Aprendizaje supervisado

Maestría en Estadística

Pablo Vena

Universidad de Buenos Aires

23 de octubre de 2021

Lecturas

ISLR Capítulo 4.

ESL Capítulo 2.

Problema de clasificación

- La variable de respuesta Y es cualitativa (categórica) en lugar de continua.
- Clasificar una observación consiste en predecir la respuesta cualitativa.
- Existen numerosas técnicas de clasificación o *clasificadores*:
  - KNN
  - Regresión logística
  - Naive Bayes
  - SVM
  - Random Forest
  - Redes neuronales
  - ...

Aplicaciones

- adtech: predicción de CVR, CTR, etc.
- análisis de sentimiento (NLP)
- Churn
- detección de transacciones fraudulentas
- clasificación de tumores
- clasificación de imágenes
- ...

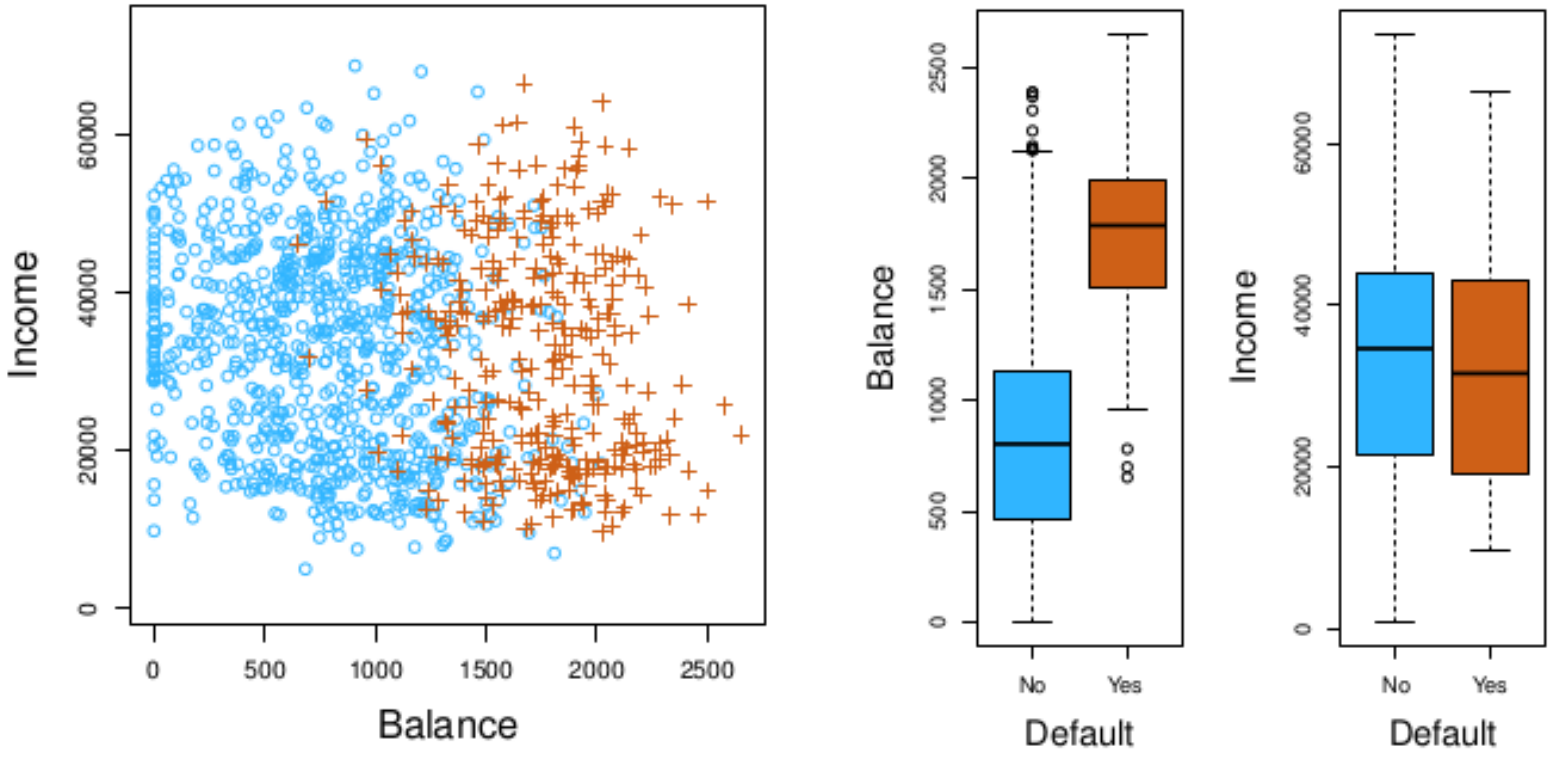
Dataset: *Default*

```
library(ISLR)
head(Default[sample(nrow(Default)), ], 10)
```

id	default	student	balance	income
1613	No	No	745.8132	42762.47
7193	No	Yes	666.4244	19177.68
6519	No	No	131.6340	42028.01
4454	No	No	486.5761	27687.34
9438	Yes	No	961.7327	27600.42
1713	No	No	381.3875	28265.09
1497	Yes	No	2074.8076	38988.86
5811	No	No	857.6997	43162.67
7650	No	No	407.2634	40524.34
7194	No	Yes	1017.6671	13553.09

Cuadro: Diez observaciones al azar del dataset *Default* del paquete ISLR.

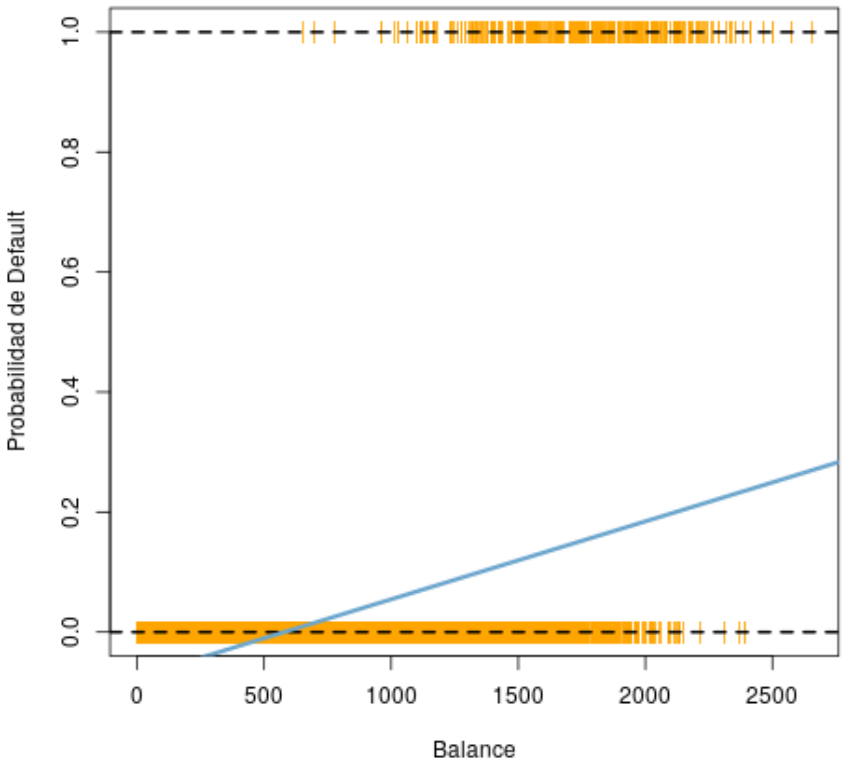
Dataset: *Default*



Modelo lineal

Si planteamos un modelo lineal entre `default` y `balance`:

```
balance = Default$balance
default = as.numeric(Default$default) - 1
fit = lm(default ~ balance)
```



- algunas estimaciones caen fuera del intervalo $[0, 1]$ (algunas son negativas),
- no funcionaría para más de una clase.

Regresión logística

En lugar de modelar la variable de respuesta Y directamente, la regresión logística modela la probabilidad de que Y pertenezca a una categoría en particular:

$$P(\texttt{default} = \texttt{Yes}|\texttt{balance})$$

- Los valores $P(\texttt{default} = \texttt{Yes}|\texttt{balance})$, que abreviamos $p(\texttt{balance})$ van de 0 a 1.
- Luego, para cualquier valor de `balance` podemos predecir el valor de `default`. Por ejemplo, podríamos predecir

$$\texttt{default = Yes}$$

para cualquier observación tal que $p(\texttt{balance} > 0{,}5)$.

Podríamos tener una postura más conservadora si fijamos el valor de corte como $p(\texttt{balance}) > 0{,}1$.

Modelo logístico

¿Cómo modelamos la relación entre $p(X) = P(Y = 1|X)$ y X?

- Si consideramos una regresión lineal

$$p(X) = \beta_0 + \beta_1 X$$

para balances cercanos a cero predecimos probabilidades negativas y para balances grandes, mayores a 1.

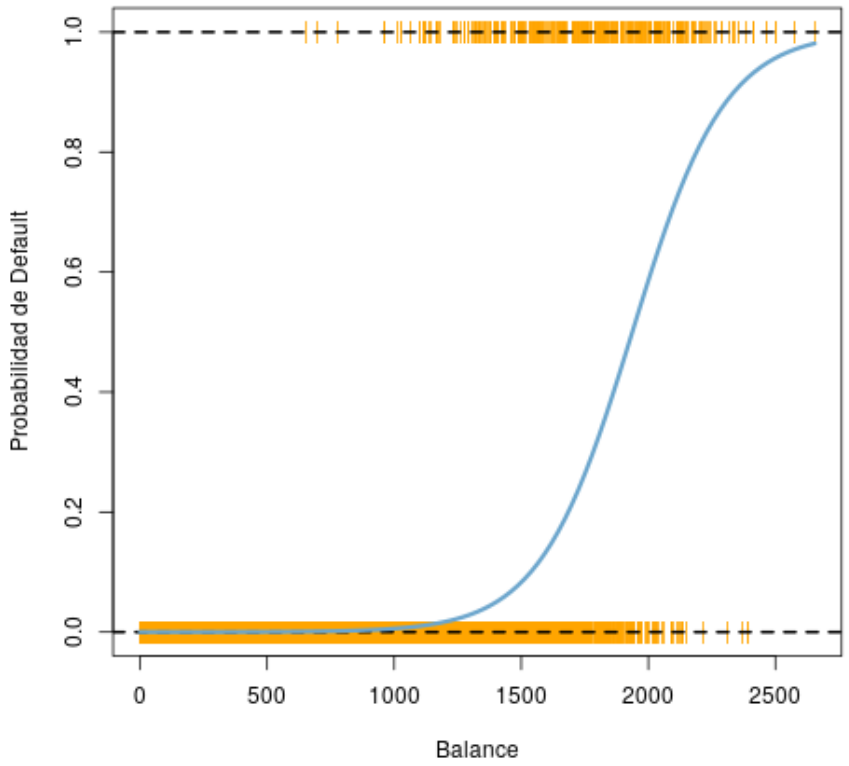
- Buscamos modelar $p(X)$ con una función cuyo rango sea el intervalo $[0, 1]$:

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}$$

es la **función logística**. Para balances cercanos a cero, la probabilidad es cercana a 0 pero nunca menor.De la misma forma, para balances grandes la probabilidad es alta pero nunca mayor a 1.

Regresión logística

Los valores de β_0 y β_1 se obtienen por el **método de máxima verosimilitud**.



Regresión logística

La función logística puede reescribirse en término de los **odds**

$$\frac{p(X)}{1-p(X)} = e^{\beta_0+\beta_1}$$

que toman valores entre 0 y ∞. Valores cercanos a 0 e ∞ indican baja y alta probabilidad de `default`, respectivamente.

Por ejemplo, si en promedio 1 de cada 5 personas no paga, $p(X) = 0{,}2$ y los odds son

$$\frac{p(X)}{1-p(X)} = \frac{0{,}2}{1-0{,}2} = \frac{1}{4}$$

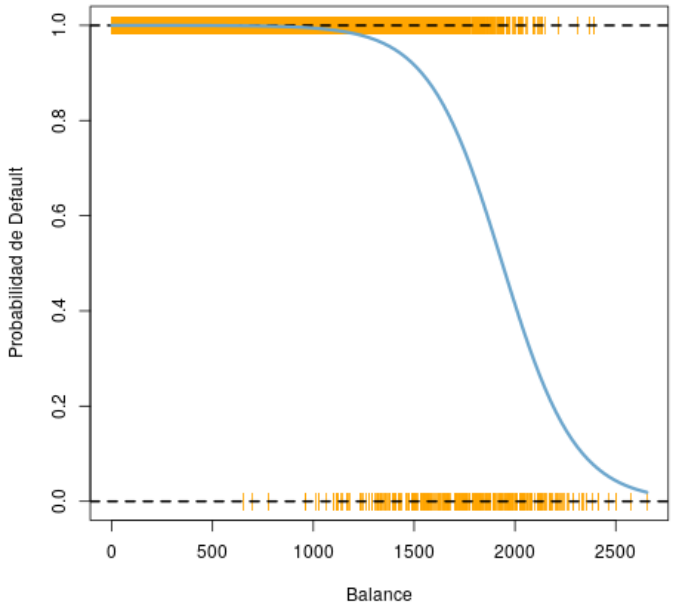
Si tomamos logaritmo, vemos que la regresión logística es una regresión lineal entre X y los **log-odds**

$$log\left(\frac{p(X)}{1-p(X)}\right) = \beta_0 + \beta_1 X$$

Regresión logística

- En el modelo lineal, β_1 es el cambio esperado en Y asociado a un incremento de X en una unidad.
- En el modelo logístico, el incremento en una unidad de X está asociado al cambio de los *logodds* (equivalentemente, multiplica los *odds* por e^{β_1})
- Cuánto cambia $p(X)$ depende del valor de X.

- si $\beta_1 > 0$, aumentar X estará asociado a aumentar $p(X)$,
- si $\beta_1 < 0$, aumentar X estará asociado a disminuir $p(X)$.



Clasificador

Para minimizar el error de mala clasificación, predecimos

$$Y = 1 \text{ si } p(X) \geq 0{,}5$$
$$Y = 0 \text{ si } p(X) \leq 0{,}5$$

Despejando $p(x)$, esto significa predecir

$$Y = 1 \text{ si } \beta_0 + x \cdot \beta \geq 0$$
$$Y = 0 \text{ si } \beta_0 + x \cdot \beta \leq 0$$

La regresión logística es un clasificador lineal. La frontera de decisión que separa a las dos clases es el hiperplano que resulta como solución de

$$\beta_0 + x \cdot \beta = 0$$

Clasificador

Ejercicio

La distancia de una observación a la frontera de decisión es

$$\frac{\beta_0}{\|\beta\|} + x \cdot \frac{\beta}{\|\beta\|}$$

La regresión logística indica la frontera de clasificación y además que la probabilidad de pertenencia depende de la distancia a la frontera. En particular va hacia los extremos (0 y 1) cuando $\|\beta\|$ es grande.

Máxima verosimilitud

Los coeficientes β_0 y β se estiman a partir de los datos

$$(x_1, y_1), \ldots, (x_n, y_n)$$

al maximizar la función de verosimilitud (condicional):

$$L(\beta_0, \beta) = \prod_{i=1}^{n} p(x_i)^{y_i} (1 - p(x_i))^{1-y_i}$$

Máxima verosimilitud

Tomamos logaritmo, y reemplazamos por la función logística:

$$
\begin{aligned}
\ell(\beta_0, \beta) &= \sum_{i=1}^{n} y_i \log p(x_i) + (1 - y_i) \log(1 - p(x_i)) \\
&= \sum_{i=1}^{n} \log(1 - p(x_i)) + \sum_{i=1}^{n} y_i \log\left(\frac{p(x_i)}{1 - p(x_i)}\right) \\
&= \sum_{i=1}^{n} \log(1 - p(x_i)) + \sum_{i=1}^{n} y_i(\beta_0 + \beta \cdot x_i) \\
&= \sum_{i=1}^{n} -\log(1 + e^{\beta_0 + \beta \cdot x_i}) + \sum_{i=1}^{n} y_i(\beta_0 + \beta \cdot x_i)
\end{aligned}
$$

Máxima verosimilitud

Derivamos e igualamos a cero:

$$\frac{\partial \ell}{\partial \beta_j} = -\sum_{i=1}^{n} \frac{e^{\beta_0 + x_i \cdot \beta} x_{ij}}{1 + e^{\beta_0 + x_i \cdot \beta}} + \sum_{i=1}^{n} y_i x_{ij} = \sum_{i=1}^{n} (y_i - p(x_i; \beta_0, \beta)) x_{ij}$$

La ecuación no puede despejarse, requiere un método numérico.

La función $-\ell(\beta_0, \beta)$ se conoce como la pérdida **cross-entropy**.

Método de Newton-Raphson

Dada una función de una variable, $f(\beta)$, buscamos el mínimo global β^*. Supongamos que g es suave y que β^* es un mínimo regular ($f'(\beta^*) = 0$ y $f(\beta^*) > 0$).

En un entorno del β^* vale el desarrollo de Taylor:

$$f(\beta) \approx f(\beta^*) + \frac{1}{2}(\beta - \beta^*)^2 f''(\beta^*)$$

Cerca del mínimo, $f(\beta)$ es aproximadamente cuadrática. El método de Newton-Raphson minimiza esta aproximación cuadrática.

Método de Newton-Raphson

Proponemos un valor inicial β_0, si está cerca del mínimo repetimos el desarrollo de Taylor

$$f(\beta) \approx f(\beta_0) + (\beta - \beta_0)f'(\beta_0) + \frac{1}{2}(\beta - \beta_0)^2 f''(\beta_0)$$

Resolvemos el problema cuadrático

$$0 = f'(\beta_0) + \frac{1}{2}f''(\beta_0) \cdot 2(\beta - \beta_0)$$

Despejamos y obtenemos un valor β_1 que debería ser una mejor aproximación de β^* que el inicial β_0:

$$\beta_1 = \beta_0 - \frac{f'(\beta_0)}{f''(\beta_0)}$$

Si iteramos este proceso obtenemos la recurrencia:

$$\beta_{n+1} = \beta_n - \frac{f'(\beta_n)}{f''(\beta_n)}$$

Método de Newton-Raphson

Convergencia

El mínimo global β^* es un punto fijo de la ecuación anterior. Se puede probar que si β_0 está lo suficientemente cerca de β^* entonces $\beta_n \to \beta^*$ y que la convergencia es cuadrática:

$$|\beta_n - \beta^*| = O(n^{-2})$$

(duplicar la cantidad de iteraciones reduce el error cuatro veces)

Método de Newton-Raphson

Varias dimensiones

El método se generaliza al caso de varias variables $\beta_1, \ldots, \beta_p$.

Si llamamos $w = (\beta_1, \ldots, \beta_p)$, la iteración del método resulta:

$$\beta_{n+1} = \beta_n - H^{-1}(\beta_n)\nabla f(\beta_n)$$

donde ∇f es el gradiente de f y H su matriz Hessiana.

El cálculo de la inversa de H puede ser costoso, lo que lleva a diferentes métodos quasi-Newton, como L-BFGS, que intentan actualizar $H^{-1}(\beta_n)$ mientras β_n cambia.

Propiedades asintóticas

La teoría asintótica del método de máxima verosimilitud nos garantiza, bajo ciertas condiciones de regularidad:

- consistencia
- insesgadez asintótica
- normalidad asintótica
- eficiencia

Además nos permite hacer inferencia sobre los parámetros de la regresión:

- desvío estándar
- intervalo de confianza
- tests (Wald test)

Softmax

Para cada clase $k = 1, \ldots, K$ modelamos

$$P(Y = k|X = x) = \frac{e^{\beta_{k0}+\beta_{k1}x_1+\cdots+\beta_{kp}x_p}}{\sum_{j=1}^{K} e^{\beta_{j0}+\beta_{j1}x_1+\cdots+\beta_{jp}x_p}}$$

Estimamos para todas las clases.

$$\log\left(\frac{P(y = k|X = x)}{P(Y = k'|X = x)}\right) = (\beta_{k0}-\beta_{k'0})+(\beta_{k1}-\beta_{k'1})x_1+\cdots+(\beta_{kp}-\beta_{k'p})x_p$$

Extensiones

- colinealidad
- outliers
- datos funcionales

Notch dataset

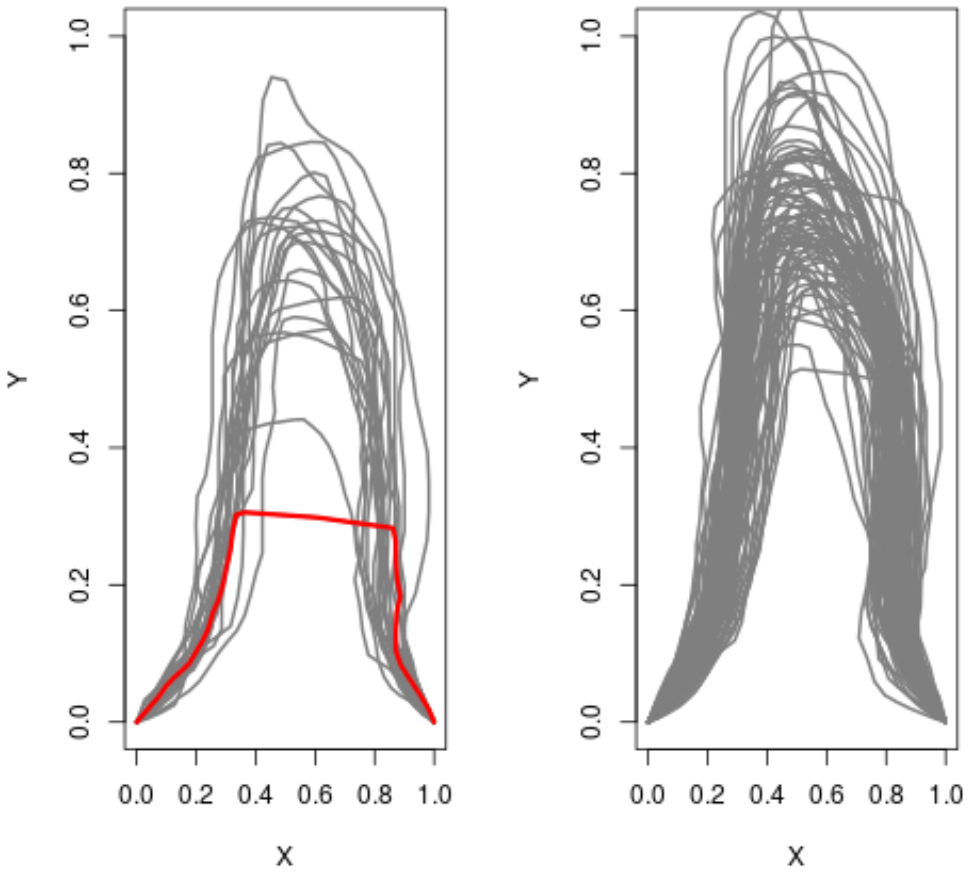


Figura: Ramsay y Silverman, 2002.

Ejercicio

1. Reproducir las figuras del modelo lineal y logístico para el dataset `Default`.
2. Implementar una función que realice una regresión logística múltiple para dos clases y devuelva coeficientes, las probabilidades estimadas y las clases predichas. Compararla con `glm` para el dataset `Default`.
    - Extra: errores estándar e intervalos de confianza para los parámetros (comparar con `confint`).
3. Entrenar un clasificador basado en regresión logística multinomial para el dataset MNIST (sugerencia: usar la función `multinom` el paquete nnet).

Machine learning

En lenguaje de *machine learning*:

1. A **feature representation** of the input. For each input observation $x^{(i)}$, this will be a vector of features $[x_1, x_2, ..., x_n]$. We will generally refer to feature i for input $x^{(j)}$ as $x_i^{(j)}$, sometimes simplified as x_i, but we will also see the notation f_i, $f_i(x)$, or, for multiclass classification, $f_i(c, x)$.
2. A classification function that computes $\hat{y}$, the estimated class, via $p(y|x)$. In the next section we will introduce the **sigmoid** and **softmax** tools for classification.
3. An objective function for learning, usually involving minimizing error on training examples. We will introduce the **cross-entropy loss function**.
4. An algorithm for optimizing the objective function. We introduce the **stochastic gradient descent** algorithm.

Logistic regression has two phases:

training: we train the system (specifically the weights w and b) using stochastic gradient descent and the cross-entropy loss.

test: Given a test example x we compute $p(y|x)$ and return the higher probability label $y = 1$ or $y = 0$.

Figura: Speech and Language Processing. Jurafsky & Martin.