Regresión Logística y Clasificación

Clasificación

- Es similar al problema de regresión, con la diferencia de que los valores y que queremos predecir toman un pequeño número de valores discretos.
- Por ahora nos centraremos en **clasificación binaria**.
- Algunos problemas de clasificación:
  - Email: Spam x No Spam.
  - Transacciones: Fraudulento x No Fraudulento.
  - Tumor: Maligno x Benigno.

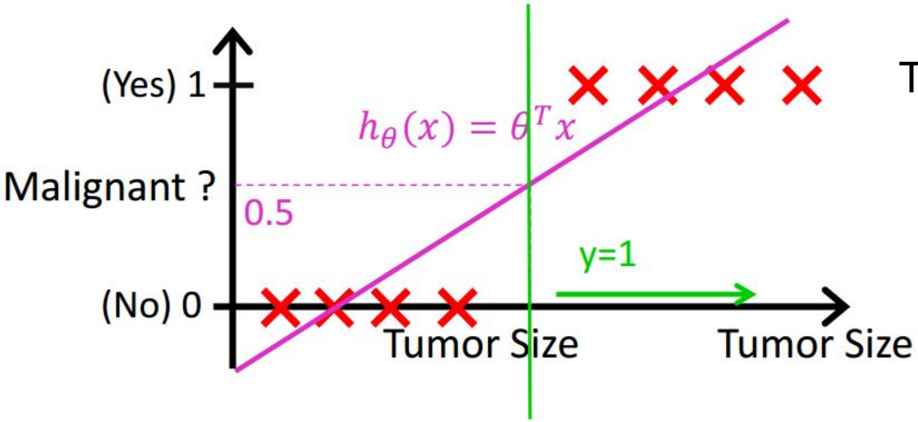
$y \in \{0,1\}$

0: “Clase Negativa” (ejemplo: No Spam)

1: “Clase Positiva” (ejemplo: Spam)

Clasificación

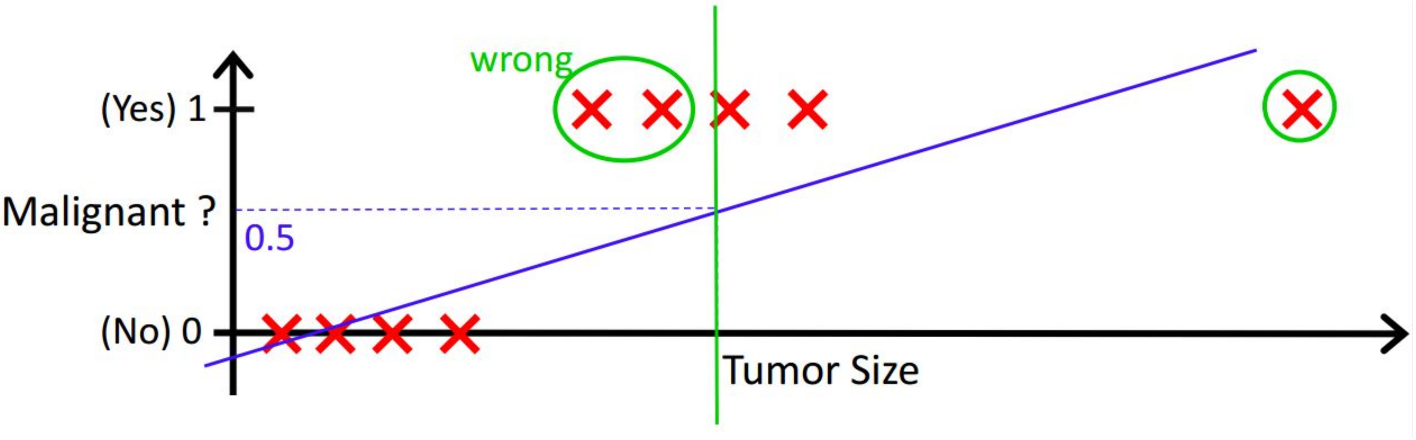
- Dados los siguientes datos para determinar si un tumor es maligno o no.
- Podríamos usar regresión linear:
  - Colocar un threshold en la salida del clasificador.
  - En nuestro ejemplo este método parece funcionar.



Threshold=0.5 a la salida del clasificador en $h_\Theta(x)$

- Si $h_\Theta(x)$ >= 0.5, predecir "y=1"
- Si $h_\Theta(x)$ < 0.5, predecir "y=0"

Clasificación



Threshold de 0.5 a la salida del clasificador en $h_\Theta(x)$

- Si $h_\Theta(x)$ >= 0.5, predecir "y=1"
- Si $h_\Theta(x)$ < 0.5, predecir "y=0"

Regresión Logística

- Nombre es algo confuso. Realmente una tecnica para clasificación, no regresión.
  - “Regresion” viene del hecho de encontrar un modelo lineal sobre un espacio de características.
- Involucra una visión más probabilista de clasificación.
- Regresión Logística es uno de los algoritmos más utilizados.
- Conocido en la literatura como logit regression, maximum-entropy classification (MaxEnt) o log-linear classifier.

Modelo de Regresión Logística

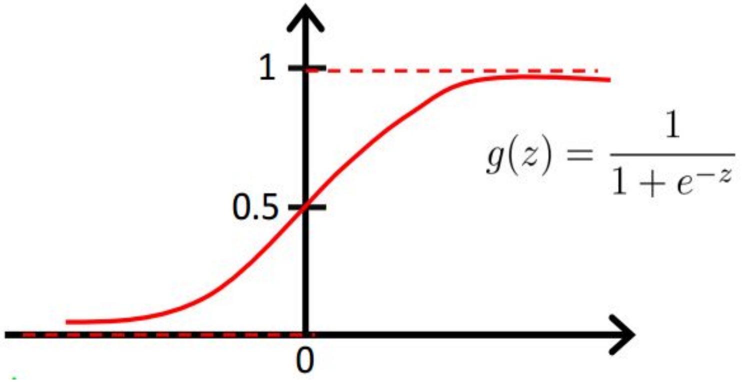
- Queremos que nuestro clasificador devuelva valores entre 0 y 1: $0 \leq h_\Theta(x) \leq 1$
- Cuando usábamos regresión lineal teníamos: $h_\Theta(x) = \theta^T x$
- Para clasificación tenemos:

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

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

$$g(\theta^T x) = \frac{1}{1+e^{-\theta^T x}}$$

Sigmoid function
Logistic function



Interpretación de salida de la hipótesis

$h_\Theta(x)$ = probabilidad estimada de y=1 para una entrada x

Ejemplo:

$$x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ tumorSize \end{bmatrix}$$

$h_\Theta(x) = 0.7$ Indica que el paciente tiene un 70% de chance de que el tumor sea maligno.

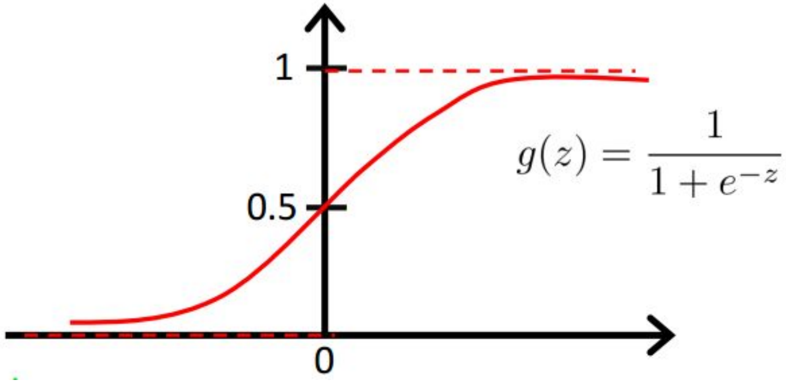
Interpretación $h_\Theta(x) = p(y = 1|x, \Theta)$ "Probabilidad de y = 1, dado x, parametrizado por θ"

$$p(y = 0|x, \Theta) = 1 - p(y = 1|x, \Theta)$$

Regresión Logística

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

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



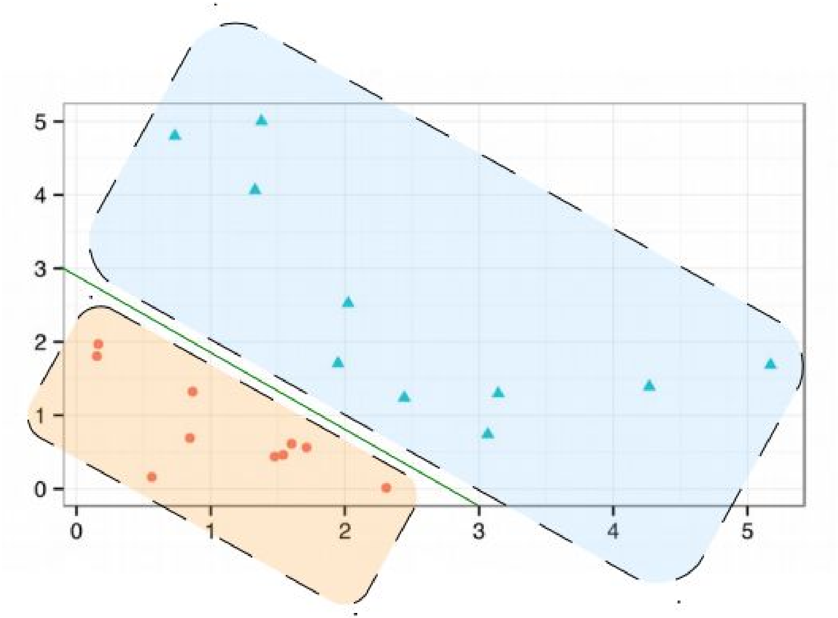
Supongamos que predecimos "y=1" si $h_\Theta \geq 0.5$

$g(z) \geq 0.5$	Cuando $z \geq 0$
$h_\Theta(x) = g(\theta^T x) \geq 0.5$	Cuando $\theta^T x \geq 0$

predecimos "y=0" si $h_\Theta < 0.5$

$g(z) < 0.5$	Cuando $z < 0$
$h_\Theta(x) = g(\theta^T x) < 0.5$	Cuando $\theta^T x < 0$

Frontera de decisión



Si $h_\Theta(x) = g(\Theta_0 + \Theta_1 x_1 + \Theta_2 x_2)$

$$\Theta = \begin{bmatrix} -3 \\ 1 \\ 1 \end{bmatrix}$$

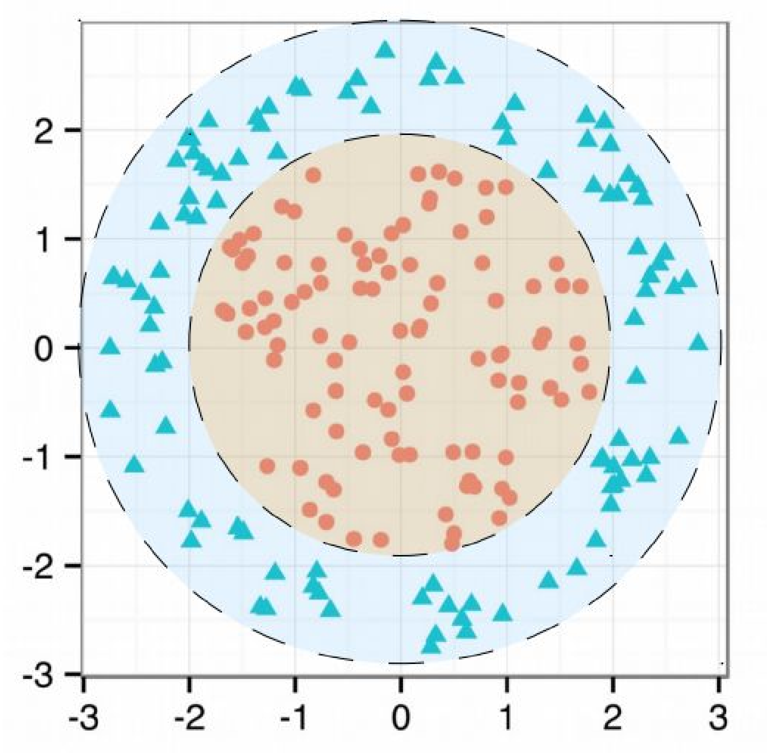
Predecimos y=1 si:

$$\Theta^T x \geq 0$$

$$-3 + x_1 + x_2 \geq 0$$

$$x_1 + x_2 \geq 3$$

Frontera de decisión



$$h_\Theta(x) = g(\Theta_0 + \Theta_1 x_1 + \Theta_2 x_2 + \Theta_3 x_1^2 + \Theta_4 x_2^2)$$

$$\Theta = \begin{bmatrix} -2 & 0 & 0 & 1 & 1 \end{bmatrix}^T$$

Predecimos y=1 si:

$$x_1^2 + x_2^2 \geq 2$$

Entrenamiento y Función de Costo

- Data de entrenamiento con **m** ejemplos y **n** características:

$$\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), ..., (x^{(m)}, y^{(m)})\}$$

- Donde:

$$x \,\epsilon\, \mathbb{R}^{n+1} \qquad\qquad x_0 = 1,\ y \,\epsilon\, \{0, 1\}$$

- Costo promedio:

$$J(\Theta) = \frac{1}{m}\sum_{i=1}^{m} Cost(h_\Theta(x^{(i)}), y^{(i)})$$

Rehusando costo de regresión lineal

- Costo de regresión lineal

$$Cost(h_\Theta(x), y) = \frac{1}{2}(h_\Theta(x) - y)^2$$

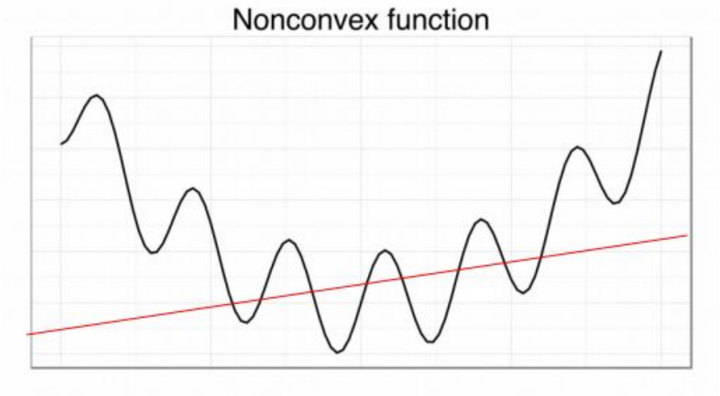
con hipótesis de regresión logística

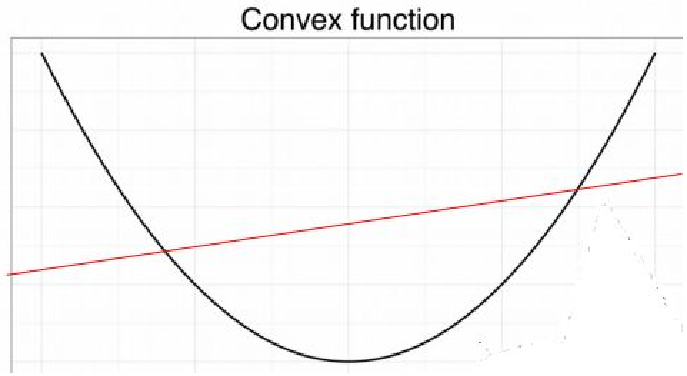
$$h_\Theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

nos lleva a un costo promedio no convexo.

$$J(\Theta) = \frac{1}{m}\sum_{i=1}^{m} Cost(h_\Theta(x^{(i)}), y^{(i)})$$

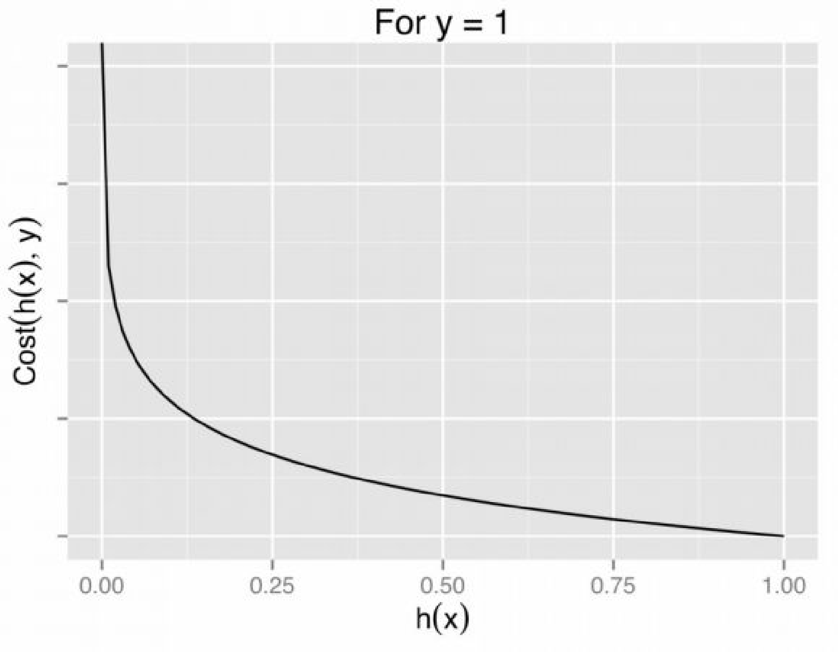
- Costo J convexo es más fácil de optimizar(no optimo local)





Función de costo

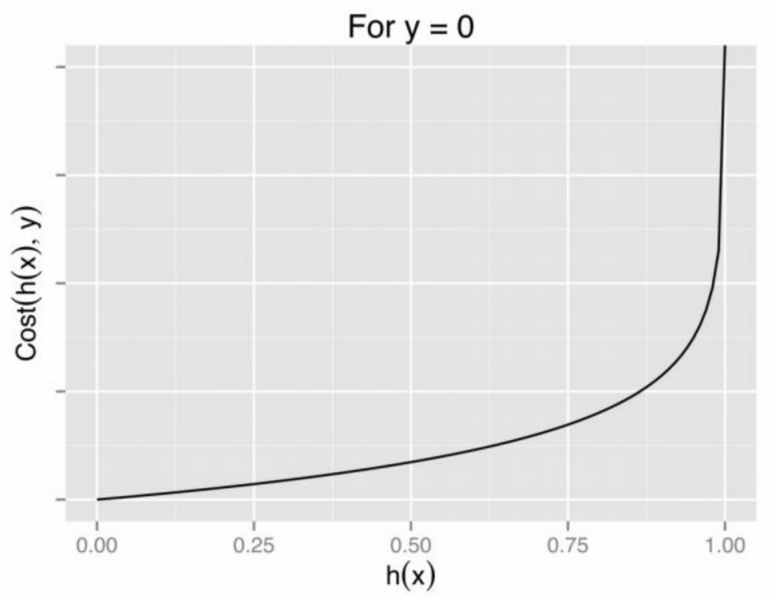
$$Cost(h_\theta(x), y) = \begin{cases} -log(h_\theta(x)) & \text{if } y = 1 \\ -log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$



- Si y=1 y h(x) = 1, Cost = 0
- Pero para h(x) → 0, Cost → ∞
- Corresponde a la intuición: Si h(x) = 0 pero el actual valor es 1, el algoritmo de aprendizaje será penalizado por un largo costo.

Función de costo

$$Cost(h_\theta(x), y) = \begin{cases} -log(h_\theta(x)) & \text{if } y = 1 \\ -log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$



- Si y=0 y h(x) = 0, Cost = 0
- Pero para
  - h(x) → 1, Cost → ∞

Función de costo simplificado

- Costo original de un ejemplo de entrenamiento

$$Cost(h_\theta(x), y) = \begin{cases} -log(h_\theta(x)) & \text{if } y = 1 \\ -log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

- Como siempre tenemos y=0 e y=1 podemos simplificar la función de costo

$$Cost(h_\Theta(x), y) = -ylog(h_\Theta(x)) - (1 - y)log(1 - h_\Theta(x))$$

- Para estar seguros, usemos la función de costo simplificada para calcular

$$Cost(h_\Theta(x), 1) = -log(h_\Theta(x))$$
$$Cost(h_\Theta(x), 0) = -log(1 - h_\Theta(x))$$

Función de costo simplificado

- Función de costo para conjunto de entrenamiento

$$J(\Theta) = \frac{1}{m}\sum_{i=1}^{m} Cost(h_\Theta(x^{(i)}), y^{(i)})$$

$$J(\Theta) = -\frac{1}{m}(\sum_{i=1}^{m} y^{(i)} log(h_\Theta(x^{(i)}) + (1 - y^{(i)}) log(1 - h_\Theta(x^{(i)}))$$

- Encontrar valores para Ɵ que minimicen J $\underset{\Theta}{argmin}\, J(\Theta)$

- Para hacer predicciones, hacemos uso de valores de Ɵ obtenidos

$$h_\Theta(x) = \frac{1}{1 + e^{-\theta^T x}} \qquad h_\Theta(x) = p(y = 1|x, \Theta)$$

Gradiente Descendente para Regresión Logística

- Gradiente Descendente para minimizar la función de costo

$$J(\Theta) = -\frac{1}{m}(\sum_{i=1}^{m} y^{(i)} log(h_\Theta(x^{(i)}) + (1 - y^{(i)}) log(1 - h_\Theta(x^{(i)}))$$

Con algoritmo similar al de regresión lineal

$$h_\Theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

$$\Theta_j = \Theta_j - \alpha \sum_{i=1}^{m} (h_\Theta(x^{(i)}) - y^{(i)}) x_j^{(i)}$$

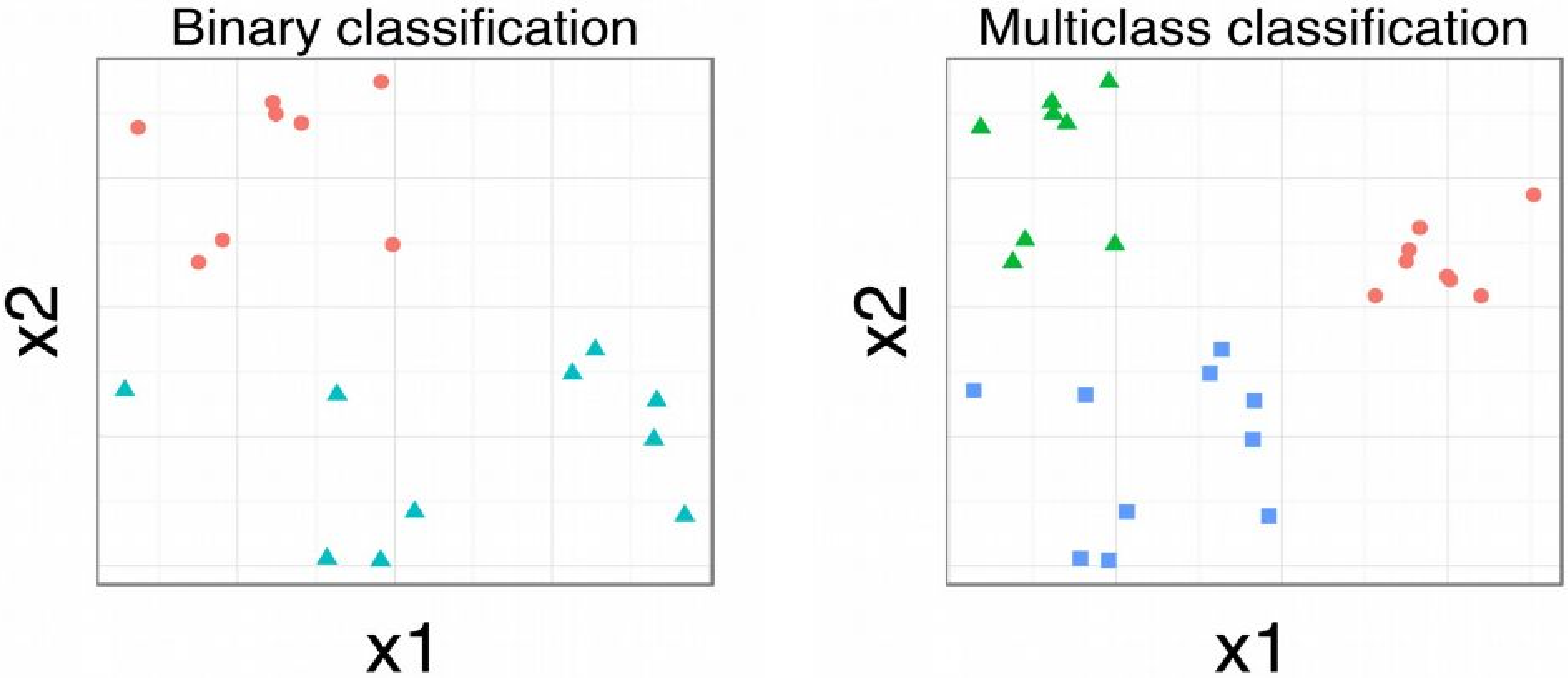
(actualización simultánea de todos los Ѳj)

Clasificación Multiclase

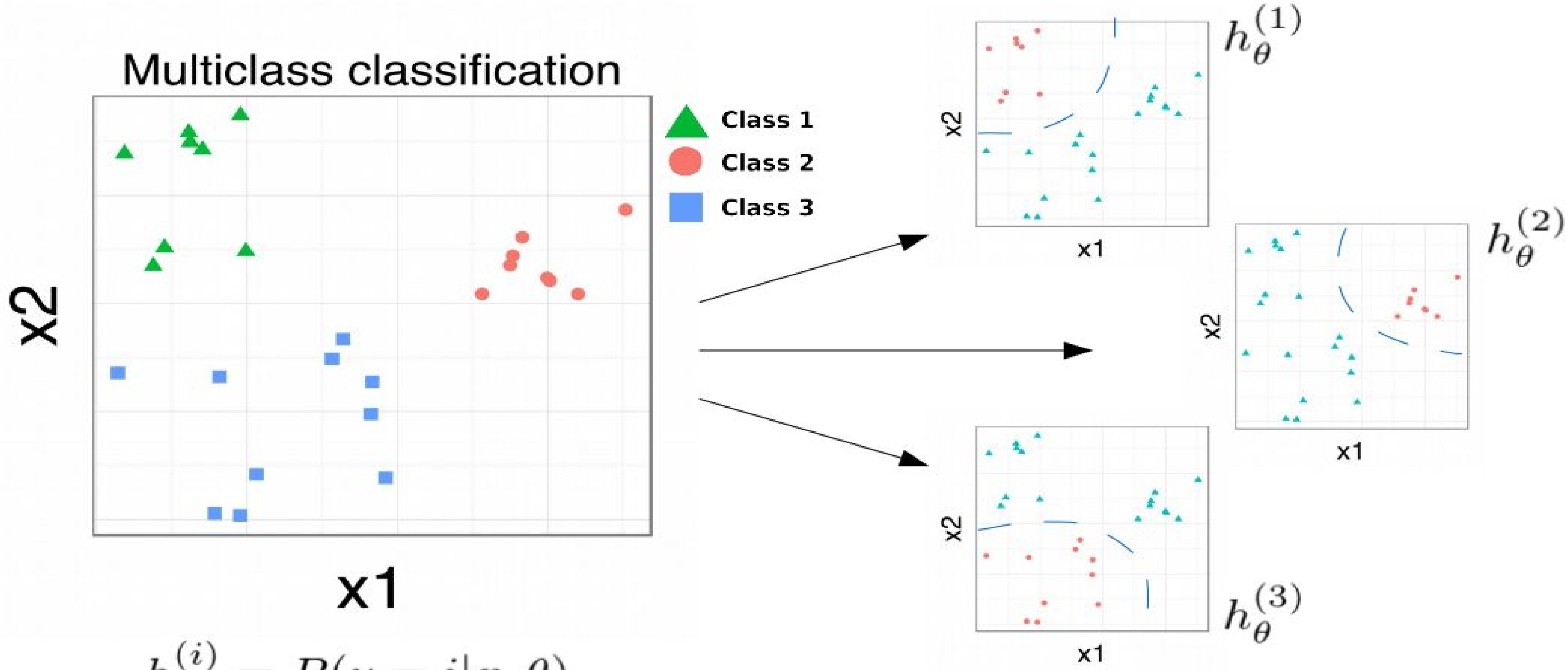
- Clases de Email: Trabajo, Amigos, Familia, Ofertas de trabajo.
- Diagnosis medica: Sano, Asma, Cancer, Gripe.
- Clima: Soleado, Nublado, Lluvia, Nieve.

$$y \in \{1,...,k\} \text{ clases}$$

Clasificación Binaria vs Multiclase



One-vs-All



$$h_\theta^{(i)} = P(y = i | x, \theta)$$

where $i \in \{1, 2, 3\}$

One-vs-All

- Entrenar un clasificador de regresión logística $h_\Theta^{(i)}(x)$

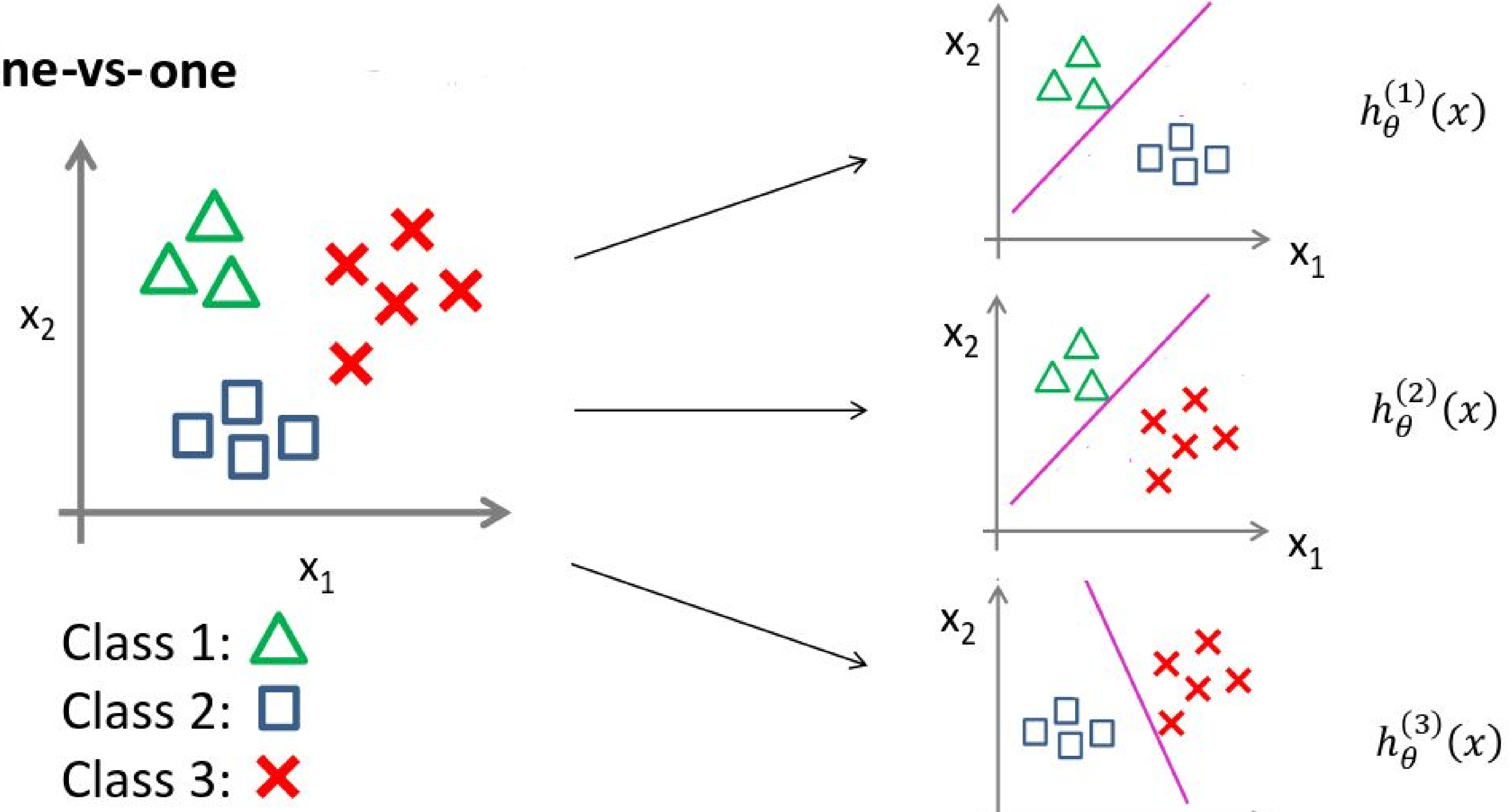
  Para cada clase i predecir la probabilidad de que y = i.

- Para una nueva entrada x, hacer la predicción en cada clasificador y escoger el de mayor probabilidad.

$$\underset{i}{argmax}\, h_\Theta^{(i)}(x)$$

- Problema cuando las clases no están balanceadas.

One-vs-one



$$h_\theta^{(i)}(x) = P(y = i | x; \theta) \qquad (i = 1, 2, 3)$$

One-vs-One

- Entrenar cada clasificador de regresión logística en pares.

$$\binom{n}{2} = \frac{n * (n-1))}{2}$$

- Para una nueva entrada x, hacer la predicción en cada clasificador e ir guardando el número de veces que una clase es preferida sobre las demás.
- Escoger la clase con la mayoría de votos.
- Método mas lento que One-vs-all.