

Regresión logística

Miguel Nunez-del-Prado, Ph. D.

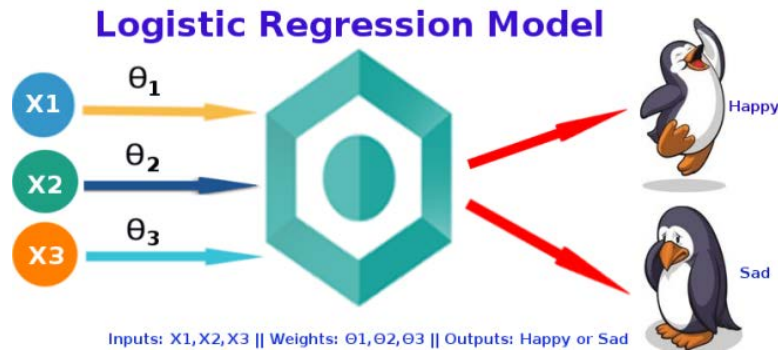
Aplicaciones de la regresión logística



Analítica sanitaria:
Regresión logística para
reducir los reingresos
de pacientes



Regresión logística



es un **clasificador discriminativo** que se utiliza para describir datos y explicar la relación entre una **variable binaria dependiente** y una o más **variables independientes** nominales, u ordinales.

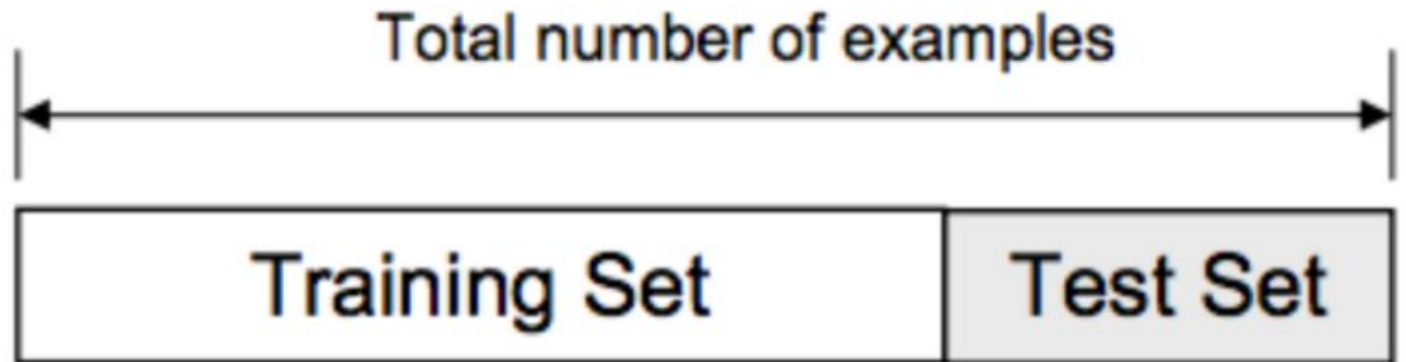
Regresión logística

Los clasificadores de aprendizaje supervisado requieren un conjunto de datos de entrenamiento de m pares de entrada/salida $x^{(i)}, y^{(i)}$. Además, necesitamos:

- Un conjunto de características como entrada. Para cada observación de entrada $x^{(i)}$, que es un vector de características $[x_1; x_2; \dots; x_n]$
- Una función de clasificación que calcula la clase estimada \hat{y} , mediante $p(y|x)$ (e.g., **Sigmoid**)
- Una función objetivo para el aprendizaje, que suele implicar la minimización del error en los ejemplos de entrenamiento (e.g., **cross-entropy loss function**)
- Un algoritmo para optimizar la función objetivo. (algoritmo de **descenso de gradiente estocástico**).

La regresión logística tiene dos fases:

1. **entrenamiento**: entrenamos los pesos w y b utilizando el descenso estocástico-gradiente y la pérdida de entropía cruzada.
2. **prueba**: Dado un ejemplo de prueba x , calculamos $p(y/x)$ y devolvemos la etiqueta de mayor probabilidad $y = 1$ o $y = 0$.



Clasificación



Dada un **vector de observación** $x = [x_1; x_2; \dots; x_n]$. La **salida** del clasificador y puede ser **1** (lo que significa que la observación es un miembro de la clase) o **0** (la observación no es un miembro de la clase). Queremos saber la **probabilidad** $P(y = 1/x)$ de que esta observación sea un **miembro de la clase**.



¿Cómo aprender los pesos (w) y término de sesgo (b)?

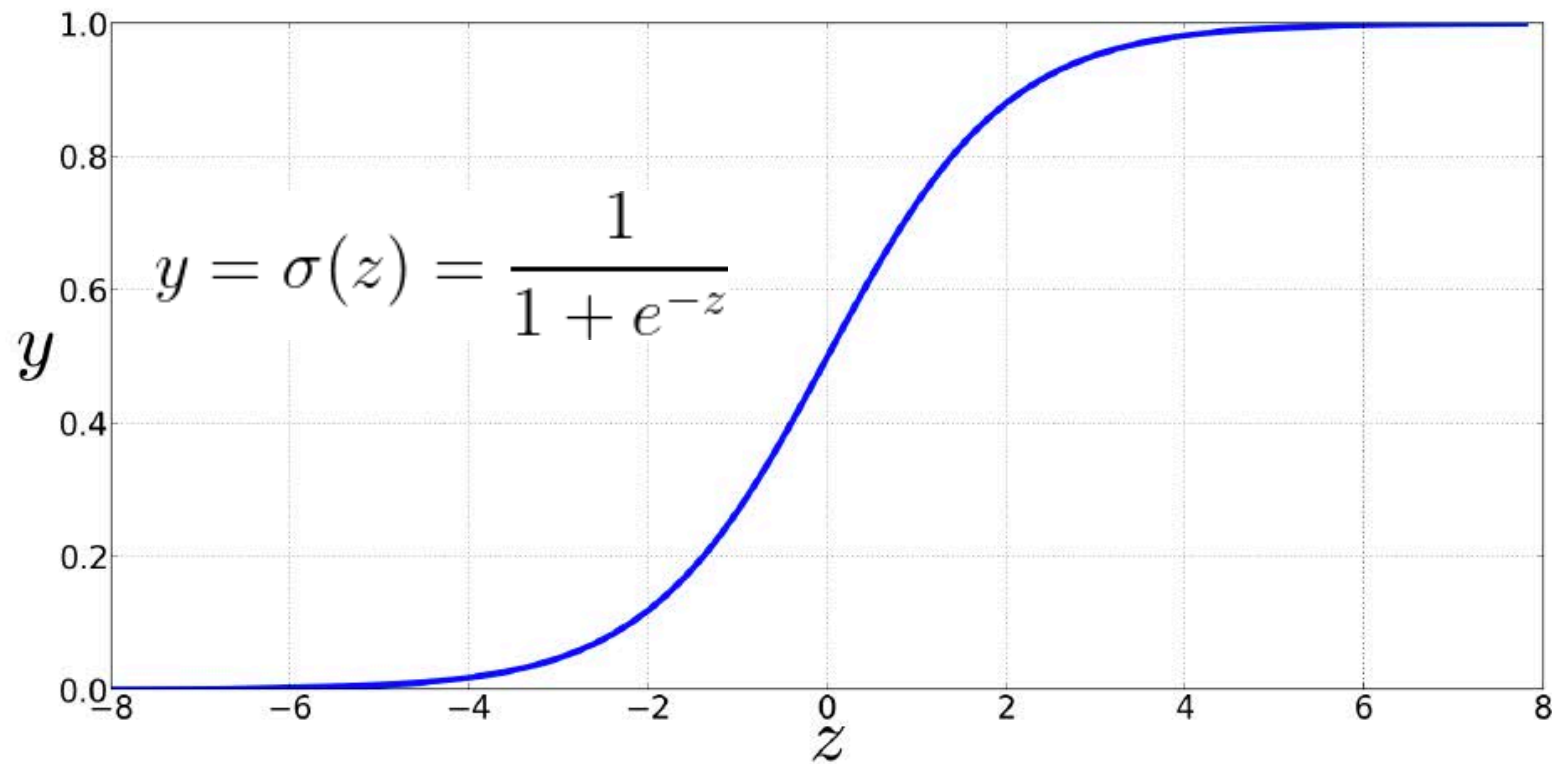
La **regresión logística** aprende w y b , a partir de un conjunto de **entrenamiento**. Cada peso w_i es un número real y está **asociado** a una de las **características** de x_i . La ponderación w_i representa la importancia de esa característica de entrada para la decisión de clasificación, y puede ser positiva o negativa.

¿Cómo tomar una decisión?

el clasificador **multiplica** primero cada x_i por su **peso** w_i , suma las características ponderadas y añade el término de sesgo b . El número único resultante z expresa la **suma ponderada de la evidencia para la clase**.

$$z = \left(\sum_{i=1}^n w_i x_i \right) + b$$
$$z = w \cdot x + b$$





La función sigmoide

Para obtener una **probabilidad**, pasaremos **z** por la función sigmoidea, **$\sigma(z)$** . La **función sigmoide o función logística**, da nombre a la regresión logística.

¿Cómo
asignarle una
clase a una
observación?



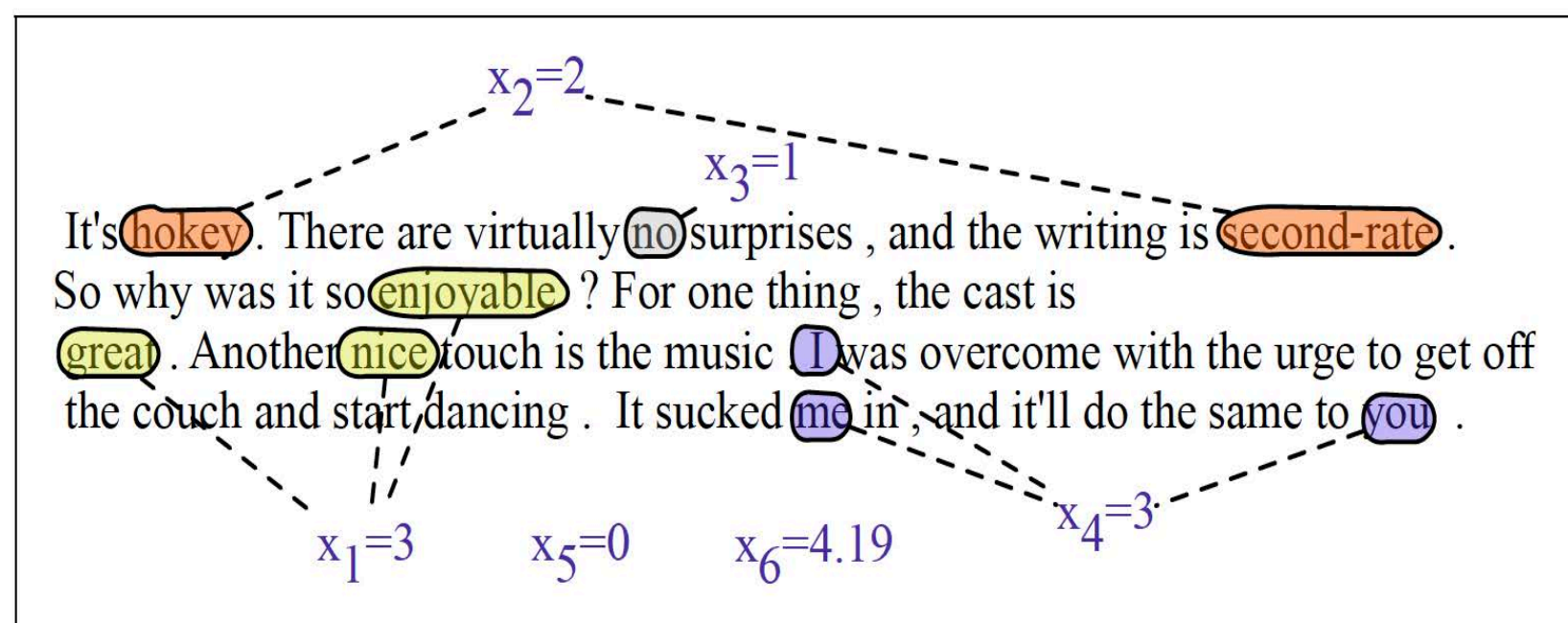
Para una prueba x ,
decimos que sí cuando la
probabilidad $P(y = 1/x)$
es superior a 0.5 , y no en
caso contrario (i.e., límite
de decisión)

$$\hat{y} = \begin{cases} 1, & P(y = 1|x) > 0.5 \\ 0, & \text{caso contrario} \end{cases}$$

Ejemplo de clasificación

Las **empresas** necesitan **comprender** lo que sus **clientes** piensan de sus productos o servicios. Para ello podemos recurrir al **análisis de comentarios de los clientes**. Con el fin de saber si los **comentarios** son (+) o (-).

Representaremos las entradas por 6 características



Var	Definition	Value <input type="text"/>
x_1	count(positive lexicon) \in doc)	3
x_2	count(negative lexicon) \in doc)	2
x_3	$\begin{cases} 1 & \text{if "no" } \in \text{ doc} \\ 0 & \text{otherwise} \end{cases}$	1
x_4	count(1st and 2nd pronouns \in doc)	3
x_5	$\begin{cases} 1 & \text{if "!" } \in \text{ doc} \\ 0 & \text{otherwise} \end{cases}$	0
x_6	log(word count of doc)	$\ln(66) = 4.19$

Ejemplo de clasificación: Pesos

w_1 indica la importancia del número de palabras del **léxico positivo**

w_2 nos indica la importancia de las palabras del **léxico negativo**.

Note que $w_1 = 2.5$ es positivo, mientras que $w_2 = -5.0$, lo que significa que las palabras negativas están asociadas negativamente a una decisión de sentimiento positiva, y son aproximadamente el doble de importantes que las palabras positivas.

$$w_i = [2.5, -5.0, -1.2, 0.5, 2.0, 0.7]$$

$$b = 0.1$$

$$\begin{aligned} p(+|x) = P(Y = 1|x) &= \sigma(w \cdot x + b) \\ &= \sigma([2.5, -5.0, -1.2, 0.5, 2.0, 0.7] \cdot [3, 2, 1, 3, 0, 4.19] + 0.1) \\ &= \sigma(.833) \\ &= 0.70 \\ p(-|x) = P(Y = 0|x) &= 1 - \sigma(w \cdot x + b) \\ &= 0.30 \end{aligned}$$

Ejemplo de
clasificación:
resultados

Dadas las 6 características de la entrada x , se pueden calcular $P(+|x)$ y $P(-|x)$ usando w y b antes definidos



¿Cómo se aprenden
los parámetros del
modelo, las
ponderaciones w y
el sesgo b ?

Aprendizaje en regresión logística

- Queremos **aprender** los **parámetros** que acercan más cada observación de entrenamiento \hat{y} a la verdadera y . Para esto necesitamos dos elementos:
 1. Una **función de distancia** que permita saber cuan cerca \hat{y} esta del verdadero y . Para ello usaremos una función de costo llamada **cross-entropy loss**
 2. Un **algoritmo de optimización** para **actualizar iterativamente** los pesos de forma que se minimice esta función de pérdida. El algoritmo estándar para esto es el **stochastic gradient descent**

La función de pérdida de entropía cruzada

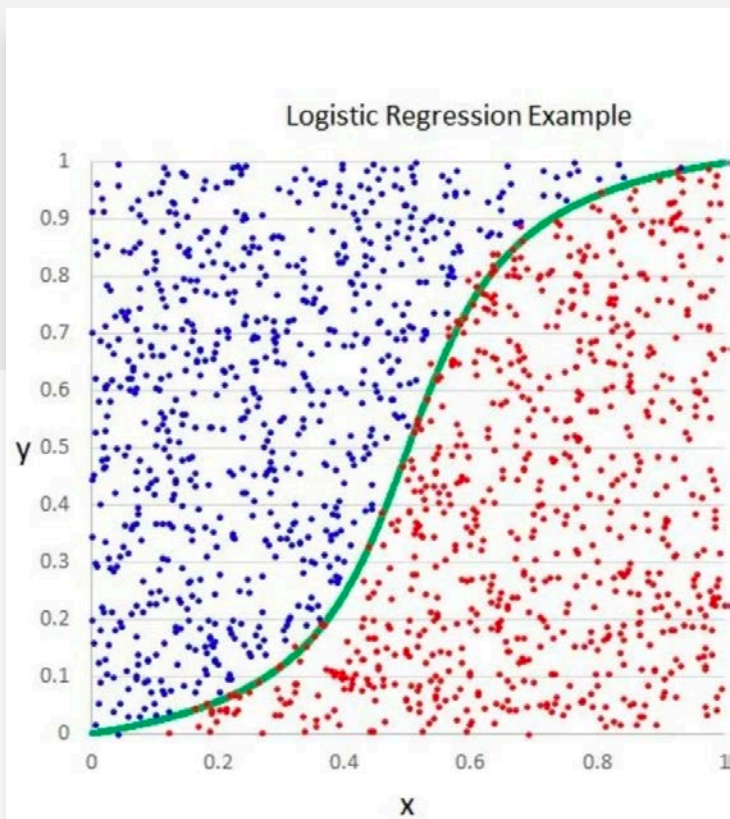
- Debemos **aprender pesos** w y b que **maximicen la probabilidad** de la etiqueta correcta $p(y|x)$.
- Como sólo hay dos **resultados discretos** (1 o 0), se trata de una distribución Bernoulli:

$$p(y|x) = \hat{y}^y (1 - \hat{y})^{1-y}$$

- Para convertir esto en **función de pérdida** a minimizar el resultado es **la pérdida de entropía** cruzada LCE:

$$L_{\text{CE}}(\hat{y}, y) = -\log p(y|x) = -[y \log \hat{y} + (1 - y) \log (1 - \hat{y})]$$

$$L_{\text{CE}}(\hat{y}, y) = -[y \log \sigma(w \cdot x + b) + (1 - y) \log (1 - \sigma(w \cdot x + b))]$$



Retomando el ejemplo de clasificación

- Si la observación es positiva $y=1$ la función de pérdida es:

$$\begin{aligned}
 L_{CE}(\hat{y}, y) &= -[y \log \sigma(w \cdot x + b) + (1 - y) \log (1 - \sigma(w \cdot x + b))] \\
 &= -[\log (1 - \sigma(w \cdot x + b))] \\
 &= -\log (.31) \\
 &= 1.17
 \end{aligned}$$

- Si la observación es negativa $y=0$ la función de pérdida es:

$$\begin{aligned}
 L_{CE}(\hat{y}, y) &= -[y \log \sigma(w \cdot x + b) + (1 - y) \log (1 - \sigma(w \cdot x + b))] \\
 &= -[\log \sigma(w \cdot x + b)] \\
 &= -\log (.69) \\
 &= .37
 \end{aligned}$$

¿Por qué minimizar esta probabilidad logarítmica negativa hace lo que queremos?

- Un clasificador perfecto asignaría la probabilidad 1 al resultado correcto y la probabilidad 0 al resultado incorrecto.
- Cuanto \hat{y} más cerca esté de 1, mejor será el clasificador; cuanto \hat{y} más cerca de 0, peor será el clasificador.
- El logaritmo negativo de esta probabilidad es una métrica de pérdida que va de 0 (ninguna pérdida) a infinito (pérdida infinita).
- Esta función de pérdida también garantiza que, a medida que se maximiza la probabilidad de la respuesta correcta, se minimiza la probabilidad de la respuesta incorrecta, ya que ambas suman uno,

Gradient Descent – gradiente descendiente

- El objetivo de la gradiente descendiente es de encontrar los pesos óptimos para minimizar la función de pérdida definida para el modelo.
- Por lo tanto, el objetivo es encontrar el conjunto de pesos que minimice la función de pérdida $\theta = w, b$, promediando todos los ejemplos:

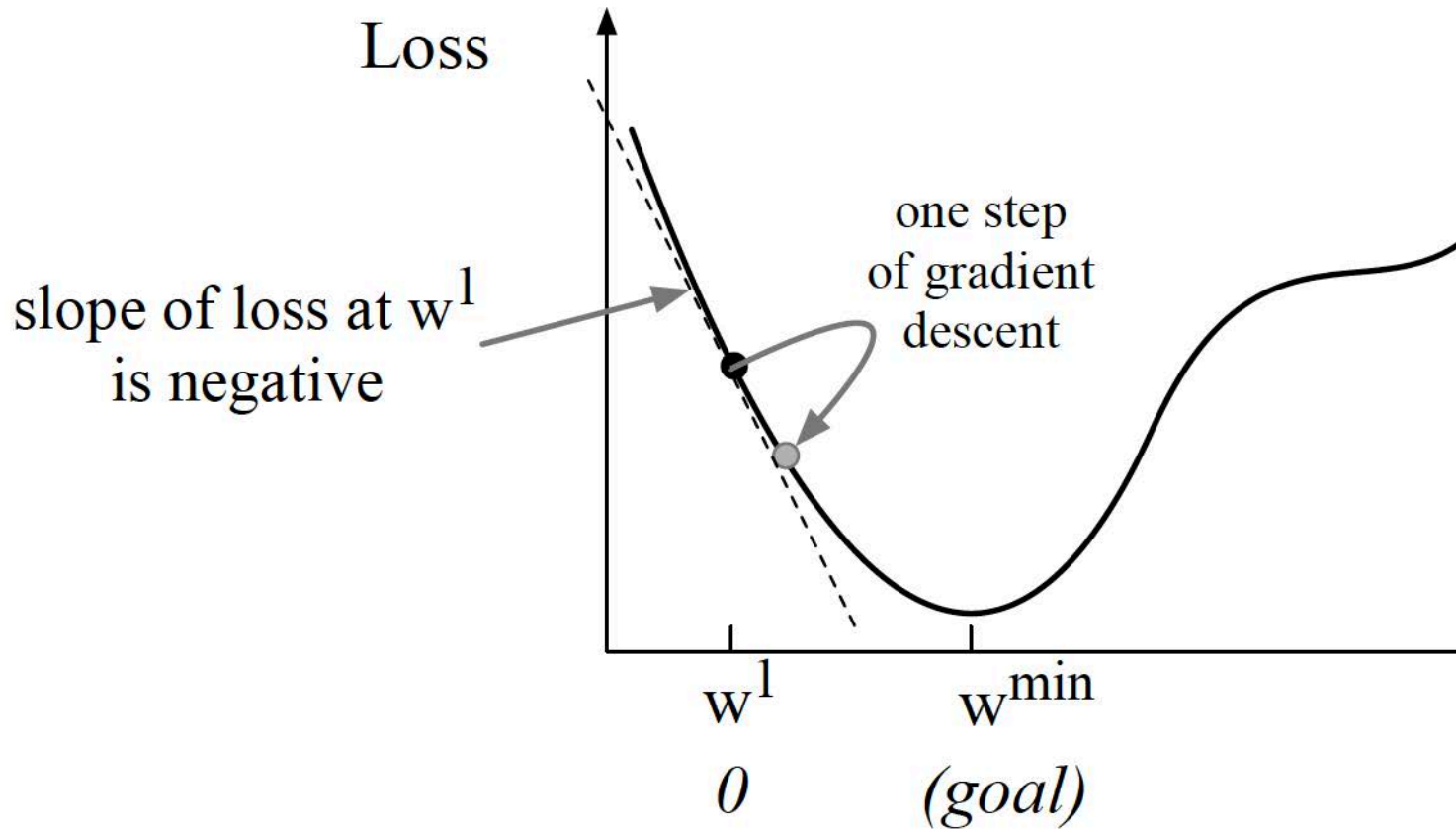
$$\hat{\theta} = \operatorname{argmin}_{\theta} \frac{1}{m} \sum_{i=1}^m L_{\text{CE}}(\overset{\hat{y}}{\boxed{f(x^{(i)}; \theta)}}, y^{(i)})$$

Gradient Descent : Intuición

La **intuición** detrás de la **gradiente descendente** es que si estás caminando en un cañón y tratas de **descender lo más rápido posible** hasta el río en el fondo, podrías mirar a tu alrededor 360 grados, **encontrar la dirección** en la que el suelo está más inclinado, y caminar cuesta abajo en esa dirección.



Gradient Descent



- Para la regresión logística, esta **función de pérdida es convexa**
- Una función convexa **sólo tiene un mínimo** (no hay mínimos locales)
- La **magnitud** de la cantidad a **mover** en el **descenso** de gradiente es el valor de la pendiente ponderado por una **tasa de aprendizaje η**

$$w^{t+1} = w^t - \eta \frac{d}{dw} f(x; w)$$

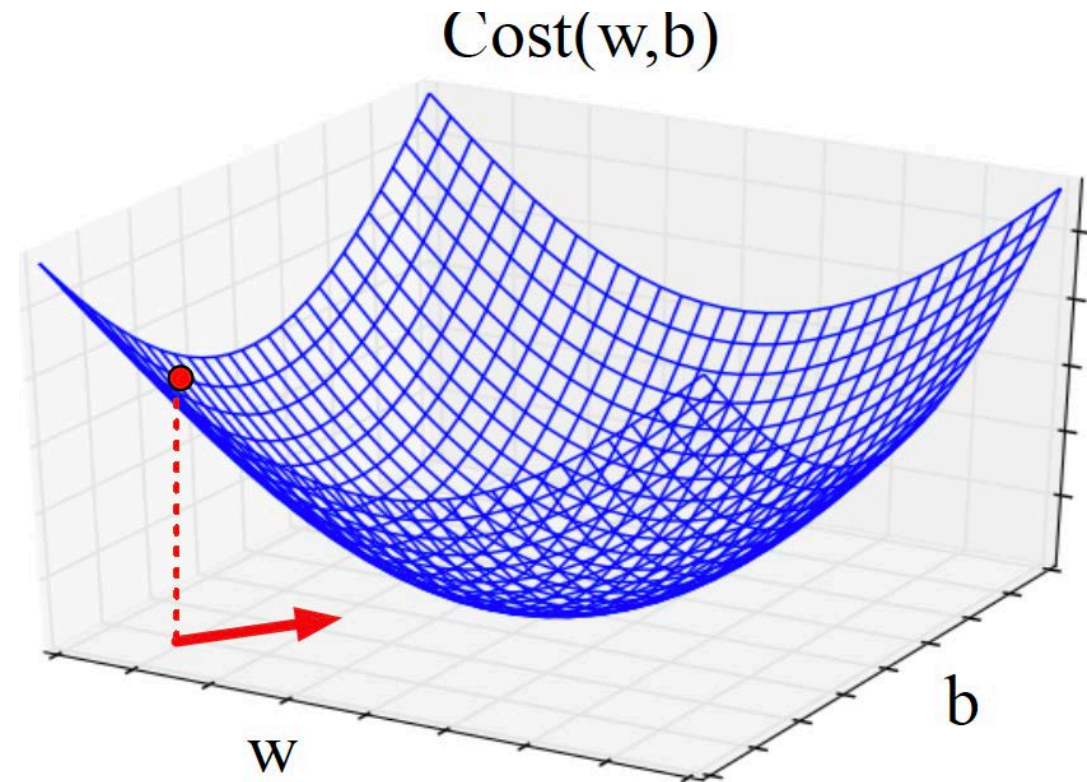
Learning rate

Magnitud y dirección

Gradient Descent en espacio N-dimensional

- El **gradiente** es un **vector** que expresa los **componentes direccionales** de la pendiente más pronunciada a **lo largo de cada una de esas N dimensiones**.

En 2 dimensiones de peso w y tasa de aprendizaje b , el gradiente es un vector con dos componentes ortogonales, cada uno de los cuales nos dice cuánto se inclina la pendiente en la dimensión w y b .



Gradient Descent en la regresión logística

- Para actualizar θ , necesitamos una definición de gradiente $\nabla L(f(x, \theta), y)$. Recordemos la función de pérdida de entropía cruzada:
- Resulta que la derivada de esta función para un vector de observación x :

Algoritmo de gradiente decendiente estocastica

Algorithm 1 Stochastic Gradient Descent Algorithm returns θ

```
1: function STOCHASTIC GRADIENT DESCENT ALGORITHM( $L(), f(), \mathbf{x}, \mathbf{y}$ )
2:   #  $L$  : Función de perdida
3:   #  $f$  : Función parametrizada de  $\theta$ 
4:   #  $\mathbf{x}$  : Conjunto de entrada  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(m)}$ 
5:   #  $\mathbf{y}$  : Conjunto de salida  $\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \dots, \mathbf{y}^{(m)}$ 
6:    $\theta = 0$ 
7:   while  $\epsilon > \theta^{t+1} - \theta^t$  do
8:     for cada tupla de entrenamiento  $(\mathbf{x}^i, \mathbf{y}^i)$  do
9:        $\hat{\mathbf{y}} = f(\mathbf{x}^i, \theta)$  ▷ Estimar  $\hat{\mathbf{y}}$ 
10:       $L(\hat{\mathbf{y}}^i, \mathbf{y}^i)$  ▷ Cuan lejos  $\hat{\mathbf{y}}^i$  esta de  $\mathbf{y}^i$ 
11:       $\mathbf{g} = \nabla_{\theta} L(f(\mathbf{x}^i, \theta), \mathbf{y}^i)$  ▷ ¿Cómo mover  $\theta$  para maximizar las pérdidas?
12:       $\theta = \theta - \eta \cdot \mathbf{g}$  ▷ Actualizar  $\theta$ 
13:   Regresa  $\theta$ 
```

Retomando el ejemplo de clasificación simplificado

Utilizaremos una versión **simplificada** del ejemplo de clasificación de sentimientos, ya que contempla una **única observación x** , cuyo valor correcto es **$y = 1$** , y sólo dos características:

- $x_1=3$ (conteo de palabras **positivas**)
- $x_2=2$ (conteo de palabras **negativas**)

Para la primera iteración:

$$\begin{aligned}w_1 = w_2 = b &= 0 \\ \eta &= 0.1\end{aligned}$$

Retomando el ejemplo de clasificación simplificado

El único paso de actualización requiere que calculemos el gradiente, multiplicado por la tasa de aprendizaje


$$\theta^{t+1} = \theta^t - \eta \nabla_{\theta} L(f(x^{(i)}; \theta), y^{(i)})$$

En nuestro mini ejemplo hay tres parámetros (w_1 , w_2 y b). Podemos calcular el primer gradiente de la siguiente manera:

$$\nabla_{w,b} = \begin{bmatrix} \frac{\partial L_{\text{CE}}(\hat{y}, y)}{\partial w_1} \\ \frac{\partial L_{\text{CE}}(\hat{y}, y)}{\partial w_2} \\ \frac{\partial L_{\text{CE}}(\hat{y}, y)}{\partial b} \end{bmatrix} = \begin{bmatrix} (\sigma(w \cdot x + b) - y)x_1 \\ (\sigma(w \cdot x + b) - y)x_2 \\ \sigma(w \cdot x + b) - y \end{bmatrix} = \begin{bmatrix} (\sigma(0) - 1)x_1 \\ (\sigma(0) - 1)x_2 \\ \sigma(0) - 1 \end{bmatrix} = \begin{bmatrix} -0.5x_1 \\ -0.5x_2 \\ -0.5 \end{bmatrix} = \begin{bmatrix} -1.5 \\ -1.0 \\ -0.5 \end{bmatrix}$$

Retomando el ejemplo de clasificación simplificado

Ahora que tenemos un gradiente, calculamos el nuevo vector de parámetros θ^1 moviendo θ^0 :

$$\theta^1 = \begin{bmatrix} w_1 \\ w_2 \\ b \end{bmatrix} - \eta \begin{bmatrix} -1.5 \\ -1.0 \\ -0.5 \end{bmatrix} = \begin{bmatrix} .15 \\ .1 \\ .05 \end{bmatrix}$$


Después de un paso de descenso de gradiente, los pesos se han desplazado para ser: $w_1 = 0.15$, $w_2 = 0.1$, y $b = 0.05$.

Batch training

Al ver tantos ejemplos, el entrenamiento por lotes ofrece una estimación excelente de la dirección en la que hay que mover los pesos:

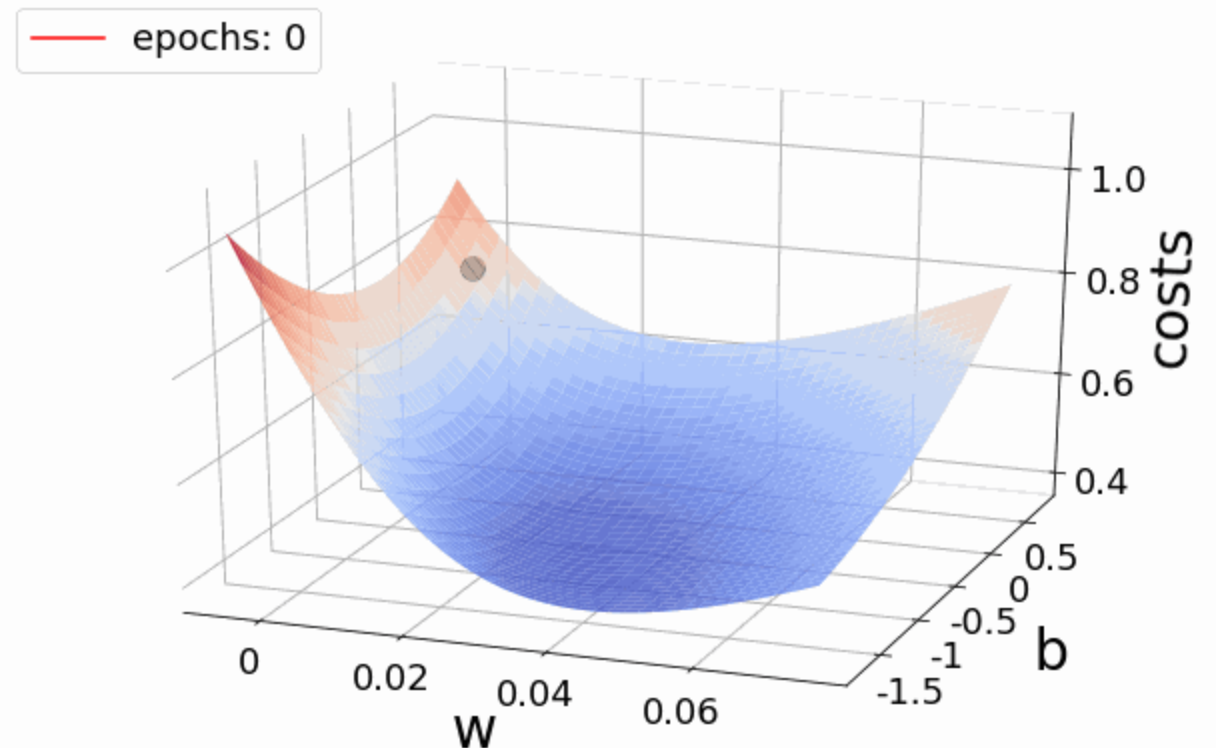
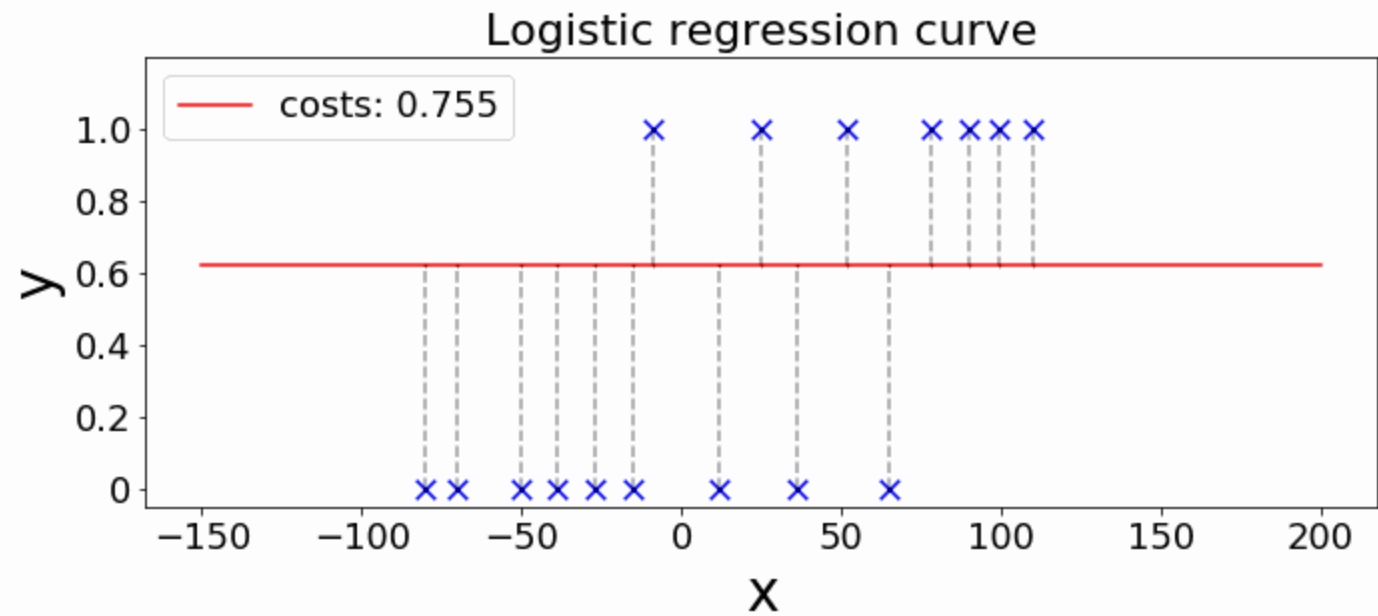
$$\begin{aligned} \text{Cost}(\hat{y}, y) &= \frac{1}{m} \sum_{i=1}^m L_{\text{CE}}(\hat{y}^{(i)}, y^{(i)}) \\ &= -\frac{1}{m} \sum_{i=1}^m y^{(i)} \log \sigma(w \cdot x^{(i)} + b) + (1 - y^{(i)}) \log (1 - \sigma(w \cdot x^{(i)} + b)) \end{aligned}$$

El gradiente del lote es la media de los gradientes individuales:

$$\frac{\partial \text{Cost}(\hat{y}, y)}{\partial w_j} = \frac{1}{m} \sum_{i=1}^m \left[\sigma(w \cdot x^{(i)} + b) - y^{(i)} \right] x_j^{(i)}$$

Animación

- Podemos apreciar como la curva L de perdida se va acomodando a medida que la gradiente hala su mínimo global



Takeaways

- La regresión logística es una técnica simple y eficaz
- No es necesario disponer de grandes recursos computacionales
- Los resultados son interpretables
- Los pesos w_i de cada una de las características determinan la importancia de esta en la decisión final

Gracias

¿Preguntas?