z|)
(Intercept)                     -2.657e+01  7.485e+04       0        1
SNP1TRUE                         4.967e-15  6.213e+04       0        1
SNP2TRUE                         3.862e-15  2.877e+04       0        1
SNP3TRUE                         6.948e-15  2.390e+04       0        1
SNP4TRUE                         1.437e-14  2.508e+04       0        1
SNP5TRUE                         4.666e-15  5.164e+04       0        1
clinica.age                      1.314e-16  5.798e+02       0        1
clinica.genderMale               2.341e-14  2.627e+04       0        1
clinica.weight.kg               -3.164e-16  9.659e+02       0        1
clinica.smoking.statusnon-smoker  1.817e-14  2.574e+04       0        1
clinica.smoking.statussmoker      1.678e-14  3.028e+04       0        1

(Dispersion parameter for binomial family taken to be 1)
```

```
    Null deviance: 0.0000e+00  on 1176  degrees of freedom
Residual deviance: 6.8285e-09  on 1166  degrees of freedom
  (325 observations deleted due to missingness)
AIC: 22

Number of Fisher Scoring iterations: 25
```

Modelo loglineales de Poisson

Datos

```
pacman::p_load(SummarizedExperiment)
data(PRJNA218851,package="tamidata2")
```

```
table(colData(PRJNA218851)[,"Stage"])
```

```
    Cancer Metastasis     Normal
        18         18         18
```

```
df = data.frame(count = assay(PRJNA218851)[1001,],
                Stage=colData(PRJNA218851)[,"Stage"])
head(df)
```

```
                            count  Stage
SRR975551Aligned.out.sam.bam  4797 Cancer
SRR975552Aligned.out.sam.bam  3149 Cancer
SRR975553Aligned.out.sam.bam  3698 Cancer
SRR975554Aligned.out.sam.bam  3269 Cancer
SRR975555Aligned.out.sam.bam  3077 Cancer
SRR975556Aligned.out.sam.bam  7384 Cancer
```

Modelo loglineal de Poisson

Ajustamos un modelo loglineal de Poisson.

```r
fit = glm(count ~ Stage, family = poisson(link = log),data = df)
summary(fit)
```

```
Call:
glm(formula = count ~ Stage, family = poisson(link = log), data = df)

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)      8.237949   0.003833 2149.33   <2e-16 ***
StageMetastasis -0.455698   0.006153  -74.06   <2e-16 ***
StageNormal      0.376512   0.004977   75.65   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 74616  on 53  degrees of freedom
Residual deviance: 51857  on 51  degrees of freedom
AIC: 52398

Number of Fisher Scoring iterations: 5
```

- La desviacion nula es la desviacion para el modelo que tiene solo la constante.
- La desviacion residual es la desviacion del modelo que tiene la constante y las variables binarias que describen `Stage`.
- La diferencia entre los valores tiene una distribucion ji-cuadrado con dos grados de libertad y nos permite contrastar si los coeficientes de `StageMetastasis` y `StageNormal` pueden considerarse simultáneamente nulos.

```r
fit$null.deviance - fit$deviance
```

```
[1] 22758.5
```

Podemos rechazar confortablemente la hipotesis nula.

Sobredispersión

- En una distribución de Poisson, la media y la varianza son iguales.
- Cuando trabajamos con conteos reales no suele ser cierta esta hipótesis.
- Con frecuencia la varianza es mayor que la media.
- A esto se le llama **sobredispersión**.

```
fit1 = glm(count ~ Stage, family = quasipoisson(link = log),data = df)
summary(fit1)$dispersion
```

```
[1] 1223.854
```

GLM binomiales negativos

- La distribución binomial negativa no está en la familia de dispersión exponencial salvo que consideremos conocido el parámetro de dispersión.

- Hemos de estimarlo previamente.

```
library(MASS)
fit = glm.nb(count~Stage,data=df)
summary(fit)
```

```
Call:
glm.nb(formula = count ~ Stage, data = df, init.theta = 3.727352335,
    link = log)

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)       8.2379     0.1221  67.444  < 2e-16 ***
StageMetastasis  -0.4557     0.1728  -2.638  0.00835 **
StageNormal       0.3765     0.1727   2.180  0.02927 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(3.7274) family taken to be 1)

    Null deviance: 79.058  on 53  degrees of freedom
Residual deviance: 56.400  on 51  degrees of freedom
AIC: 966.78
```

```
Number of Fisher Scoring iterations: 1

              Theta:  3.727
          Std. Err.:  0.689

 2 x log-likelihood:  -958.781
```

- Podemos ver que los errores estándar de los coeficientes son mucho mayores porque tenemos en cuenta la sobre dispersión que el modelo de Poisson no lo considera.