UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Fundació Politècnica de Catalunya

Métodos cuantitativos II: modelos lineales generalizados

Jose M Sallan jose.maria.sallan@upc.edu

15 de febrero de 2018

Outline

Modelos lineales generalizados

Modelos para variables binarias

Modelos para recuentos

Modelos lineales generalizados

Un modelo ineal generalizado tiene tres componentes:

- **Componente aleatoria:** identifica la variable dependiente y, y su distribución de probabilidad
- **Componente sistemática:** especifica la función predictora lineal de las variables independientes x_j
- **Función link:** es una función del valor esperado de y

La componente aleatoria consiste en las observaciones $y_1, \ldots, y_n$ de una variable dependiente, que puede tener diversas distribuciones de probabilidad de la *familia exponencial*:

- **variable binaria:** distribución binomial
- **variable recuento:** distribución Poisson
- **variable continua:** distribución normal

La componente sistemática es una función lineal de las variables predictoras:

$$\alpha + \beta_1 x_1 + \cdots + \beta_p x_p$$

Esta función nos permite tener una predicción de y para cada observación i:

$$\eta_i = \sum_j \beta_j x_{ij}$$

La función de enlace relaciona el valor esperado de la variable dependiente $\mu = E(y)$ con el predictor lineal η_i:

$$g(\mu) = \alpha + \beta_1 x_1 + \cdots + \beta_p x_p$$

Para cada observación, tenemos:

$$\mu_i = E(y_i)$$
$$\eta_i = g(\mu_i) = \sum_j \beta_j x_{ij}$$

Modelos lineales generalizados

Modelos para variables binarias

Modelos para recuentos

Regresión para variables binarias

Es frecuente que tengamos modelos en los que la variable respuesta sea binaria (éxito / fracaso):

- Gana o pierde el partido
- Participa o no participa en el mercado de trabajo
- Compra o no compra el partido

Para este tipo de variables tenemos dos modelos lineales generalizados:

- Regresión logística o **logit**: la función de enlace es la función logísitca
- Regresión **probit**: la función de enlace es la función normal

Regresión logística

En el modelo de regresión logística, la función de enlace que relaciona la probabilidad de éxito p con la predicción es:

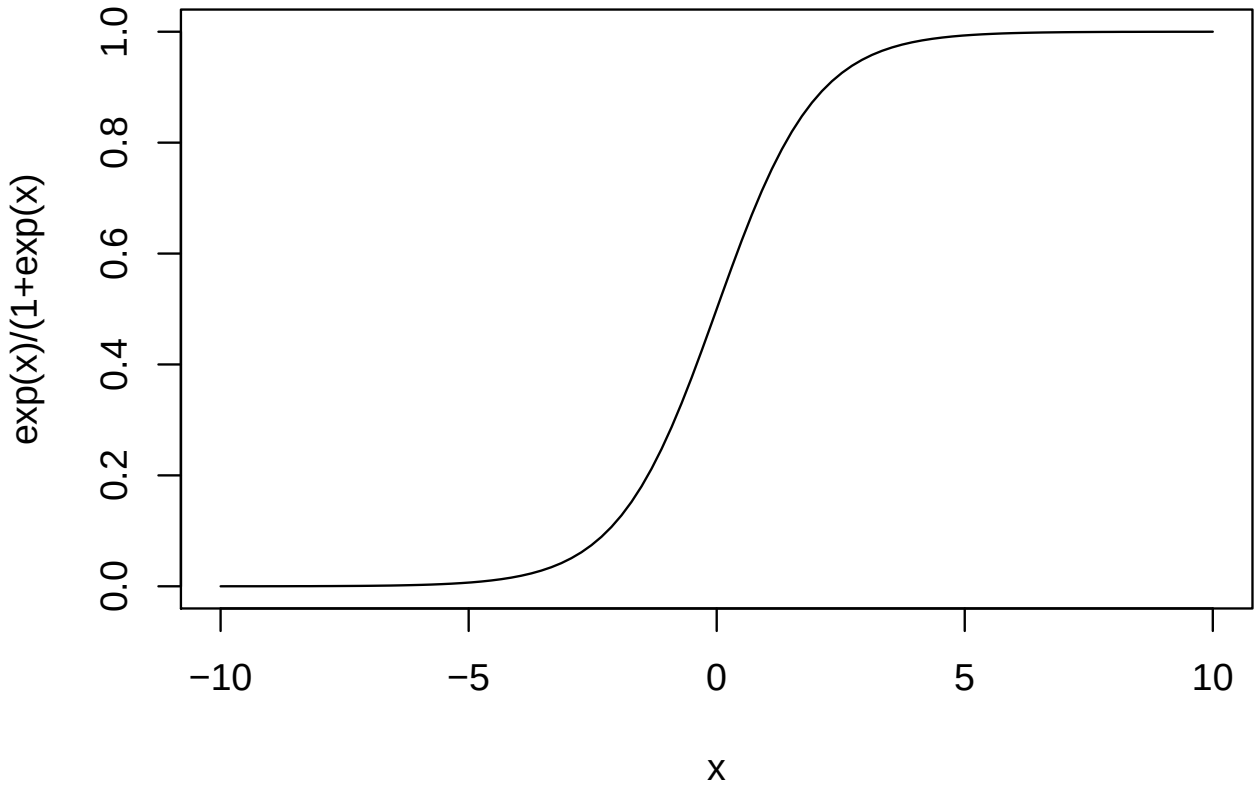
$$\eta_i = \log\left(\frac{p_i}{1-p_i}\right) = \sum_j \beta_j x_{ij}$$

Así que la probabilidad es igual a:

$$p_i = \frac{exp\left(\sum \beta_j x_{ij}\right)}{1 + exp\left(\sum \beta_j x_{ij}\right)}$$

Regresión logística

Forma de la función logística:



Regresión logística

Ejemplo `RavensData`

`RavensData` contiene datos de una serie de partidos de los Ravens, con las siguientes variables:

- `ravenWinNum`: indica si los Ravens ganan (1=gana, 0=pierde)
- `ravenScore`: puntuación de los Ravens en cada partido
- `opponentScore`: puntuación del oponente en cada partido

Queremos evaluar la influencia de la puntuación de los Ravens en que los Ravens ganen el partido.

Regresión logística

Ejemplo RavensData

Los modelos lineales generalizados se estiman con la función `glm`. Para especificar que se estima regresión logística se indica que el modelo es de la familia `binomial`

```
> load("ravensData.rda")
> logit.ravens <- glm(ravenWinNum ~ ravenScore,
+                     data=ravensData, family = "binomial")
```

Regresión logística

Ejemplo RavensData


```
> summary(logit.ravens)

Call:
glm(formula = ravenWinNum ~ ravenScore, family = "binomial",
    data = ravensData)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7575  -1.0999   0.5305   0.8060   1.4947

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.68001    1.55412  -1.081     0.28
ravenScore   0.10658    0.06674   1.597     0.11

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 24.435  on 19  degrees of freedom
Residual deviance: 20.895  on 18  degrees of freedom
AIC: 24.895

Number of Fisher Scoring iterations: 5
```

Regresión logística

Interpretación de los coeficientes

El exponencial de los coeficientes indica la variación de cuota (odds) por unidad de la variable dependiente

```
> odds <- exp(coef(logit.ravens))
> odds
```

```
(Intercept)  ravenScore
  0.1863724   1.1124694
```

Regresión logística

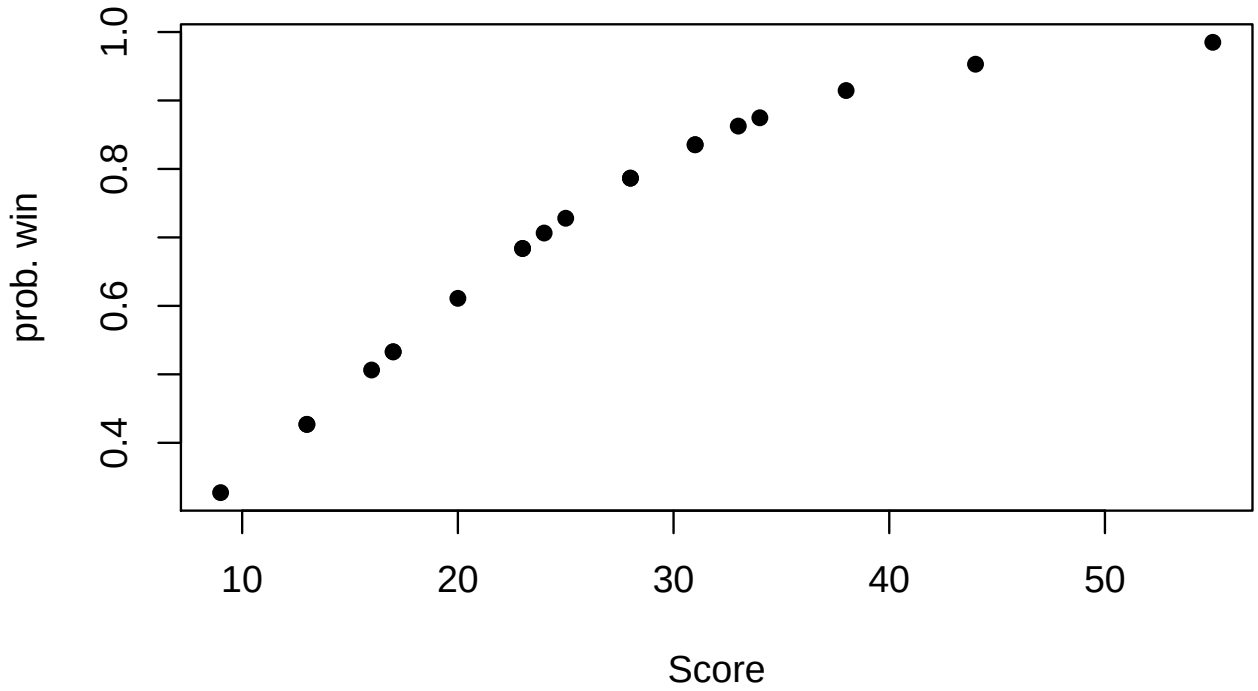
Interpretación de los coeficientes

Los `fitted.values` o `fitted` del modelo indican la probabilidad de ganar para cada observación

```
> plot(ravensData$ravenScore, logit.ravens$fitted,
+      pch=16, xlab = "Score", ylab="prob. win")
```

Regresión logística

Interpretación de los coeficientes



Regresión logística

Diagnósticos del modelo

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Fundació Politècnica de Catalunya

ANOVA del modelo:

```
> anova(logit.ravens, test = "Chisq")

Analysis of Deviance Table

Model: binomial, link: logit

Response: ravenWinNum

Terms added sequentially (first to last)


           Df Deviance Resid. Df Resid. Dev Pr(>Chi)
NULL                         19     24.435
ravenScore  1   3.5398        18     20.895  0.05991 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Regresión logística

Diagnóstico del modelo

Puede calcularse una pseudo-R^2 comparando la *log likelihood* del modelo con la del modelo nulo (sin regresores):

```
> logit.ravens0 <- update(logit.ravens, . ~ 1)
> pseudoR2 <- 1 -
+   as.vector(logLik(logit.ravens)/logLik(logit.ravens0))
> pseudoR2
```

```
[1] 0.1448696
```

Regresión probit

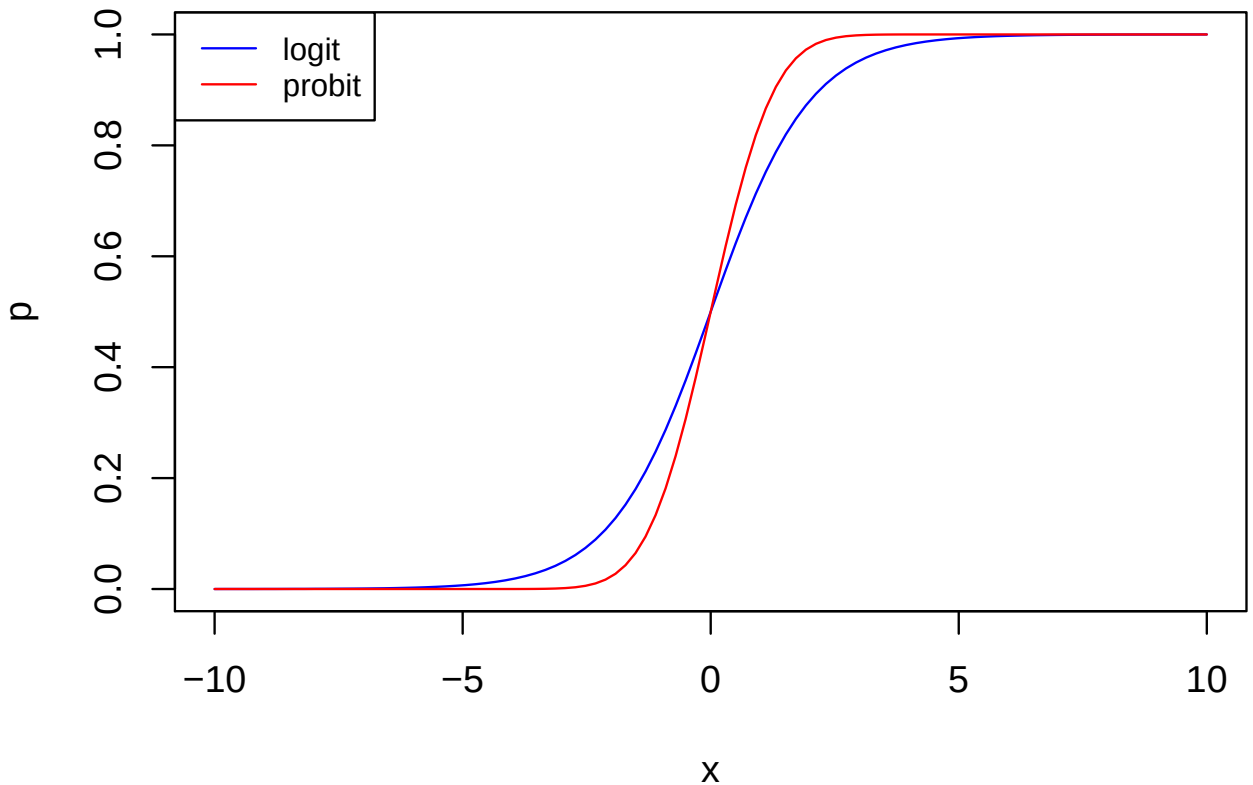
La regresión probit es una alternativa a la regresión logit para variables dependientes binarias

La función de enlace en un modelo probit es la distribución normal

Ambos modelos tienen resultados similares

Regresión probit

Funciones de enlace logit y probit



Regresión probit

Ejemplo `RavensData`

Para la regresión probit, en `glm` hay que especificar la familia `binomial("probit")`

```
> probit.ravens <- glm(ravenWinNum ~ ravenScore,
+                      data=ravensData,
+                      family = binomial("probit"))
```

Regresión probit

Ejemplo RavensData

```
> summary(probit.ravens)

Call:
glm(formula = ravenWinNum ~ ravenScore, family = binomial("probit"),
    data = ravensData)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7684  -1.1040   0.5081   0.8036   1.4889

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.03085    0.92970  -1.109   0.2675
ravenScore   0.06569    0.03868   1.698   0.0894 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 24.435  on 19  degrees of freedom
Residual deviance: 20.738  on 18  degrees of freedom
AIC: 24.738

Number of Fisher Scoring iterations: 6
```

Modelos lineales generalizados

Modelos para variables binarias

Modelos para recuentos

Regresión para recuentos

Es frecuente que la variable a predecir sea un recuento:

- Llamadas a un *call center*
- Número de accidentes por empresa
- Número de coches que cruzan un puente

También podemos tener tasas como variable dependiente:

- Porcentaje de estudiantes que superan un examen
- Porcentaje de accesos a una web desde un país

Variable discreta, no negativa

Regresión para recuentos

Modelos lineales para analizar recuentos como variable dependiente:

- Regresión de Poisson
- Regresión binomial negativa

Regresión de Poisson

En la regresión de Poisson, la función de enlace es el logaritmo de la variable dependiente:

$$\eta_i = \log(y_i) = \sum_j \beta_j x_{ij}$$

Regresión de Poisson

Ejemplo `RecreationDemand`

Los datos `RecreationDemand` del paquete `AER` indica el número de viajes de recreo `trips` en barco de los visitantes de Lake Somerville (Texas).
La estimación de un modelo de Poisson para estos datos es:

```
> library(AER)
> data("RecreationDemand")
> rd.poisson <- glm(trips ~ ., data=RecreationDemand,
+                   family = poisson)
```

Regresión de Poisson

Ejemplo RecreationDemand


```
> coeftest(rd.poisson)

z test of coefficients:

              Estimate Std. Error  z value  Pr(>|z|)
(Intercept)  0.2649934  0.0937222   2.8274  0.004692 **
quality      0.4717259  0.0170905  27.6016 < 2.2e-16 ***
skiyes       0.4182137  0.0571902   7.3127 2.619e-13 ***
income      -0.1113232  0.0195884  -5.6831 1.323e-08 ***
userfeeyes   0.8981653  0.0789851  11.3713 < 2.2e-16 ***
costC       -0.0034297  0.0031178  -1.1001  0.271309
costS       -0.0425364  0.0016703 -25.4667 < 2.2e-16 ***
costH        0.0361336  0.0027096  13.3353 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Regresión de Poisson

Interpretación de los coeficientes

El modelo de Poisson es de la forma:

$$\log(y) = \alpha + \beta_1 x_1 + \cdots + \beta_p x_p$$

De manera que:

$$y = exp(\alpha + \beta_1 x_1 + \cdots + \beta_p x_p)$$

$$y = exp(\alpha)\, exp(\beta_1 x_1) \ldots exp(\beta_p x_p)$$

Si x_j aumenta en una unidad, y se multiplica por $exp(b_j)$

Regresión de Poisson

Sobredispersión

En la distribución de Poisson, la media y la varianza son iguales, pero en recuentos reales la varianza suele ser mayor que la media *(sobredispersión)*

```
> var(RecreationDemand$trips)/mean(RecreationDemand$trips)
```

```
[1] 17.6425
```

Para estas situaciones, puede utilizarse una regresión **binomial negativa**

Regresión binomial negativa

En la regresión binomial negativa tenemos que:

$$E(y) = \mu$$
$$V(y) = \mu + \frac{\mu^2}{\theta}$$

Donde $1/\theta$ es un parámetro de dispersión. Si $\theta \to \infty$ converge a una regresión Poisson.
La función `glm.nb` del paquete `MASS` calcula un modelo de regresión binomial negativa en el que estima el parámetro θ.

Regresión binomial negativa

Ejemplo `RecreationDemand`

```
> library(MASS)
> rd.nb <- glm.nb(trips ~ ., data=RecreationDemand,
+                 init.theta = 0.9)
```

Regresión binomial negativa

Ejemplo `RecreationDemand`

El ajuste del modelo binomial negativo es mejor que el Poisson

```
> coeftest(rd.nb)

z test of coefficients:

              Estimate Std. Error  z value  Pr(>|z|)
(Intercept) -1.1219364  0.2143029  -5.2353 1.647e-07 ***
quality      0.7219990  0.0401165  17.9976 < 2.2e-16 ***
skiyes       0.6121388  0.1503029   4.0727 4.647e-05 ***
income      -0.0260588  0.0424527  -0.6138   0.53933
userfeeyes   0.6691677  0.3530211   1.8955   0.05802 .
costC        0.0480086  0.0091848   5.2269 1.723e-07 ***
costS       -0.0926910  0.0066534 -13.9314 < 2.2e-16 ***
costH        0.0388357  0.0077505   5.0107 5.423e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Conclusiones

Los modelos lineales generalizados permiten obtener modelos de regresión para:

- **Variable dependiente binaria:** modelos logit y probit
- **Variable dependiente recuento:** modelos Poisson y binomial negativa

Los modelos lineales generalizados pueden estimarse con la función `glm` de **R**