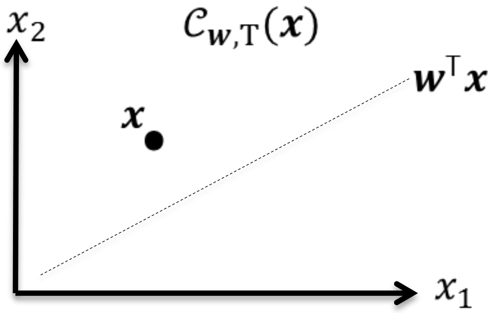
CLASIFICACIÓN DE OBJETOS

Regresión Logística – Aprendizaje (I)

Antonio M. López

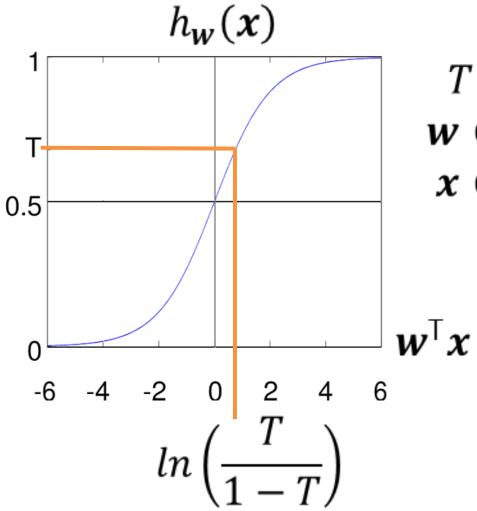
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

Universitat Autònoma de Barcelona



$$\mathcal{C}_{\boldsymbol{w},\mathrm{T}}(\boldsymbol{x}) = \begin{cases} 1 & si\ h_{\boldsymbol{w}}(\boldsymbol{x}) < T \sim \boldsymbol{w}^{\top}\boldsymbol{x} < ln\left(\frac{T}{1-T}\right) \\ 0 & en\ otro\ caso \end{cases}$$

Donde $h_{\boldsymbol{w}}(\boldsymbol{x}) = Logistic\,(\boldsymbol{w}^{\top}\boldsymbol{x}\,) = \frac{1}{1+e^{-\boldsymbol{w}^{\top}\boldsymbol{x}}}$



$T \in [0,1]$

$\boldsymbol{w} \in \mathbb{R}^{n+1}$

$\boldsymbol{x} \in \mathbb{R}^{n+1}$

$x_0 = 1$

- ¿Qué $\boldsymbol{w}$ usar? → Aprendizaje automático

$\boldsymbol{w}$ se aprende a partir de un conjunto de muestras

Conjunto de entrenamiento: $\mathcal{S} = \{(\boldsymbol{x'}^1, y^1), \ldots, (\boldsymbol{x'}^M, y^M)\}$

$(\boldsymbol{x'}^j, y^j)$ j-ésima muestra de entrenamiento

$\boldsymbol{x'}^j \in \mathbb{R}^n \rightarrow$ Lo transformamos en $\boldsymbol{x}^j \in \mathbb{R}^{n+1}$, donde $\boldsymbol{x}^j = (1, \ \boldsymbol{x'}^j)$

$y^j \in \{0, 1\}$ <u>clasificación binaria</u> y <u>supervisada</u>

$\mathcal{S} \rightarrow$ Regresión logística $\rightarrow \boldsymbol{w}$ (El que minimiza el "coste de equivocarse" al clasificar $\{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^M\}$)

- El coste de equivocarse al clasificar $\boldsymbol{x}^j$ se puede definir como $\left(y^j - h_{\boldsymbol{w}}(\boldsymbol{x}^j)\right)^2$, es decir, utilizando la denominada pérdida cuadrática (*squared loss*, L_2).

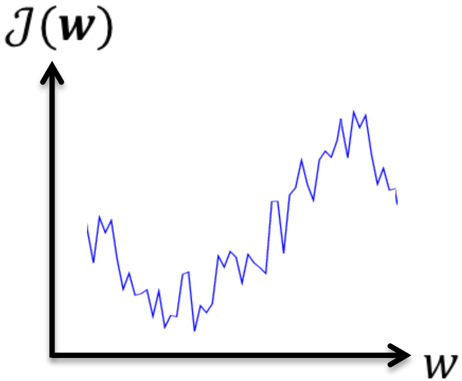
- El vector $\boldsymbol{w}^*$ que minimiza el coste de equivocarse para $\mathcal{S}$ en su totalidad se podría obtener resolviendo:

$$\boldsymbol{w}^* \longleftarrow \underset{\boldsymbol{w}\,\in\,\mathbb{R}^{n+1}}{argmin} \underbrace{\boxed{\frac{1}{M}\sum_{j=1}^{M}\left(y^j - h_{\boldsymbol{w}}(\boldsymbol{x}^j)\right)^2}}_{\mathcal{J}(\boldsymbol{w})}$$

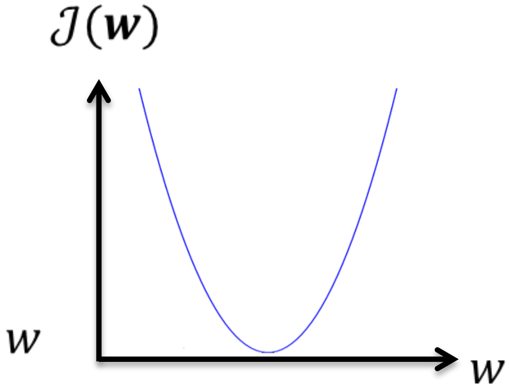
Universitat Autònoma de Barcelona

Comentarios:

- No hay una solución cerrada fácil para obtener $\boldsymbol{w}^*$, es decir, no podemos "despejar" $\boldsymbol{w}$ de una manera sencilla a partir de $\mathcal{J}(\boldsymbol{w})$.
- Por tanto, se opta por utilizar un método de optimización que a partir de un valor inicial de $\boldsymbol{w}$ y unas ecuaciones de "actualización", nos lleve a la solución que buscamos.
- Sin embargo, $\mathcal{J}(\boldsymbol{w})$ no es una función convexa respecto a $\boldsymbol{w}$. Por lo tanto, es difícil de optimizar (en este caso minimizar).



no convexa



convexa

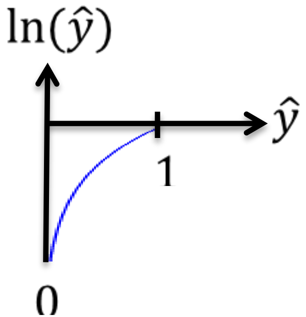
Universitat Autònoma de Barcelona

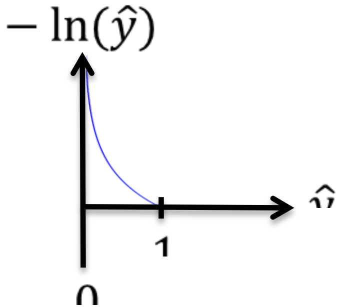
- Por ese motivo, se utiliza la siguiente alternativa:

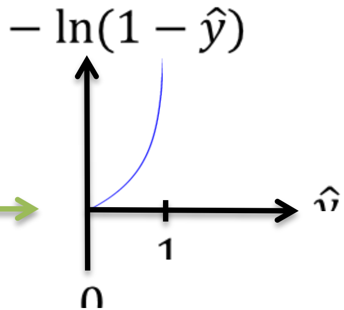
$$\left(y^j - h_{\boldsymbol{w}}(\boldsymbol{x}^j)\right)^2 \longrightarrow coste\left(h_{\boldsymbol{w}}(\boldsymbol{x}^j), y^j\right)$$

$$coste(\hat{y}, y) = \begin{cases} -ln(\hat{y}) & si\ y = 1 \\ -ln(1-\hat{y}) & si\ y = 0 \end{cases}$$

$$\hat{y} \in [0,1]$$







$$\hat{y} \to 1 \Rightarrow coste \to 0$$
$$\hat{y} \to 0 \Rightarrow coste \to \infty$$

$$\hat{y} \to 0 \Rightarrow coste \to 0$$
$$\hat{y} \to 1 \Rightarrow coste \to \infty$$

UAB Universitat Autònoma de Barcelona

- $coste(\hat{y}, y) = \begin{cases} -\ln(\hat{y}) & si\ y = 1 \\ -\ln(1-\hat{y}) & si\ y = 0 \end{cases}$

compactado

$$-(\,y \ln(\hat{y}) + (1-y)\ln(1-\hat{y})\,)$$

$\mathcal{J}(\boldsymbol{w})$ es convexa respecto $\boldsymbol{w}$

$$\boxed{\begin{aligned} &\mathcal{J}(\boldsymbol{w}) = -\frac{1}{M}\sum_{j=1}^{M} y^j \ln(h_{\boldsymbol{w}}(\boldsymbol{x}^j)) + (1-y^j)\ln\left(1-h_{\boldsymbol{w}}(\boldsymbol{x}^j)\right) \\ &\boldsymbol{w}^* \leftarrow \underset{\boldsymbol{w}\in\mathbb{R}^{n+1}}{argmin}\, \mathcal{J}(\boldsymbol{w}) \end{aligned}}$$

Regresión logística

- Conceptos clave de este vídeo:
    - Conjunto de entrenamiento
    - Aprendizaje automático supervisado
    - Regresión logística
    - Función de coste