Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

TEMA 3: REGULARIZACIÓN

Mikel Galar Idoate
mikel.galar@unavarra.es

Ciencia de datos con técnicas inteligentes
Experto Universitario en Ciencia de Datos y Big Data

Índice

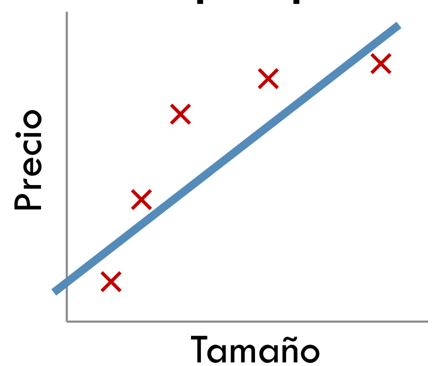
1. Regularización
   - El problema de sobre-aprendizaje
   - Modificación de la función de coste
     - Regularización en regresión lineal
     - Regularización en regresión logística

Índice

1. Regularización
   - El problema de sobre-aprendizaje
   - Modificación de la función de coste
     - Regularización en regresión lineal
     - Regularización en regresión logística

Regularización

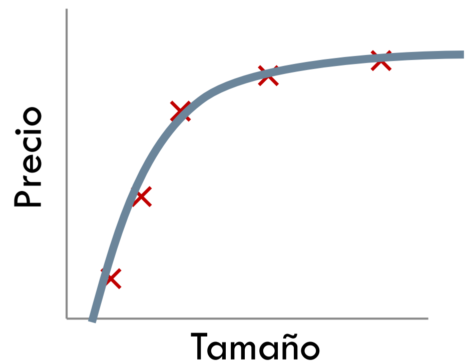
- **El problema de sobre-aprendizaje**
  - Ejemplo: Regresión lineal



$\theta_0 + \theta_1 x$

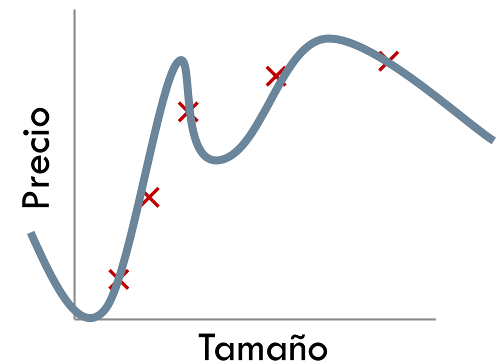
Bias alto

(no se ajusta)



$\theta_0 + \theta_1 x + \theta_2 x^2$

Buen ajuste



$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

Varianza alta

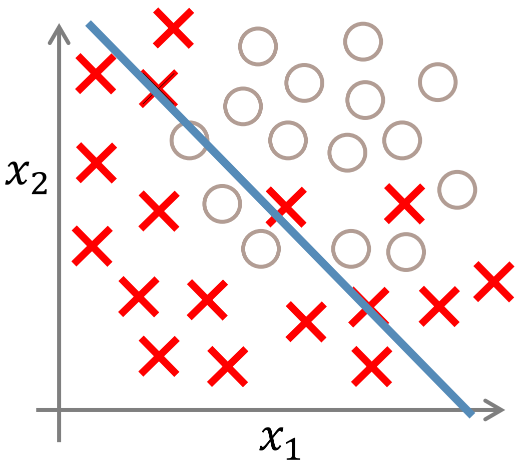
(sobre-aprendizaje)

Problema: Sobre-aprendizaje

Si tenemos **muchas características,** la hipótesis puede modelar los **datos de entrenamiento casi perfectamente** $\left(J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 \approx 0\right)$, pero el modelo **falla al generalizar** a nuevos ejemplos (predecir el precio de otras casas)

Regularización

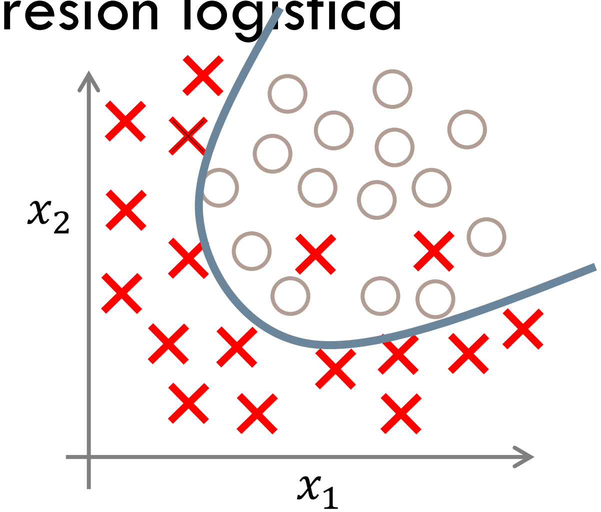
- **El problema de sobre-aprendizaje**
  - Ejemplo: Regresión logística



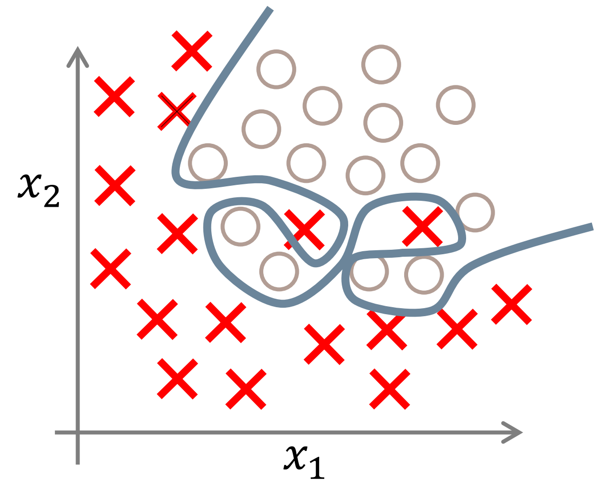
$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

$$g(z) = \frac{1}{1 + e^{-z}}$$

No se ajusta



$$h_\theta(x) = g\big(\theta_0 + \theta_1 x_1 + \theta_2 x_2 \\ + \theta_3 x_1^2 + \theta_4 x_2^2 \\ + \theta_5 x_1 x_2\big)$$



$$h_\theta(x) = g\big(\theta_0 + \theta_1 x_1 + \theta_2 x_2 \\ + \theta_3 x_1^2 + \theta_4 x_2^2 \\ + \theta_5 x_1 x_2 + \theta_6 x_1^3 x_2 + \cdots\big)$$

Sobre-aprendizaje

Regularización

- **El problema de sobre-aprendizaje**
  - Pero no siempre podemos dibujar la gráfica
    - Si tenemos más de 2 características
      - Ejemplo
        - Tamaño de la casa
        - Nº habitaciones
        - Nº pisos
        - Antigüedad
        - Tamaño de la cocina
        - …
    - Eliminar características puede ser una opción
      - Pero todas podrían aportar algo
    - Si tenemos muchas características pero pocos ejemplos…
      - Sobre-entrenamiento

Regularización

- **El problema de sobre-aprendizaje**
  - **Posibles soluciones**
    1. Reducir el número de características
       - Seleccionarlas manualmente
       - Algoritmos de selección de modelos (siguiente tema)
    2. **Regularización**
       - Utilizamos todas las características
         - Pero **reducimos la magnitud de los parámetros** $\boldsymbol{\theta_j}$
       - Funciona cuando tenemos muchas características y todas contribuyen a la predicción

Índice

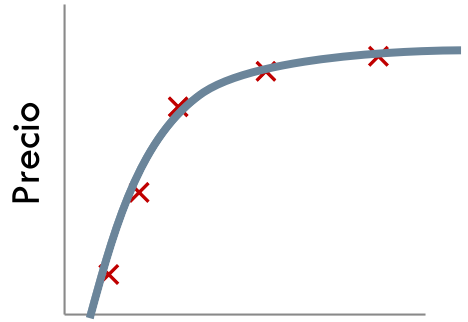
1. Regularización
   - El problema de sobre-aprendizaje
   - Modificación de la función de coste
     - Regularización en regresión lineal
     - Regularización en regresión logística

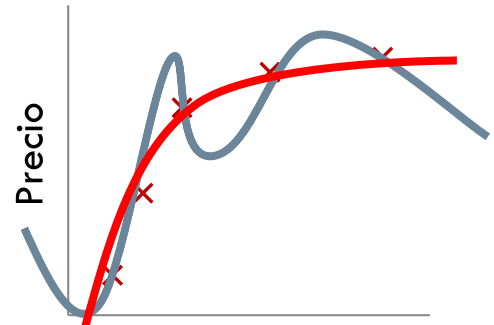
Regularización

- **Modificación de la función de coste**
  - Idea
    - Si reducimos la magnitud de los parámetros θ_j
      - Reducimos la flexibilidad del modelo
      - Reducimos las probabilidad de sobre-aprendizaje

Regularización

- **Modificación de la función de coste**
  - Idea





$$\theta_0 + \theta_1 x + \theta_2 x^2 \qquad \theta_0 + \theta_1 x + \theta_2 x^2 + \cancel{\theta_3 x^3} + \cancel{\theta_4 x^4}$$

  - Supongamos que penalizamos θ_3, θ_4 haciendo que sean muy pequeños

$$\min_{\theta} \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + 1000 \cdot \theta_3^2 + 1000 \cdot \theta_4^2 \implies \theta_3 \approx 0, \theta_4 \approx 0$$

Regularización

- **Modificación de la función de coste**
  - Idea
    - Valores bajos para los parámetros $\theta_1, \theta_2 \dots, \theta_n$
      - Hipótesis más simples
      - Menor tendencia al sobre-aprendizaje
    - Nueva función de coste

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \boxed{\lambda \sum_{j=1}^{n} \theta_j^2}\right]$$

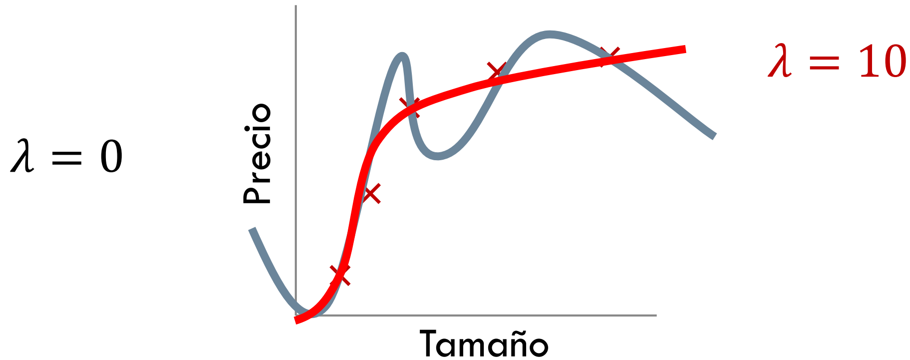
    - Minimizamos también los valores de los parámetros
      - En base a un parámetro de regularización λ
        - Que controla el balance entre la complejidad y el error

Regularización

Modificación de la función de coste

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \lambda \sum_{j=1}^{n} \theta_j^2\right]$$

- Minimizamos también los valores de los parámetros



$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

- No es una función cuadrática
  - Pero las curvas son más suaves y hay menor complejidad

Regularización

- **Modificación de la función de coste**
  - **¡Cuidado¡**

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \lambda\sum_{j=1}^{n}\theta_j^2\right]$$

  - ¿Qué ocurre si λ toma un valor muy alto ($\lambda = 10^{10}$)?
    - El algoritmo funciona correctamente
      - Da igual lo grande que sea λ
    - El algoritmo no elimina el sobre-aprendizaje
    - El algoritmo no se ajusta
      - Ni si quiera a los datos de entrenamiento
    - El descenso por gradiente no converge

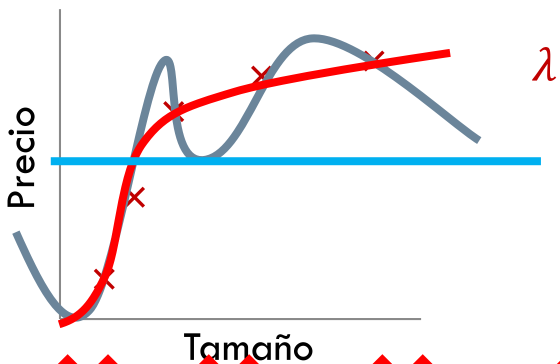
Regularización

- **Modificación de la función de coste**

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \lambda\sum_{j=1}^{n}\theta_j^2\right]$$

$\lambda = 10^{10}$

$\lambda = 0$



$\lambda = 10$

$$\theta_0 + \cancel{\theta_1 x} + \cancel{\theta_2 x^2} + \cancel{\theta_3 x^3} + \cancel{\theta_4 x^4} \Rightarrow h_\theta(x) = \theta_0$$

Índice

1. Regularización
   - El problema de sobre-aprendizaje
   - Modificación de la función de coste
     - Regularización en regresión lineal
     - Regularización en regresión logística

Regularización

- **Regularización en regresión lineal**
  - Función de coste a minimizar

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \lambda \sum_{j=1}^{n} \theta_j^2\right]$$

$$\min_\theta J(\theta)$$

  - Para el **descenso por gradiente**
    - Tenemos que volver a calcular $\frac{\partial J(\theta)}{\partial \theta_j}$
  - Igualmente para la solución directa

Regularización

- **Solución directa**

$$\theta = \left(X^T X + \lambda \begin{bmatrix} 0 & \cdots & 0 & 0 & 0 \\ \vdots & 1 & & 0 & 0 \\ 0 & & 1 & & 0 \\ 0 & 0 & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \right)^{-1} X^T y$$

- Con $\lambda > 0$ tenemos que **la matriz es invertible**

Regularización

- **Descenso por gradiente**
  - Asignar a $\theta = \{\theta_0, \dots, \theta_n\}$ valores aleatorios
  - Repetir hasta convergencia
    - (nº iteraciones o |error – error_anterior| < umbral)

$$\boxed{\theta_j := \theta_j - \alpha \frac{\partial J(\theta)}{\partial \theta_j} \quad \text{para } j = 0, \dots, n}$$

ACTUALIZAR TODOS LOS θ_j SIMULTÁNEAMENTE

En regresión lineal con múltiples variables y regularización …

$$h_\theta(x) = \theta^T x$$

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \lambda \sum_{j=1}^{n} \theta_j^2\right]$$

Regularización

- **Descenso por gradiente (anterior)**
  - Asignar a θ valores aleatorios (o a ceros)
  - Repetir hasta convergencia
    - (nº iteraciones o |error – error_anterior| < umbral)

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) \cdot x_j^{(i)} \quad \textbf{para todo } \boldsymbol{j} = \mathbf{0}, \dots, \boldsymbol{n}$$

$\Downarrow$

ACTUALIZAR TODOS LOS θ_j SIMULTÁNEAMENTE

Regularización

- **Descenso por gradiente**
  - Asignar a $\theta = \{\theta_0, \dots, \theta_n\}$ valores aleatorios
  - Repetir hasta convergencia
    - (nº iteraciones o |error – error_anterior| < umbral)

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} \frac{1}{2m} \left[\sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^{n} \theta_j^2 \right] \text{ para } j = 0, \dots, n$$

Regularización

- **Descenso por gradiente**
  - Asignar a $\theta = \{\theta_0, \dots, \theta_n\}$ valores aleatorios
  - Repetir hasta convergencia
    - (nº iteraciones o |error – error_anterior| < umbral)

$$\theta_0 := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} \frac{1}{2m} \left[\sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^{n} \theta_j^2 \right] \text{para } j = 0$$

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} \frac{1}{2m} \left[\sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda \sum_{j=1}^{n} \theta_j^2 \right] \text{para } j = 1, \dots, n$$

θ_0 no se regulariza

Regularización

- **Descenso por gradiente**
  - Asignar a $\theta = \{\theta_0, \dots, \theta_n\}$ valores aleatorios
  - Repetir hasta convergencia
    - (n° iteraciones o |error – error_anterior| < umbral)

0 por no estar θ_0

$$\theta_0 := \theta_0 - \alpha \frac{1}{2m}\left[\boxed{\frac{\partial}{\partial\theta_0}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2} + \boxed{\frac{\partial}{\partial\theta_0}\lambda\sum_{j=1}^{n}\theta_j^2}\right] \quad \text{para } j = 0$$

$$\theta_j := \theta_j - \alpha \frac{1}{2m}\left[\boxed{\frac{\partial}{\partial\theta_j}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2} + \frac{\partial}{\partial\theta_j}\lambda\sum_{j=1}^{n}\theta_j^2\right] \quad \text{para } j = 1, \dots, n$$

Ya lo conocemos

$$2\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)\cdot x_j^{(i)}$$

Regularización

- **Descenso por gradiente**
  - Asignar a $\theta = \{\theta_0, \dots, \theta_n\}$ valores aleatorios
  - Repetir hasta convergencia
    - (nº iteraciones o |error – error_anterior| < umbral)

$$\theta_0 := \theta_0 - \alpha \frac{1}{2m}\left[2\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)\cdot x_j^{(i)}\right] \quad \text{para } j = 0$$

$$\theta_j := \theta_j - \alpha \frac{1}{2m}\left[2\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)\cdot x_j^{(i)} \boxed{+ \frac{\partial}{\partial \theta_j}\lambda \sum_{j=1}^{n}\theta_j^2}\right] \quad \text{para } j = 1, \dots, n$$

$$\frac{2\lambda}{m}\cdot\theta_j$$

Regularización

- **Descenso por gradiente**
  - Asignar a $\theta = \{\theta_0, \dots, \theta_n\}$ valores aleatorios
  - Repetir hasta convergencia
    - (n° iteraciones o $|$error − error_anterior$|$ < umbral)

$$\theta_0 \coloneqq \theta_0 - \alpha \frac{1}{2m}\left[2\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)\cdot x_j^{(i)}\right] \qquad \text{para } j = 0$$

$$\theta_j \coloneqq \theta_j - \alpha \frac{1}{2m}\left[2\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)\cdot x_j^{(i)} + \frac{2\lambda}{m}\cdot\theta_j\right] \qquad \text{para } j = 1, \dots, n$$

Regularización

- **Descenso por gradiente**
  - Asignar a $\theta = \{\theta_0, \dots, \theta_n\}$ valores aleatorios
  - Repetir hasta convergencia
    - (nº iteraciones o |error – error_anterior| < umbral)

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) \cdot x_j^{(i)} \qquad \text{para } j = 0$$

$$\theta_j := \theta_j - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) \cdot x_j^{(i)} + \frac{\lambda}{m} \cdot \theta_j\right] \qquad \text{para } j = 1, \dots, n$$

ACTUALIZAR TODOS LOS θ_j SIMULTÁNEAMENTE

Regularización

- **Descenso por gradiente (reescrito)**
  - Asignar a $\theta = \{\theta_0, \dots, \theta_n\}$ valores aleatorios
  - Repetir hasta convergencia
    - (nº iteraciones o |error – error_anterior| < umbral)

$$\theta_0 \coloneqq \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) \cdot x_j^{(i)} \qquad \text{para } j = 0$$

$$\theta_j \coloneqq \theta_j \left(1 - \boxed{\alpha \frac{\lambda}{m}}\right) - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) \cdot x_j^{(i)} \qquad \text{para } j = 1, \dots, n$$

Suele tomar un valor pequeño y hace que el valor de θ_j decrezca

Índice

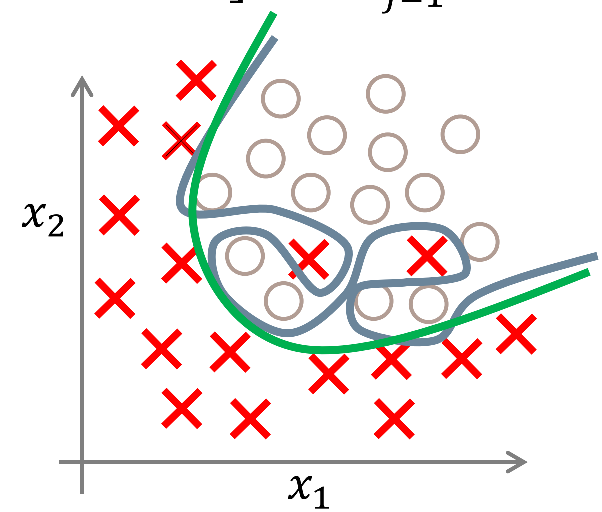
1. Regularización
   - El problema de sobre-aprendizaje
   - Modificación de la función de coste
     - Regularización en regresión lineal
     - Regularización en regresión logística

Regularización

- **Regularización en regresión logística**
  - Función de coste a minimizar

$$J(\theta) = -\left[\frac{1}{m}\sum_{i=1}^{m} y^{(i)} \log\left(h_\theta\left(x^{(i)}\right)\right) + \left(1 - y^{(i)}\right)\log\left(1 - h_\theta\left(x^{(i)}\right)\right)\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

$$\min_\theta J(\theta)$$



$$h_\theta(x) = g\big(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2 + \theta_6 x_1^3 x_2 + \cdots\big)$$

  - Para el **descenso por gradiente**
    - Tenemos que volver a calcular $\frac{\partial J(\theta)}{\partial \theta_j}$

Regularización

- **Descenso por gradiente en regresión logística**
  - Asignar a θ valores aleatorios (o a ceros)
  - Repetir hasta convergencia
    - (nº iteraciones o |error – error_anterior| < umbral)

$$\boxed{\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)} \right) \cdot x_j^{(i)} \quad \textbf{para todo } \boldsymbol{j} = \mathbf{0}, \dots, \boldsymbol{n}}$$

ACTUALIZAR TODOS LOS θ_j SIMULTÁNEAMENTE

Regularización

- **Descenso por gradiente**
  - Asignar a $\theta = \{\theta_0, \dots, \theta_n\}$ valores aleatorios
  - Repetir hasta convergencia
    - (n° iteraciones o |error – error_anterior| < umbral)

$$\theta_j := \theta_j - \alpha \left[\frac{\partial}{\partial \theta_j} \boxed{-\left[\frac{1}{m}\sum_{i=1}^{m} y^{(i)} \log\left(h_\theta(x^{(i)})\right) + (1 - y^{(i)}) \log\left(1 - h_\theta(x^{(i)})\right)\right]} + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2 \right]$$

La derivada sigue igual $\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) \cdot x_j^{(i)}$ para $j = 0, \dots, n$

Regularización

- **Descenso por gradiente**
  - Asignar a $\theta = \{\theta_0, \dots, \theta_n\}$ valores aleatorios
  - Repetir hasta convergencia
    - (n° iteraciones o |error – error_anterior| < umbral)

$$\theta_j := \theta_j - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)} \right) \cdot x_j^{(i)} + \frac{\partial}{\partial \theta_j} \frac{\lambda}{2m} \sum_{j=1}^{n} \theta_j^2 \right] \quad \text{para } j = 0, \dots, n$$

Regularización

- **Descenso por gradiente**
  - Asignar a $\theta = \{\theta_0, \dots, \theta_n\}$ valores aleatorios
  - Repetir hasta convergencia
    - (nº iteraciones o |error – error_anterior| < umbral)

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) \cdot x_j^{(i)} \qquad \text{para } j = 0$$

$$\theta_j := \theta_j - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) \cdot x_j^{(i)} + \boxed{\frac{\partial}{\partial \theta_j} \frac{\lambda}{2m} \sum_{j=1}^{n} \theta_j^2} \right] \qquad \text{para } j = 1, \dots, n$$

$$\frac{2\lambda}{2m}\theta_j = \frac{\lambda}{m}\theta_j$$

θ_0 no se regulariza

Regularización

- **Descenso por gradiente**
  - Asignar a $\theta = \{\theta_0, \ldots, \theta_n\}$ valores aleatorios
  - Repetir hasta convergencia
    - (nº iteraciones o $|$error – error_anterior$|$ < umbral)

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)} \right) \cdot x_j^{(i)}$$ para $j = 0, \ldots, n$

$$\theta_j := \theta_j - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)} \right) \cdot x_j^{(i)} + \frac{\lambda}{m} \theta_j \right]$$ para $j = 1, \ldots, n$

ACTUALIZAR TODOS LOS θ_j SIMULTÁNEAMENTE