Aprendizaje automatizado

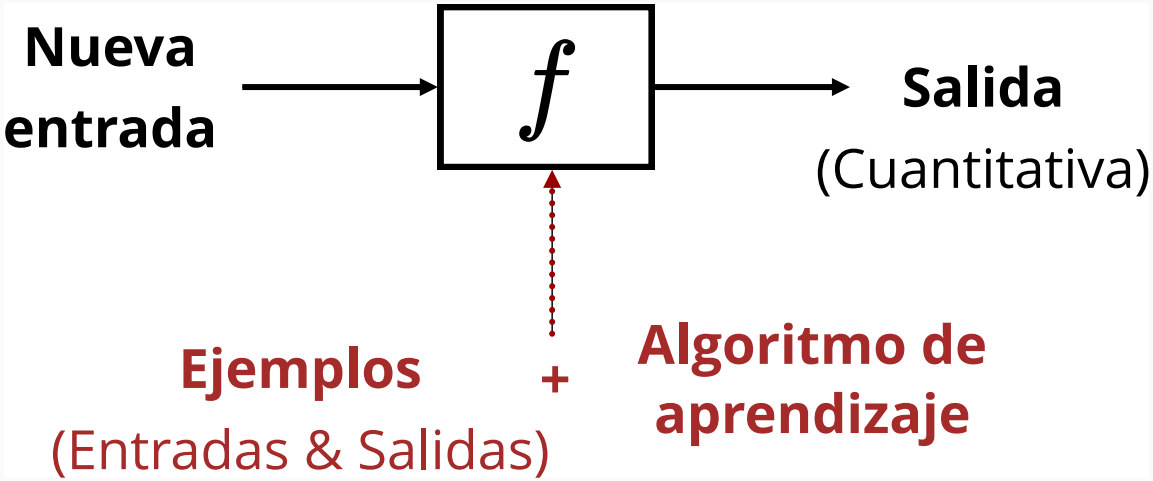
Métodos lineales de regresión y clasificación

Gibran Fuentes Pineda

Marzo 2021

Regresión

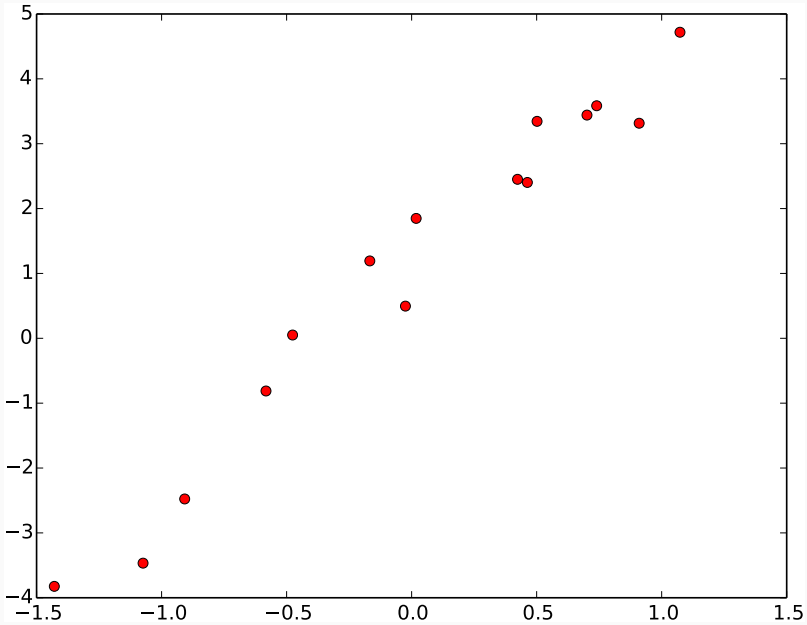
- Salida continua (cuantitativa)
- Ejemplos: predicción de temperatura de un cuarto, etc.



Prediciendo el precio de casas

- ¿Cómo podemos ajustar nuestra función f para modelar la relación entre el tamaño y el precio de casas?

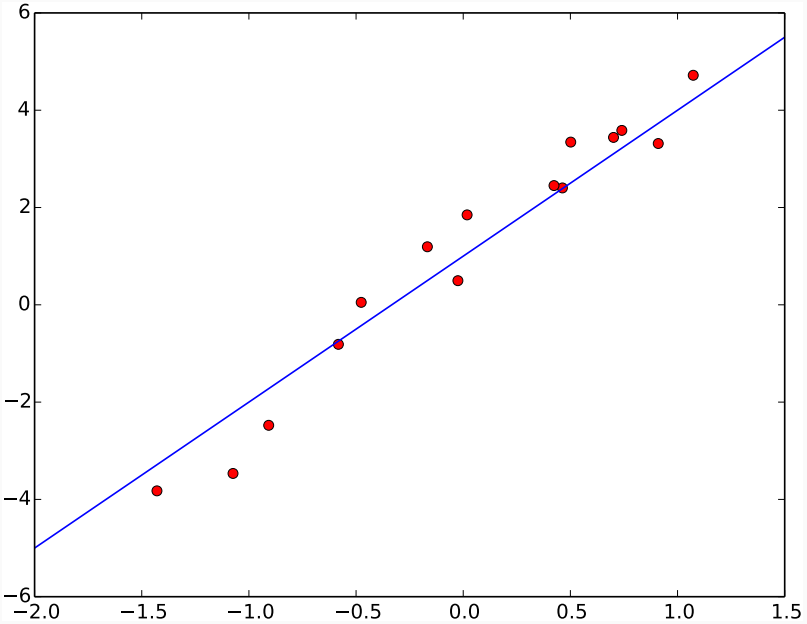
Tamaño (m^2)	Precio (USD)
489.59	489.59
556.08	556.08
570.35	570.35
772.84	772.84
970.95	970.95
1162.00	1162.00
1263.10	1263.10
⋮	⋮



Prediciendo el precio de casas

- Podemos hacer presuposiciones sobre f, por ejemplo que la relación es lineal:

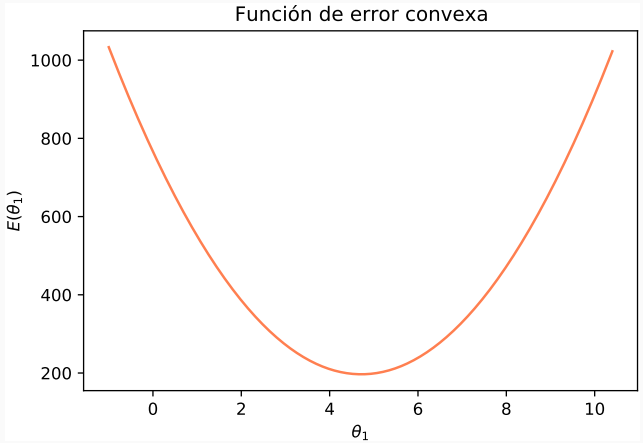
$$f_{\boldsymbol{\theta}}(x) = \theta_0 + \theta_1 \cdot x$$



¿Cómo medimos la calidad del ajuste?

- Definimos una función de error, por ejemplo la suma de errores cuadráticos:

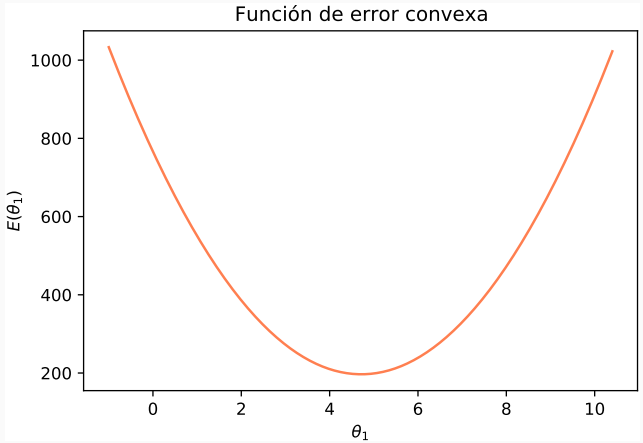
$$E(\boldsymbol{\theta}) = \frac{1}{2} \cdot \sum_{i=1}^{n} \left\{ f_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \boldsymbol{\theta}) - y^{(i)} \right\}^2$$



¿Cómo medimos la calidad del ajuste?

- Definimos una función de error, por ejemplo la suma de errores cuadráticos:

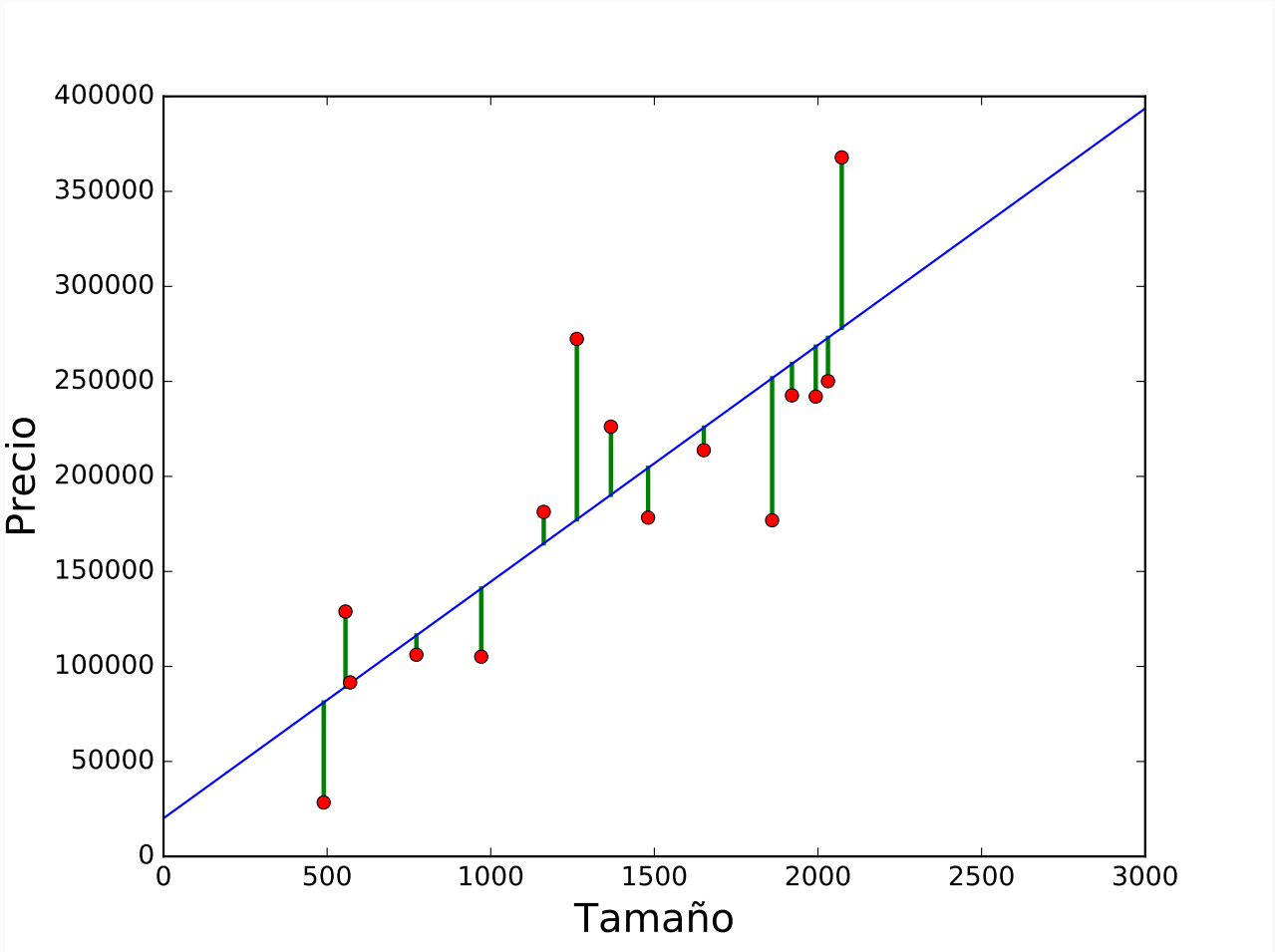
$$E(\boldsymbol{\theta}) = \frac{1}{2} \cdot \sum_{i=1}^{n} \left\{ f_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \boldsymbol{\theta}) - y^{(i)} \right\}^2$$



- Objetivo: encontrar el valor de $\boldsymbol{\theta}$ que minimice $E(\boldsymbol{\theta})$

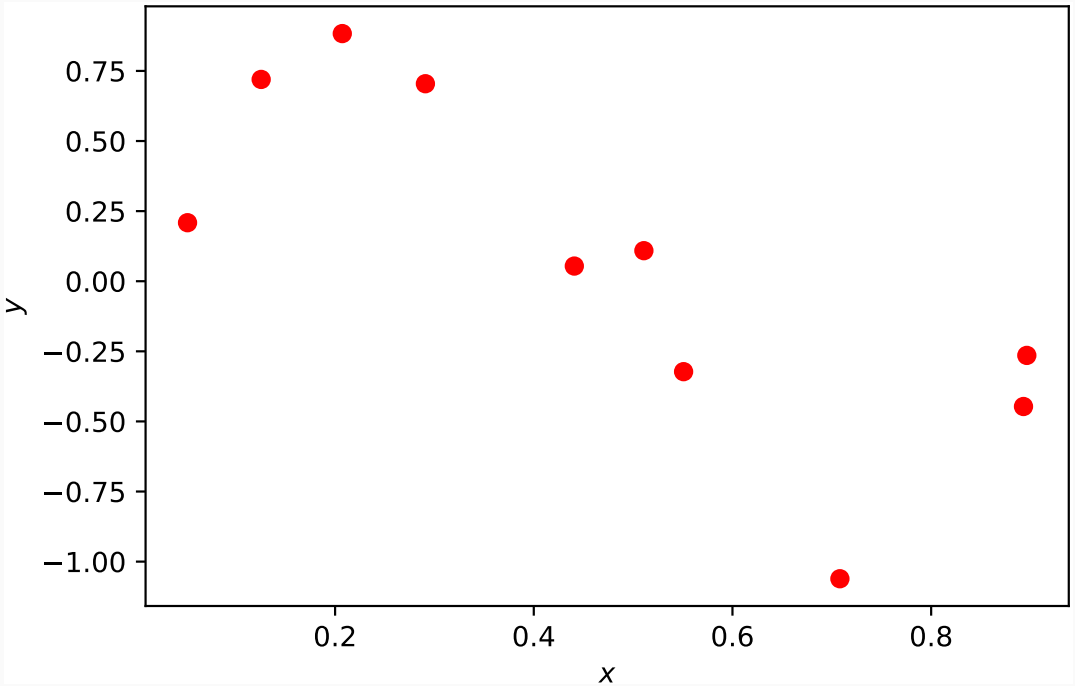
$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} E(\boldsymbol{\theta})$$

¿Cómo medimos la calidad del ajuste?



Modelando relaciones no lineales

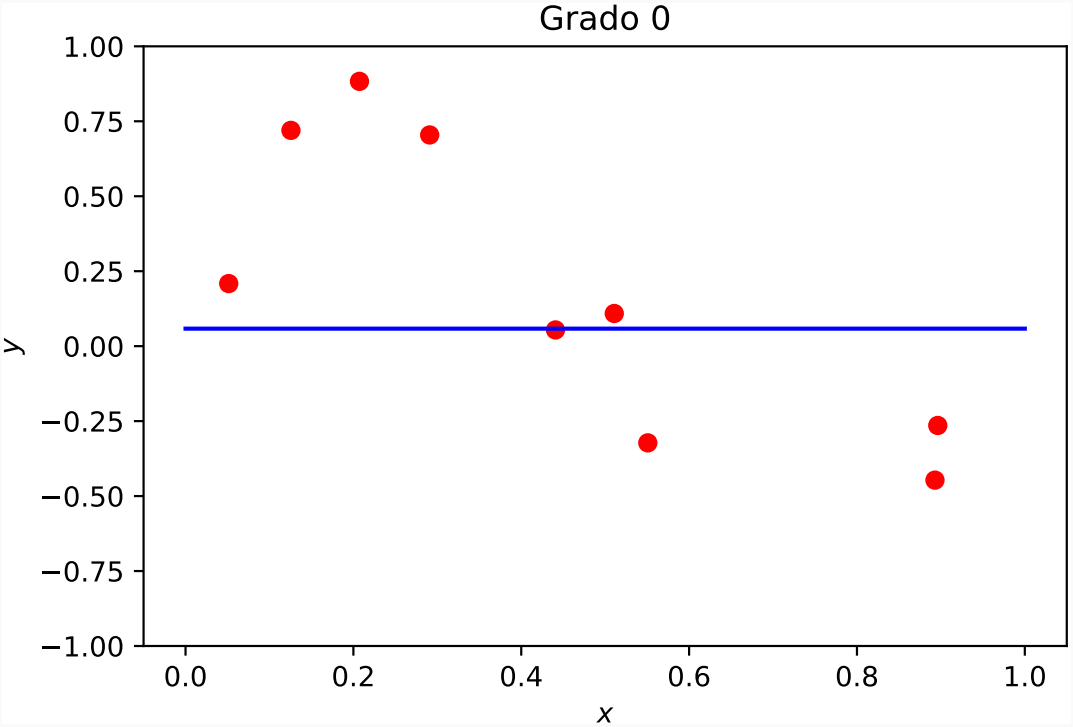
- ¿Qué función se ajusta a estos datos?



Modelando relaciones no lineales

- Podemos ajustar un polinomio de la siguiente forma[1]

$$f_{\boldsymbol{\theta}}(x) = \theta_0 + \theta_1 \cdot x + \theta_1 \cdot x^2 + \cdots + \theta_d \cdot x^d$$

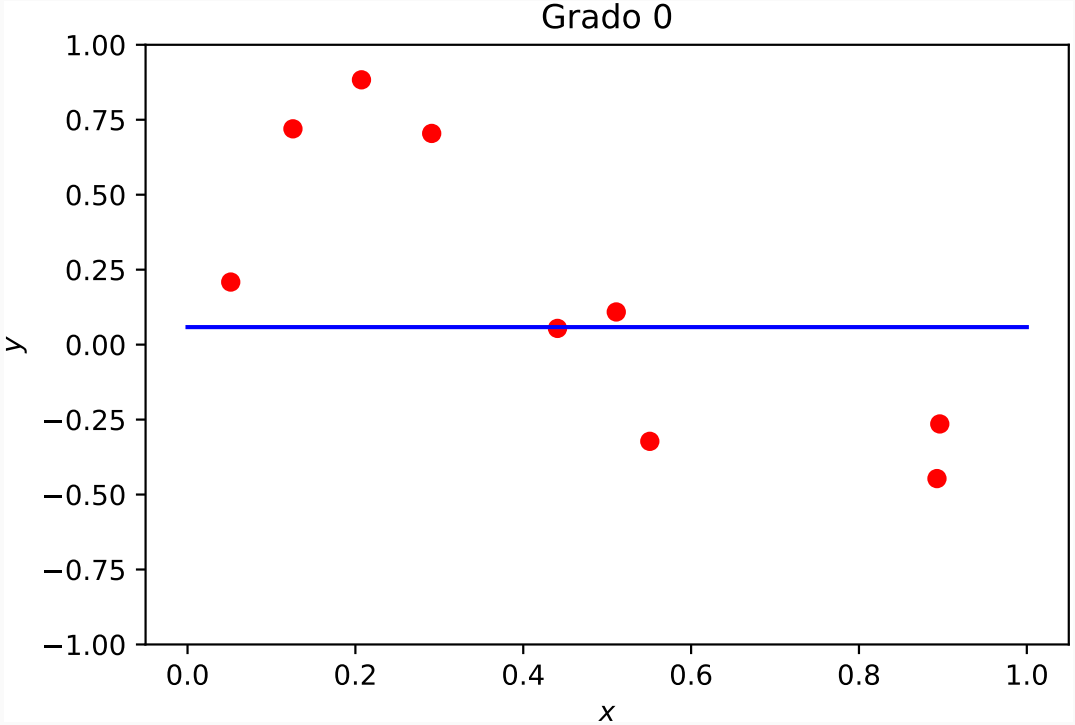


[1] Note que esta forma no está considerando interacciones

¿Qué grado del polinomio es adecuado?

- Podemos usar uno lineal nuevamente

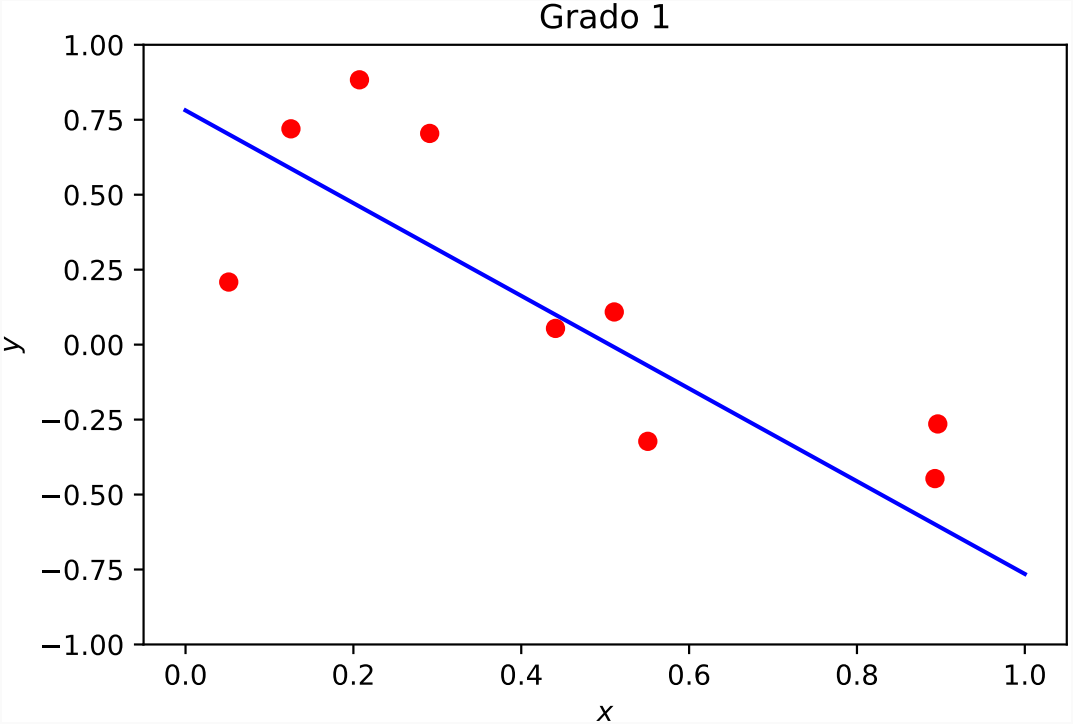
$$f_{\boldsymbol{\theta}}(x) = \theta_0 + \theta_1 \cdot x$$



¿Qué grado del polinomio es adecuado?

- O uno cuadrático

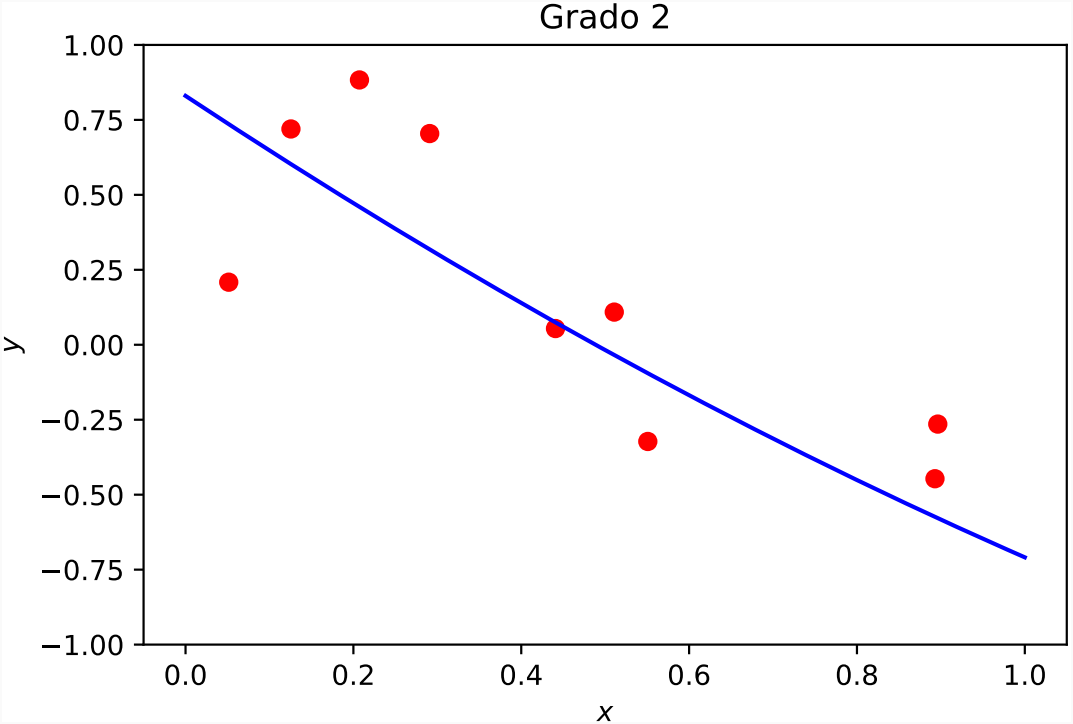
$$f_{\boldsymbol{\theta}}(x) = \theta_0 + \theta_1 \cdot x + \theta_2 \cdot x^2$$



¿Qué grado del polinomio es adecuado?

- Grado 3

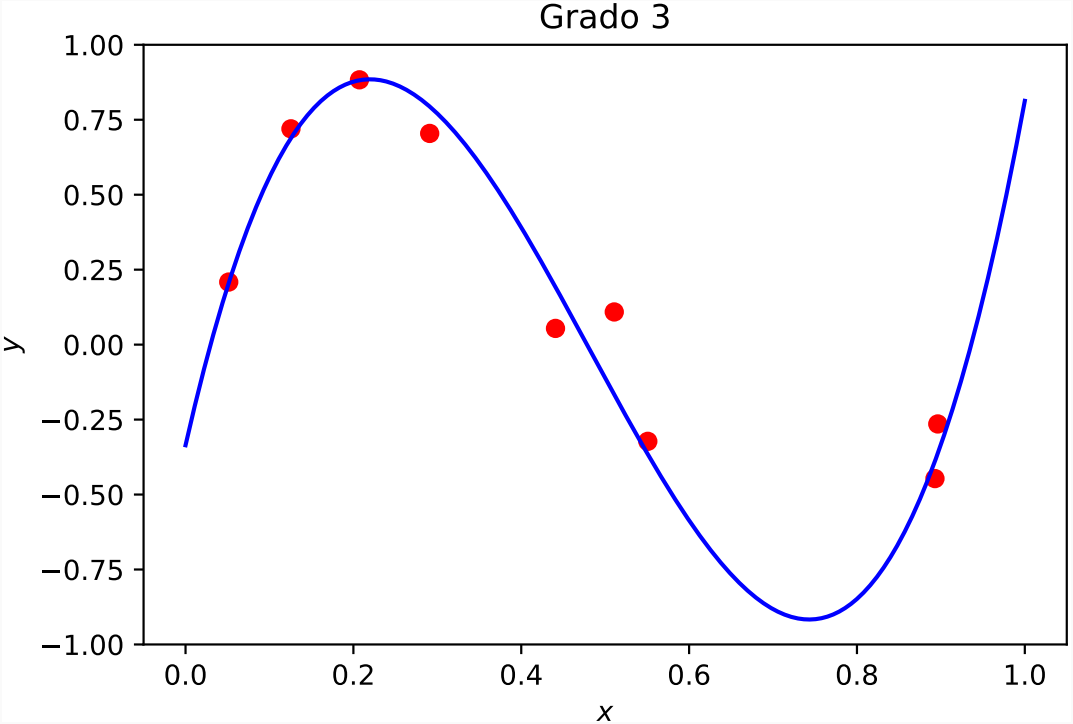
$$f_{\boldsymbol{\theta}}(x) = \theta_0 + \theta_1 + \theta_2 \cdot x^2 + \theta_3 \cdot x^3$$



¿Qué grado del polinomio es adecuado?

- O grado 9

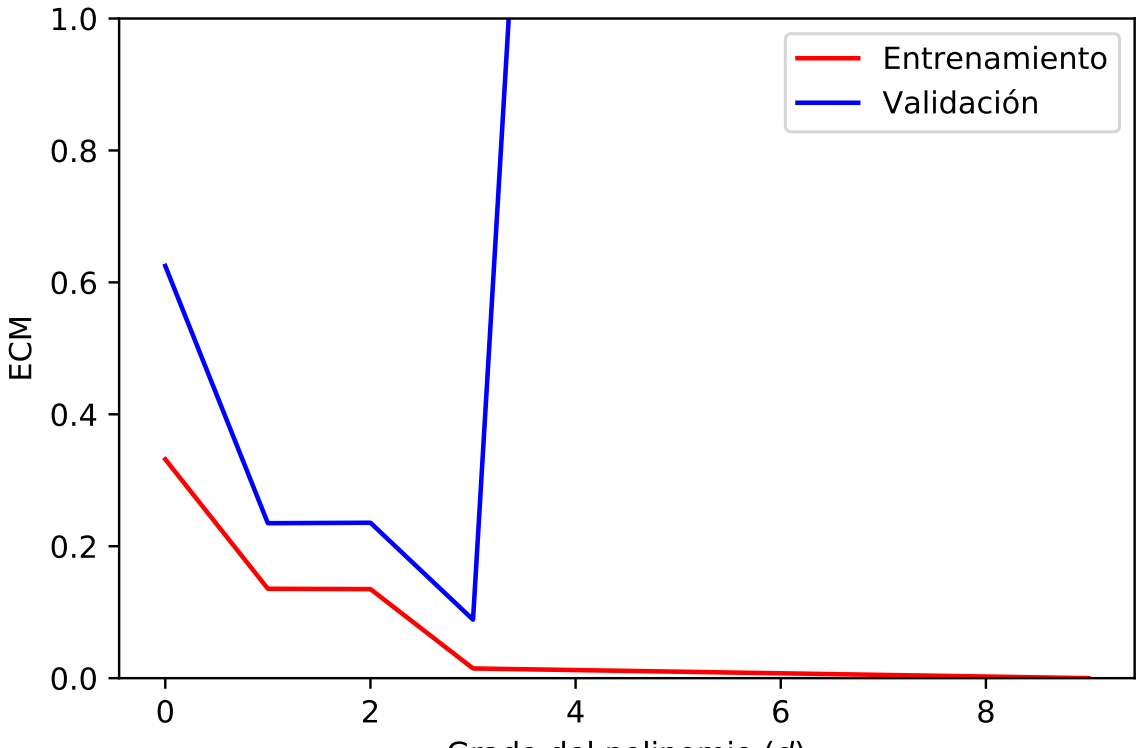
$$f_{\boldsymbol{\theta}}(x) = \theta_0 + \theta_1 + \theta_2 \cdot x^2 + \cdot x + \cdots + \theta_9 \cdot x^9$$



El problema de la generalización

- Comparamos los desempeños con distintos grados de polinomio usando el error cuadrático medio (ECM)

$$E(\boldsymbol{\theta}) = \frac{1}{n} \cdot \sum_{i=1}^{n} \left\{ f_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \boldsymbol{\theta}) - y^{(i)} \right\}^2$$



¿Por qué está sobreajustando?

	$d = 0$	$d = 1$	$d = 3$	$d = 9$
θ_0	0.05	0.78	−0.33	−17.62
θ_1		−1.54	12.32	762.18
θ_2			−36.32	12071.82
θ_3			25.14	98135.73
θ_4				−459092.41
θ_5				1301097.36
θ_6				−2263938.71
θ_7				2358449.27
θ_8				−1347197.15
θ_9				324015.43

¿Cómo evito el sobreajuste?

- Penalizando parámetros con valores grandes

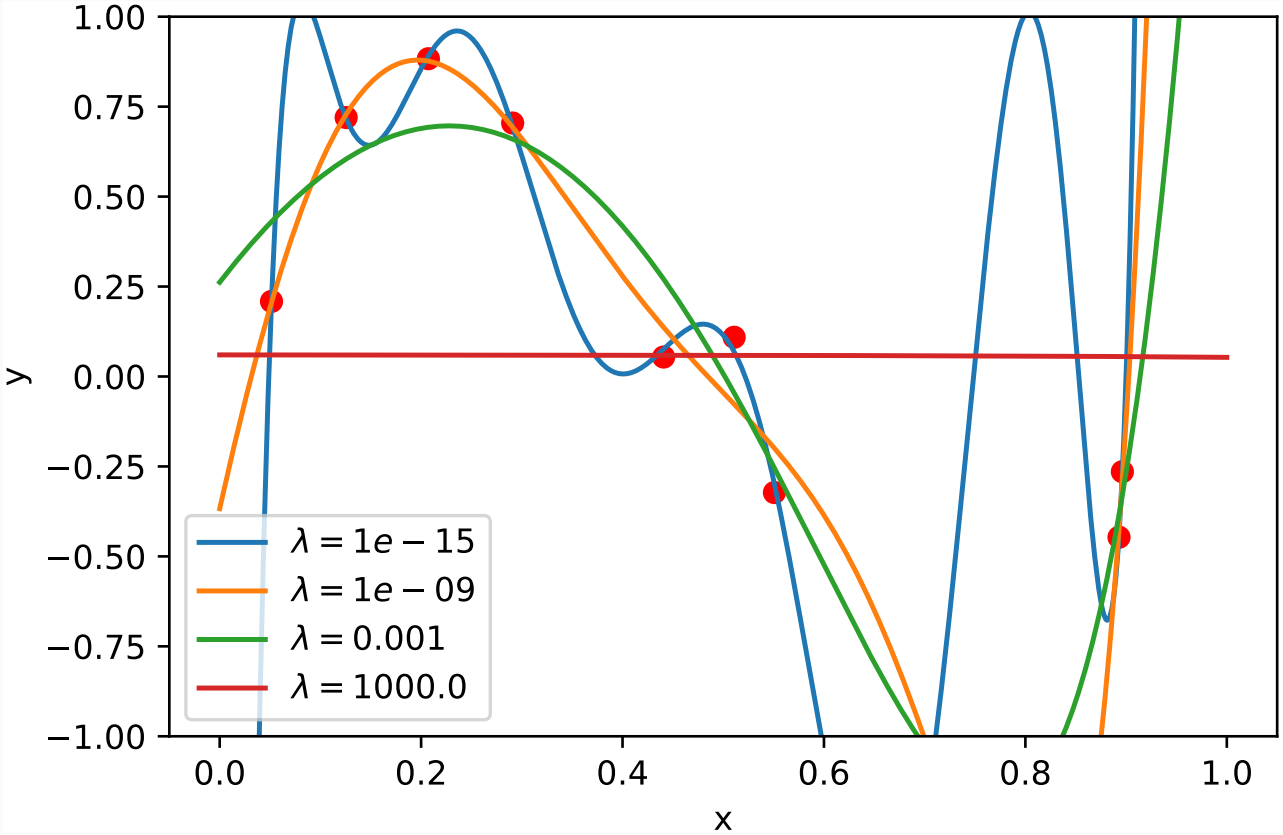
$$\tilde{E}(\boldsymbol{\theta}) = \frac{1}{2} \cdot \sum_{i=1}^{n} \left\{ f_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \boldsymbol{\theta}) - y^{(i)} \right\}^2 + \frac{\lambda}{2} \cdot \|\boldsymbol{\theta}\|_2^2$$

- λ determina la ponderación que se le da al término de penalización

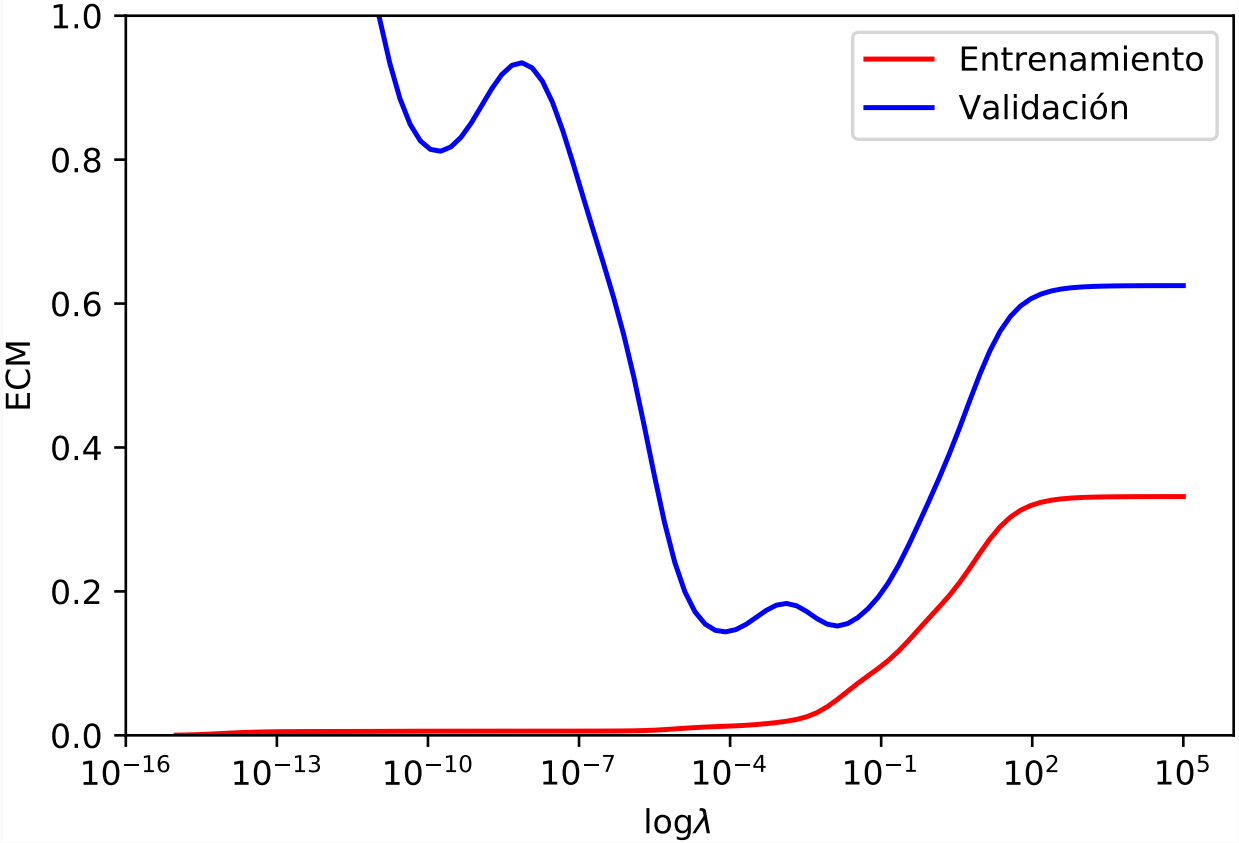
¿Cómo evito el sobreajuste?

	$\log \lambda = -\infty$	$\log \lambda = -18$	$\log \lambda = 0$
θ_0	0.35	0.35	−17.62
θ_1	232.37	4.74	−0.05
θ_2	−5321.83	−0.77	−0.06
θ_3	48568	−31.97	−0.05
θ_4	−231639.30	−3.89	−0.03
θ_5	640042.26	55.28	−0.02
θ_6	−1061800.52	41.32	−0.01
θ_7	1042400.18	−45.95	−0.00
θ_8	−557682.99	−91.53	0.00
θ_9	125201.43	72.68	0.01

Mínimos cuadrados penalizados



Mínimos cuadrados penalizados



Regresión lineal

- Modelo lineal

$$f_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\theta}) = \boldsymbol{\theta}^\top \mathbf{x} = \sum_{i=1}^{d} \theta_i \cdot x_i$$

Regresión lineal

- Modelo lineal

$$f_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\theta}) = \boldsymbol{\theta}^\top \mathbf{x} = \sum_{i=1}^{d} \theta_i \cdot x_i$$

- Con expansión de funciones base ϕ

$$f_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\theta}) = \boldsymbol{\theta}^\top \phi(\mathbf{x}) = \sum_{i=1}^{d} \theta_i \cdot \phi(\mathbf{x})_i$$

Regresión lineal

- Modelo lineal

$$f_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\theta}) = \boldsymbol{\theta}^\top \mathbf{x} = \sum_{i=1}^{d} \theta_i \cdot x_i$$

- Con expansión de funciones base ϕ

$$f_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\theta}) = \boldsymbol{\theta}^\top \phi(\mathbf{x}) = \sum_{i=1}^{d} \theta_i \cdot \phi(\mathbf{x})_i$$

- Lineal en los parámetros $\boldsymbol{\theta}$

Interpretación probabilística

- Asumiendo ruido ϵ con distribución normal en el modelo

$$y = f_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\theta}) + \epsilon$$

Interpretación probabilística

- Asumiendo ruido ϵ con distribución normal en el modelo

$$y = f_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\theta}) + \epsilon$$

- Tratamos de modelar la probabilidad condicional de la salida dados los datos y parámetros

$$P(y|\mathbf{x}, \boldsymbol{\theta}, \sigma^2) = \mathcal{N}(y|f_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\theta}), \sigma^2)$$

Obteniendo el estimador de máxima verosimilitud

- Se busca minimizar el negativo de la verosimilitud logarítmica

$$
\begin{aligned}
NVL(\boldsymbol{\theta}) &= -\sum_{i=1}^{n} \log P(y^{(i)}|\mathbf{x}^{(i)}, \boldsymbol{\theta}) \\
&= -\sum_{i=1}^{n} \log \mathcal{N}(y^{(i)}|f_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \boldsymbol{\theta}), \sigma^2) \\
&= -\frac{1}{2\sigma^2} \sum_{i=1}^{n} \left\{ f_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \boldsymbol{\theta}) - y^{(i)} \right\}^2 - \frac{n}{2} log 2\pi\sigma^2
\end{aligned}
$$

Obteniendo el estimador de máxima verosimilitud

- Se busca minimizar el negativo de la verosimilitud logarítmica

$$
\begin{aligned}
NVL(\boldsymbol{\theta}) &= -\sum_{i=1}^{n} \log P(y^{(i)}|\mathbf{x}^{(i)}, \boldsymbol{\theta}) \\
&= -\sum_{i=1}^{n} \log \mathcal{N}(y^{(i)}|f_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \boldsymbol{\theta}), \sigma^2) \\
&= -\frac{1}{2\sigma^2} \sum_{i=1}^{n} \left\{ f_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \boldsymbol{\theta}) - y^{(i)} \right\}^2 - \frac{n}{2} log 2\pi\sigma^2
\end{aligned}
$$

- Equivalente a minimizar suma de errores cuadráticos

$$
E(\boldsymbol{\theta}) = \frac{1}{2} \sum_{i=1}^{n} \{ f_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \boldsymbol{\theta}) - y^{(i)} \}^2
$$

Obteniendo el estimador de máxima verosimilitud

- Reformulando NVL

$$
\begin{aligned}
NVL(\boldsymbol{\theta}) &= \frac{1}{2}(\mathbf{y} - \mathbf{X}\boldsymbol{\theta})^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\theta}) \\
&= \frac{1}{2}\mathbf{y}^\top\mathbf{y} - \frac{1}{2}\mathbf{y}^\top\mathbf{X}\boldsymbol{\theta} - \frac{1}{2}\boldsymbol{\theta}^\top\mathbf{X}^\top\mathbf{y} + \frac{1}{2}\boldsymbol{\theta}^\top\mathbf{X}^\top\mathbf{X}\boldsymbol{\theta} \\
&= \frac{1}{2}\mathbf{y}^\top\mathbf{y} - \boldsymbol{\theta}^\top\mathbf{X}^\top\mathbf{y} + \frac{1}{2}\boldsymbol{\theta}^\top\mathbf{X}^\top\mathbf{X}\boldsymbol{\theta}
\end{aligned}
$$

Obteniendo el estimador de máxima verosimilitud

- Reformulando NVL

$$\begin{aligned} NVL(\boldsymbol{\theta}) &= \frac{1}{2}(\mathbf{y} - \mathbf{X}\boldsymbol{\theta})^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\theta}) \\ &= \frac{1}{2}\mathbf{y}^\top\mathbf{y} - \frac{1}{2}\mathbf{y}^\top\mathbf{X}\boldsymbol{\theta} - \frac{1}{2}\boldsymbol{\theta}^\top\mathbf{X}^\top\mathbf{y} + \frac{1}{2}\boldsymbol{\theta}^\top\mathbf{X}^\top\mathbf{X}\boldsymbol{\theta} \\ &= \frac{1}{2}\mathbf{y}^\top\mathbf{y} - \boldsymbol{\theta}^\top\mathbf{X}^\top\mathbf{y} + \frac{1}{2}\boldsymbol{\theta}^\top\mathbf{X}^\top\mathbf{X}\boldsymbol{\theta} \end{aligned}$$

- Derivando con respecto a $\boldsymbol{\theta}$ e igualando a cero

$$\mathbf{X}^\top\mathbf{X}\boldsymbol{\theta} = \mathbf{X}^\top\mathbf{y}$$

Obteniendo el estimador de máxima verosimilitud

- Reformulando NVL

$$\begin{aligned} NVL(\boldsymbol{\theta}) &= \frac{1}{2}(\mathbf{y} - \mathbf{X}\boldsymbol{\theta})^\top(\mathbf{y} - \mathbf{X}\boldsymbol{\theta}) \\ &= \frac{1}{2}\mathbf{y}^\top\mathbf{y} - \frac{1}{2}\mathbf{y}^\top\mathbf{X}\boldsymbol{\theta} - \frac{1}{2}\boldsymbol{\theta}^\top\mathbf{X}^\top\mathbf{y} + \frac{1}{2}\boldsymbol{\theta}^\top\mathbf{X}^\top\mathbf{X}\boldsymbol{\theta} \\ &= \frac{1}{2}\mathbf{y}^\top\mathbf{y} - \boldsymbol{\theta}^\top\mathbf{X}^\top\mathbf{y} + \frac{1}{2}\boldsymbol{\theta}^\top\mathbf{X}^\top\mathbf{X}\boldsymbol{\theta} \end{aligned}$$

- Derivando con respecto a $\boldsymbol{\theta}$ e igualando a cero

$$\mathbf{X}^\top\mathbf{X}\boldsymbol{\theta} = \mathbf{X}^\top\mathbf{y}$$

- El estimador de máxima verosimilitud es

$$\hat{\boldsymbol{\theta}}_{EMV} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y}$$

¿Y si tenemos múltiples variables de salida?

- Solución de mínimos cuadrados

$$\hat{\Theta}_{EMV} = (X^\top X)^{-1} X^\top Y$$

- Equivalente a

$$\hat{\theta}_{kEMV} = (X^\top X)^{-1} X^\top y_k$$

Obteniendo el estimador de máximo a posteriori

- Asumiendo distribución a priori normal sobre $\boldsymbol{\theta}$

$$\hat{\boldsymbol{\theta}}_{MAP} = \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} log\mathcal{N}(y^{(i)}|\theta_0 + \boldsymbol{\theta}^\top \phi(\mathbf{x}^{(i)}), \sigma^2) + \sum_{j=0}^{d} log\mathcal{N}(\theta_j|0, \tau^2)$$

Obteniendo el estimador de máximo a posteriori

- Asumiendo distribución a priori normal sobre $\boldsymbol{\theta}$

$$\hat{\boldsymbol{\theta}}_{MAP} = \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} log\mathcal{N}(y^{(i)}|\theta_0 + \boldsymbol{\theta}^\top \phi(\mathbf{x}^{(i)}), \sigma^2) + \sum_{j=0}^{d} log\mathcal{N}(\theta_j|0, \tau^2)$$

- Equivalente a minimizar suma de errores cuadráticos con los parámeros penalizados con la norma ℓ_2

$$\tilde{E}(\boldsymbol{\theta}) = \frac{1}{2}\sum_{i=1}^{n}\{f_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \boldsymbol{\theta}) - y^{(i)}\}^2 + \frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2$$

Obteniendo el estimador de máximo a posteriori

- Asumiendo distribución a priori normal sobre $\boldsymbol{\theta}$

$$\hat{\boldsymbol{\theta}}_{MAP} = \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} log\mathcal{N}(y^{(i)}|\theta_0 + \boldsymbol{\theta}^\top \phi(\mathbf{x}^{(i)}), \sigma^2) + \sum_{j=0}^{d} log\mathcal{N}(\theta_j|0, \tau^2)$$

- Equivalente a minimizar suma de errores cuadráticos con los parámeros penalizados con la norma ℓ_2

$$\tilde{E}(\boldsymbol{\theta}) = \frac{1}{2}\sum_{i=1}^{n}\{f_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \boldsymbol{\theta}) - y^{(i)}\}^2 + \frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2$$

- Derivando $\tilde{E}(\boldsymbol{\theta})$ con respecto a $\boldsymbol{\theta}$ e igualando a cero

$$\hat{\boldsymbol{\theta}}_{ridge} = (\lambda \cdot \mathbf{I}_D + \mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \mathbf{y}$$

Regularización con norma ℓ_1

- Cuando la regularización es por norma ℓ_1 se conoce como LASSO

$$\hat{\boldsymbol{\theta}}_{LASSO} = \arg\min_{\boldsymbol{\theta}} \left[\frac{1}{2} \cdot \sum_{i=1}^{n} \left\{ f_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \boldsymbol{\theta}) - y^{(i)} \right\}^2 + \frac{\lambda}{2} \cdot \|\boldsymbol{\theta}\|_1 \right]$$

Regularización con norma ℓ_1

- Cuando la regularización es por norma ℓ_1 se conoce como LASSO

$$\hat{\boldsymbol{\theta}}_{LASSO} = \arg\min_{\boldsymbol{\theta}} \left[\frac{1}{2} \cdot \sum_{i=1}^{n} \left\{ f_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \boldsymbol{\theta}) - y^{(i)} \right\}^2 + \frac{\lambda}{2} \cdot \|\boldsymbol{\theta}\|_1 \right]$$

- Optimización cuadrática: no existe solución cerrada pero existen algoritmos eficientes

Regularización con diferentes normas

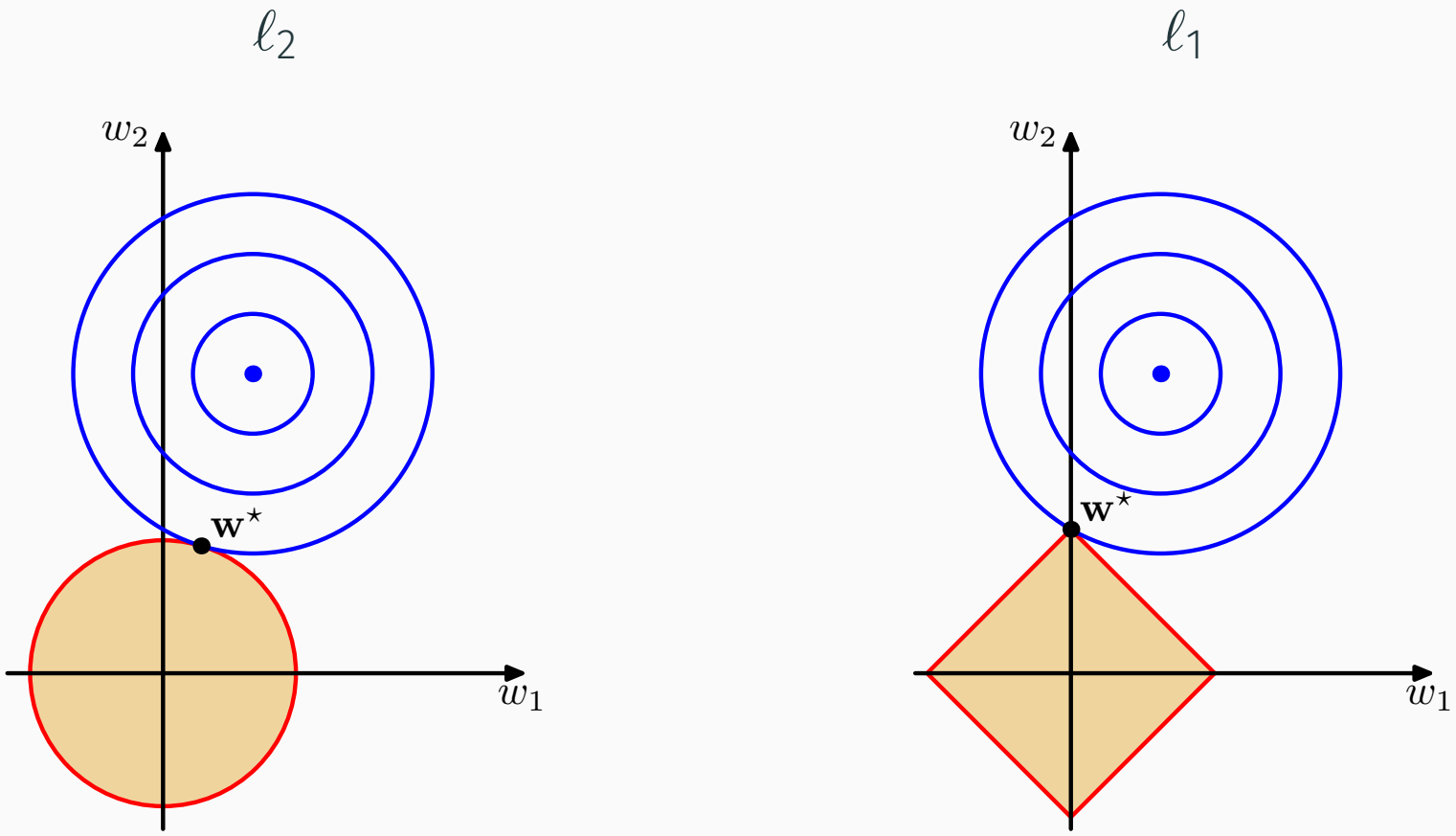


Imagen tomada de C. Bishop. PRML, 2009

Método alternativo: descenso por gradiente

- Algoritmo iterativo de primer orden que va moviendo los parámetros hacia donde el error descienda más rápido en el vecindario

$$\boldsymbol{\theta}^{[t+1]} = \boldsymbol{\theta}^{[t]} - \alpha \nabla E(\boldsymbol{\theta}^{[t]})$$

  donde

$$\nabla E(\boldsymbol{\theta}^{[t]}) = \left[\frac{\partial E}{\partial \theta_0^{[t]}}, \cdots, \frac{\partial E}{\partial \theta_d^{[t]}} \right]$$

- A α se le conoce como tasa de aprendizaje

Gradiente para regresión lineal

- Gradiente de la función de error de suma de errores cuadráticos respecto a los parámetros está dado por

$$\nabla E(\boldsymbol{\theta}) = \nabla \left[\frac{1}{2} \cdot \sum_{i=1}^{n} \left\{ f_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \boldsymbol{\theta}) - y^{(i)} \right\}^2 \right] = \mathbf{X}^\top (f_{\boldsymbol{\theta}}(\mathbf{X}, \boldsymbol{\theta}) - \mathbf{y})$$

- donde $\mathbf{X}$ es la matriz de diseño

Algoritmo del descenso por gradiente para regresión lineal

1. Asignar valores aleatorios a los parámetros $\boldsymbol{\theta}$

2. Repetir hasta que converja

$$\theta_0 \leftarrow \theta_0 - \alpha \overbrace{\frac{1}{n}\sum_{i=1}^{n}\left(f_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) - y^{(i)}\right)}^{\frac{\partial E(\theta_0)}{\partial \theta_0}}$$

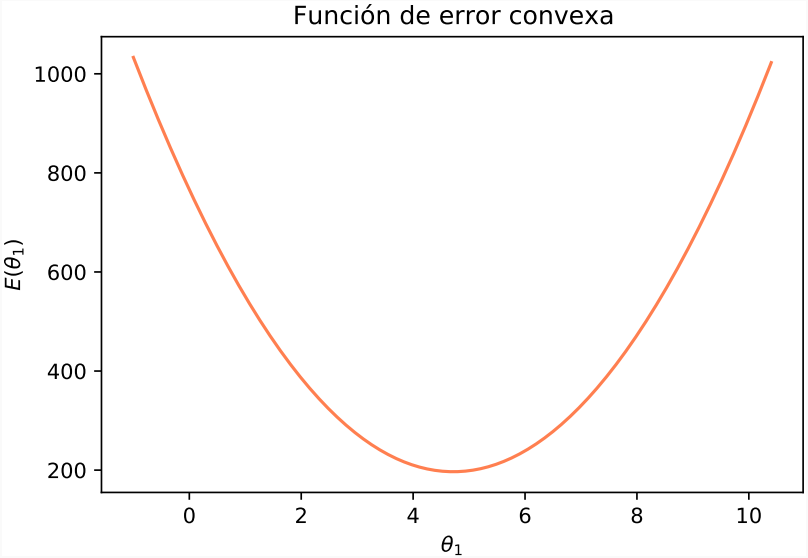
$$\theta_j \leftarrow \theta_j - \alpha \underbrace{\frac{1}{n}\sum_{i=1}^{n}\left(f_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) - y^{(i)}\right) \cdot x_j^{(i)}}_{\frac{\partial E(\theta_j)}{\partial \theta_j}}$$

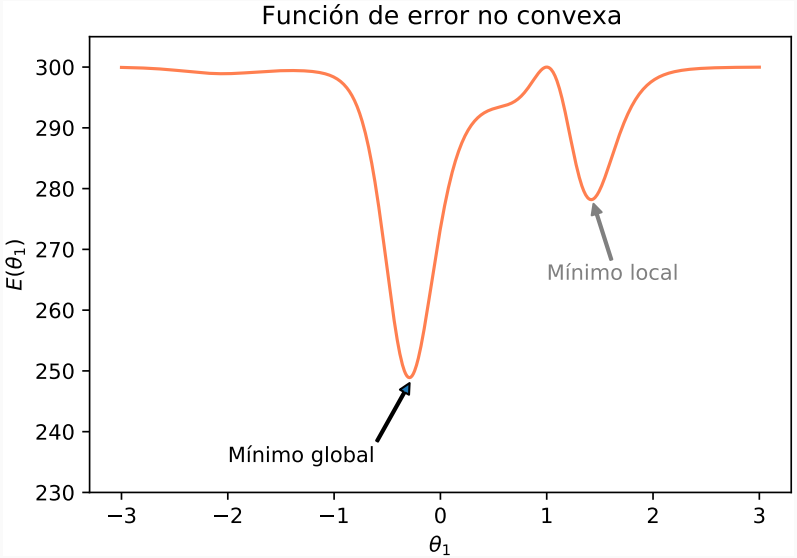
(Actualización simultánea de θ_0 y todos los θ_j)

Función de error convexa vs no convexa

- Cuando $E(\boldsymbol{\theta})$ es convexa, la solución puede converger al mínimo global

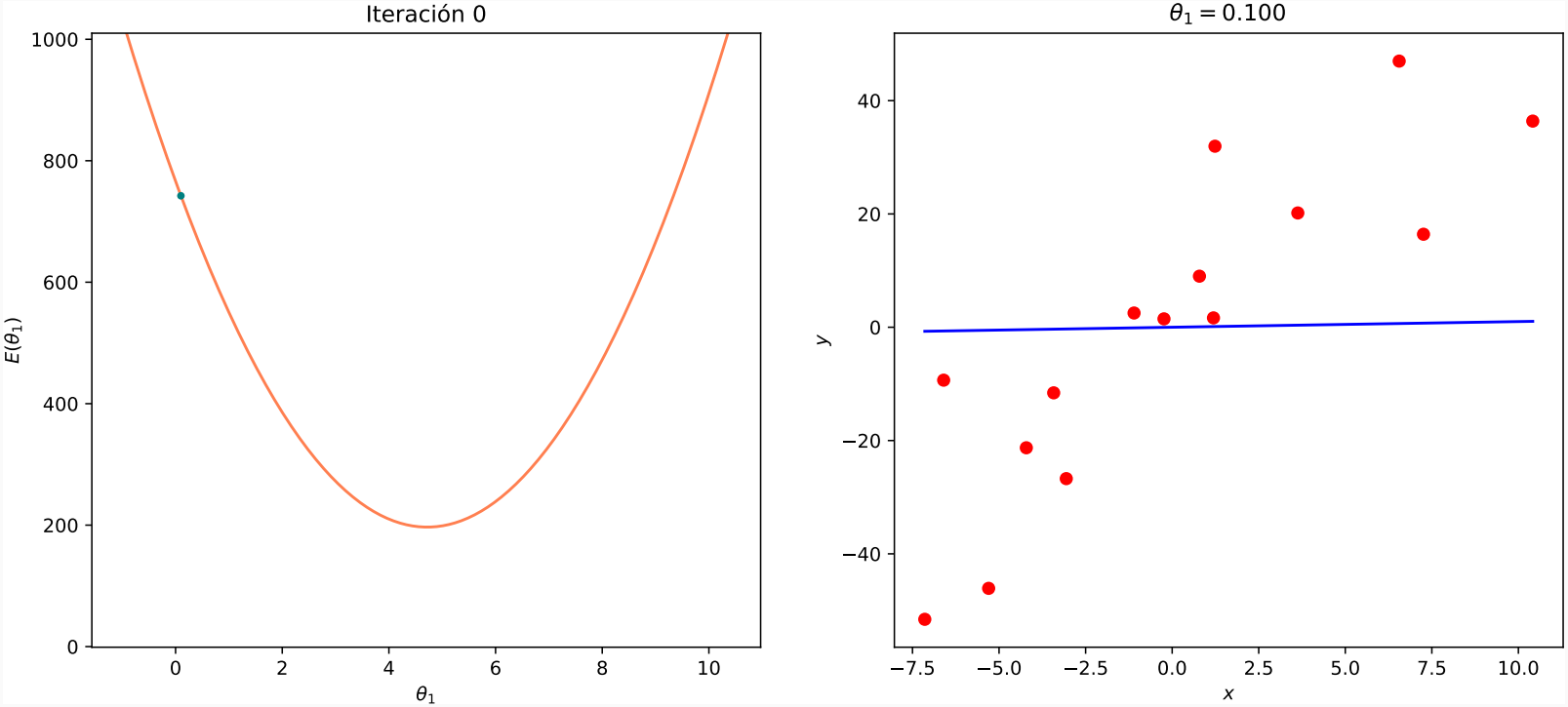
- Cuando $E(\boldsymbol{\theta})$ no es convexa, la solución puede converger a cualquier mínima





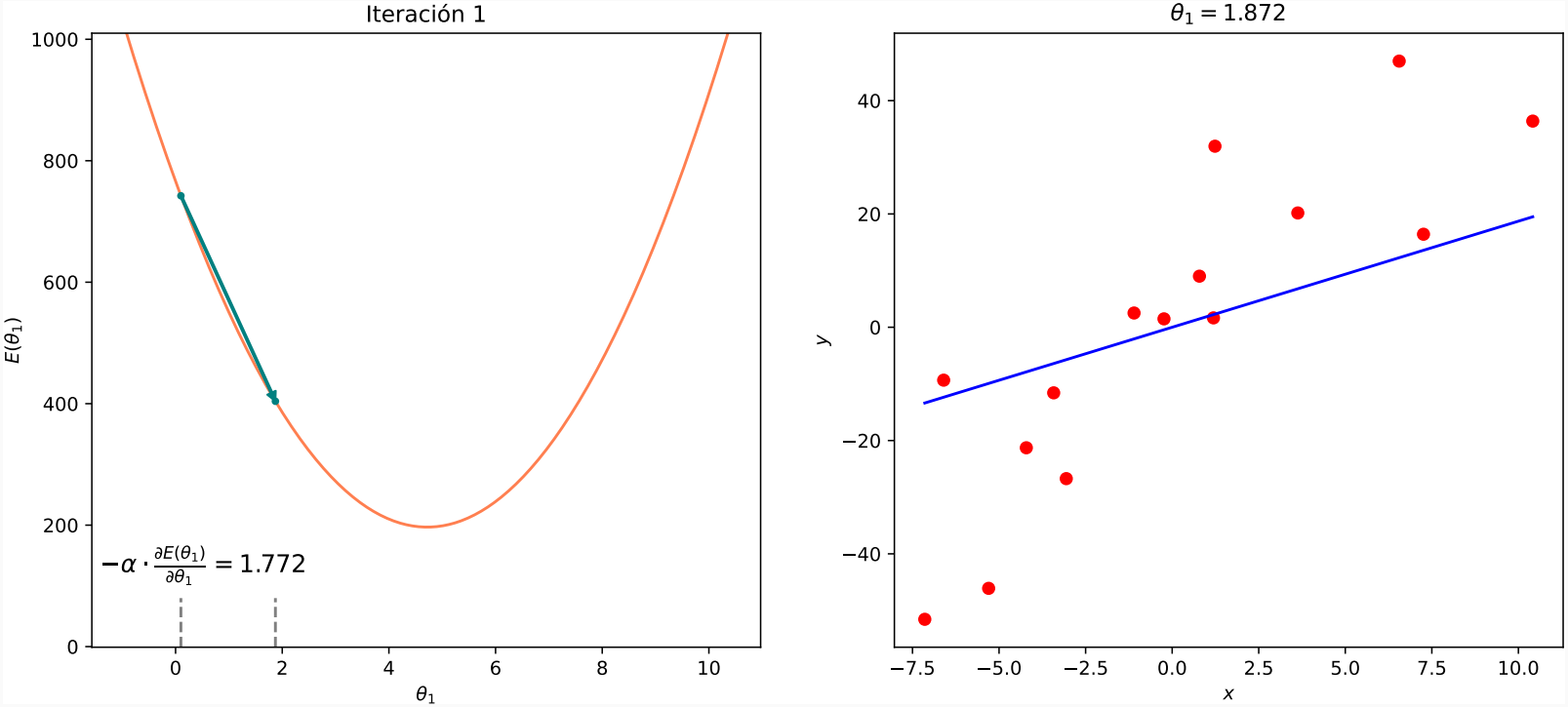
Ejemplo del descenso por gradiente (GD)

- Inicializando θ_1 con un valor menor al que minimiza la función de pérdida



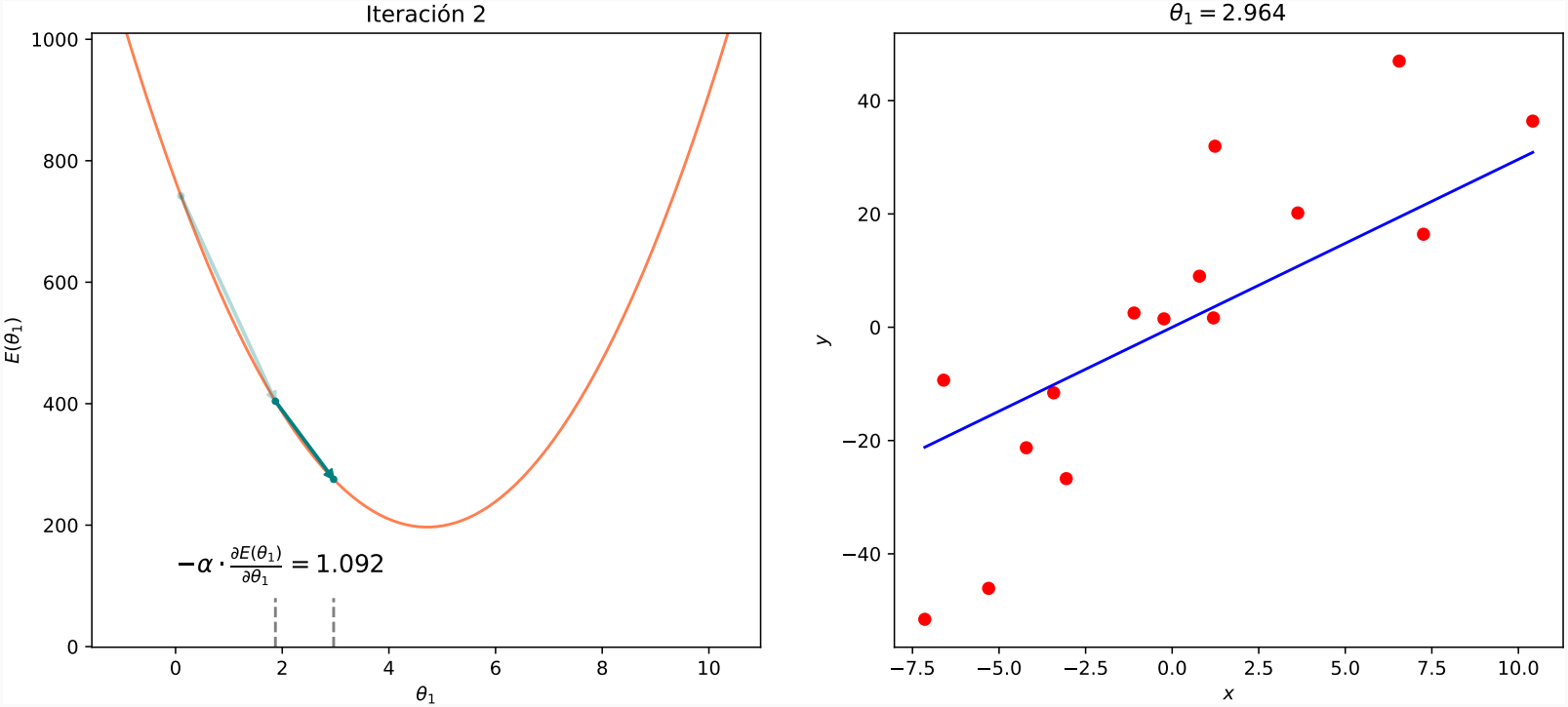
Ejemplo del algoritmo de descenso por gradiente (GD)

- Inicializando θ_1 con un valor menor al que minimiza la función de pérdida



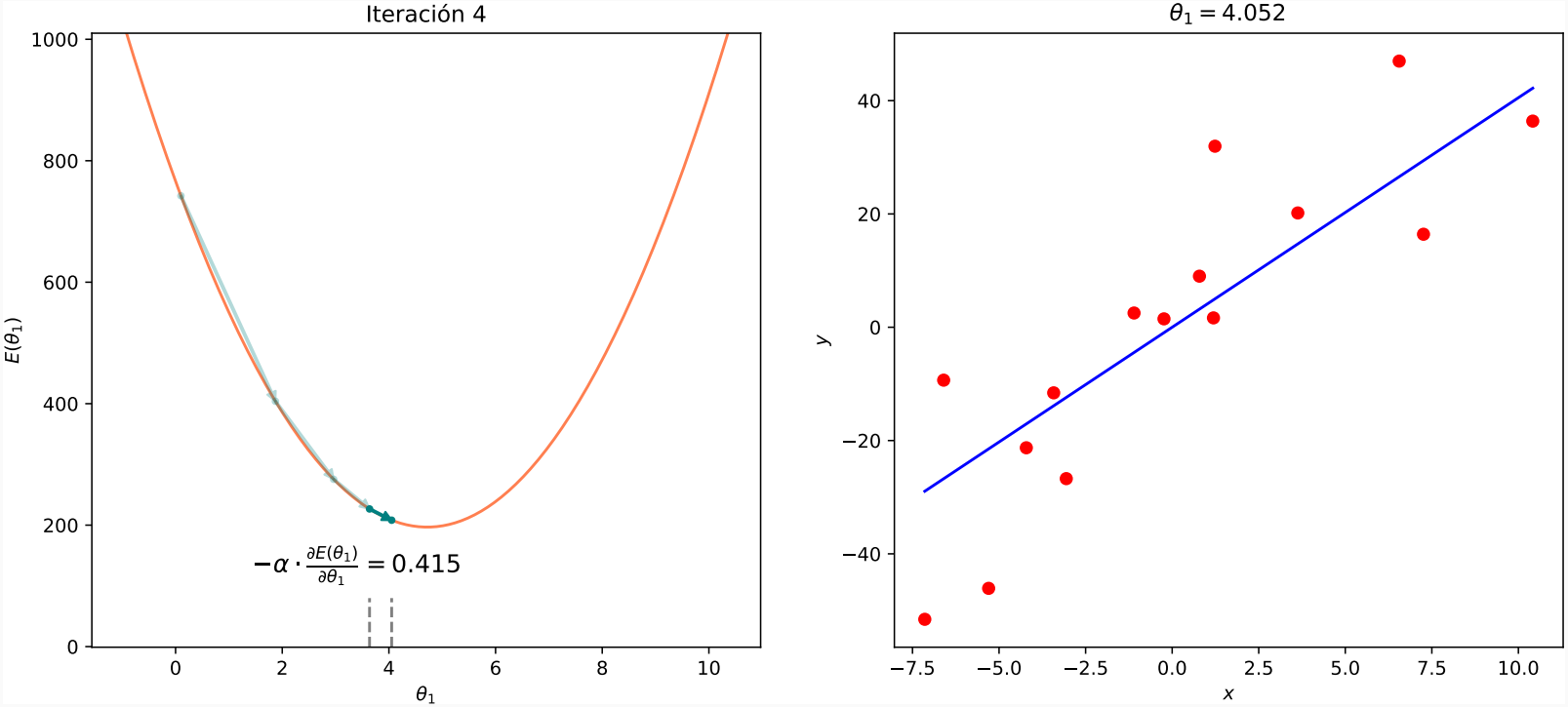
Ejemplo del algoritmo de descenso por gradiente (GD)

- Inicializando θ_1 con un valor menor al que minimiza la función de pérdida



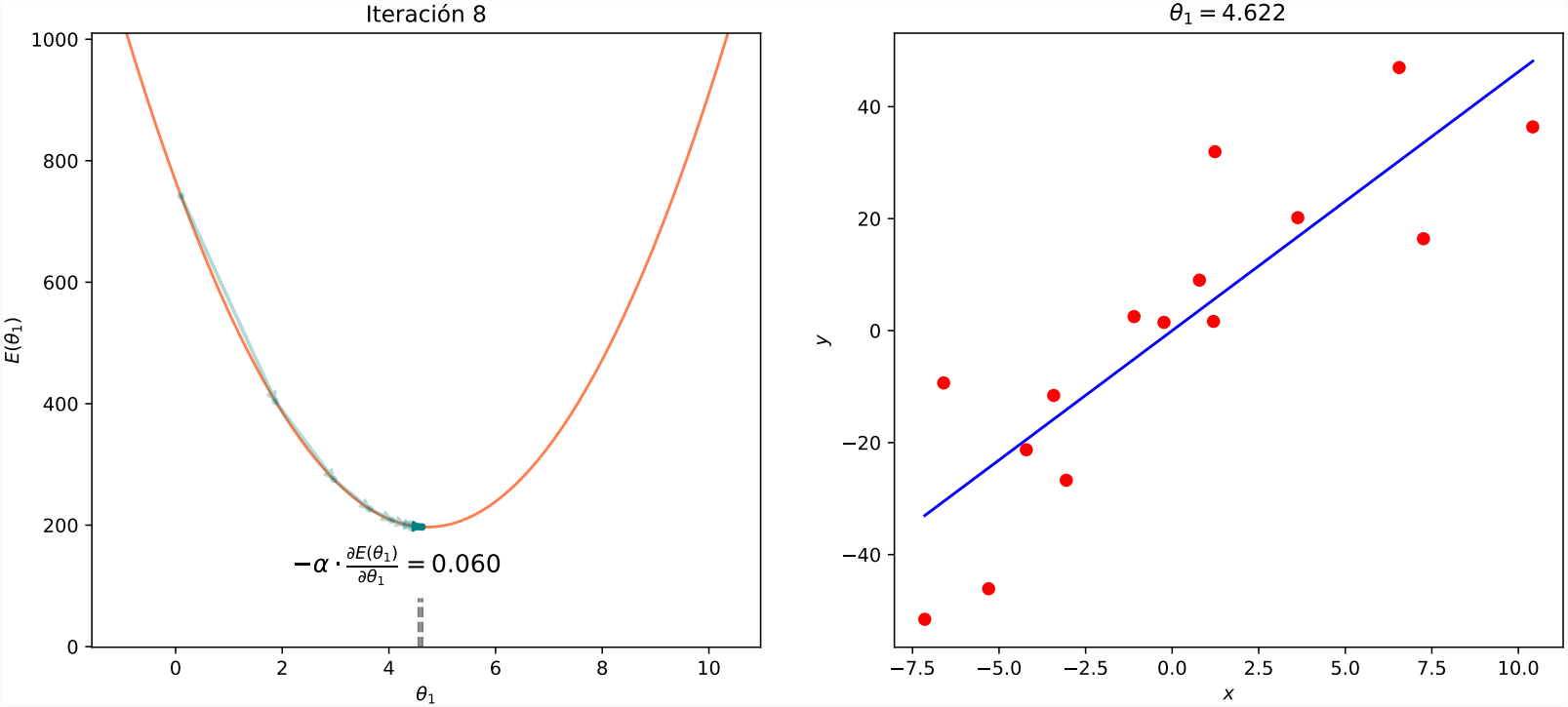
Ejemplo del descenso por gradiente (GD)

- Inicializando θ_1 con un valor menor al que minimiza la función de pérdida



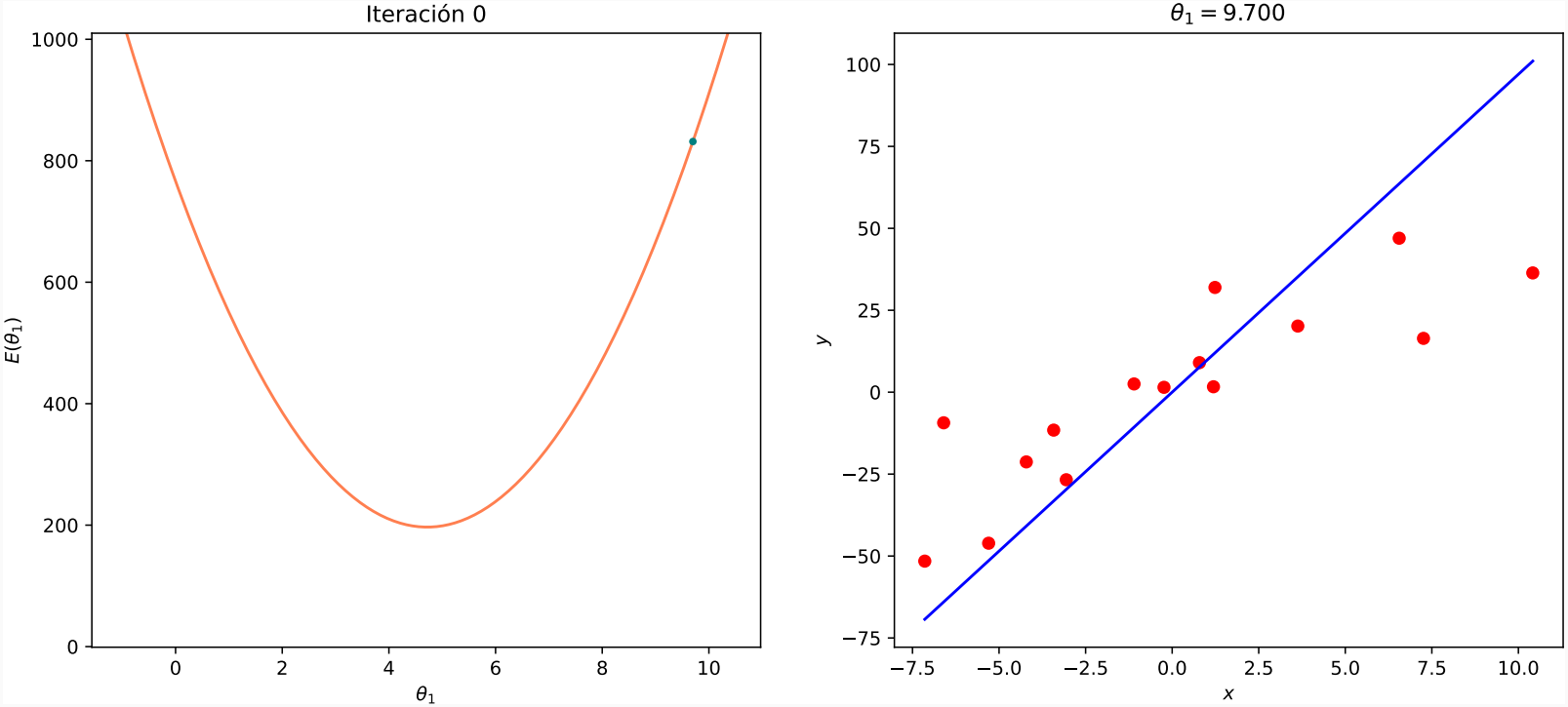
Ejemplo del descenso por gradiente (GD)

- Inicializando θ_1 con un valor menor al que minimiza la función de pérdida



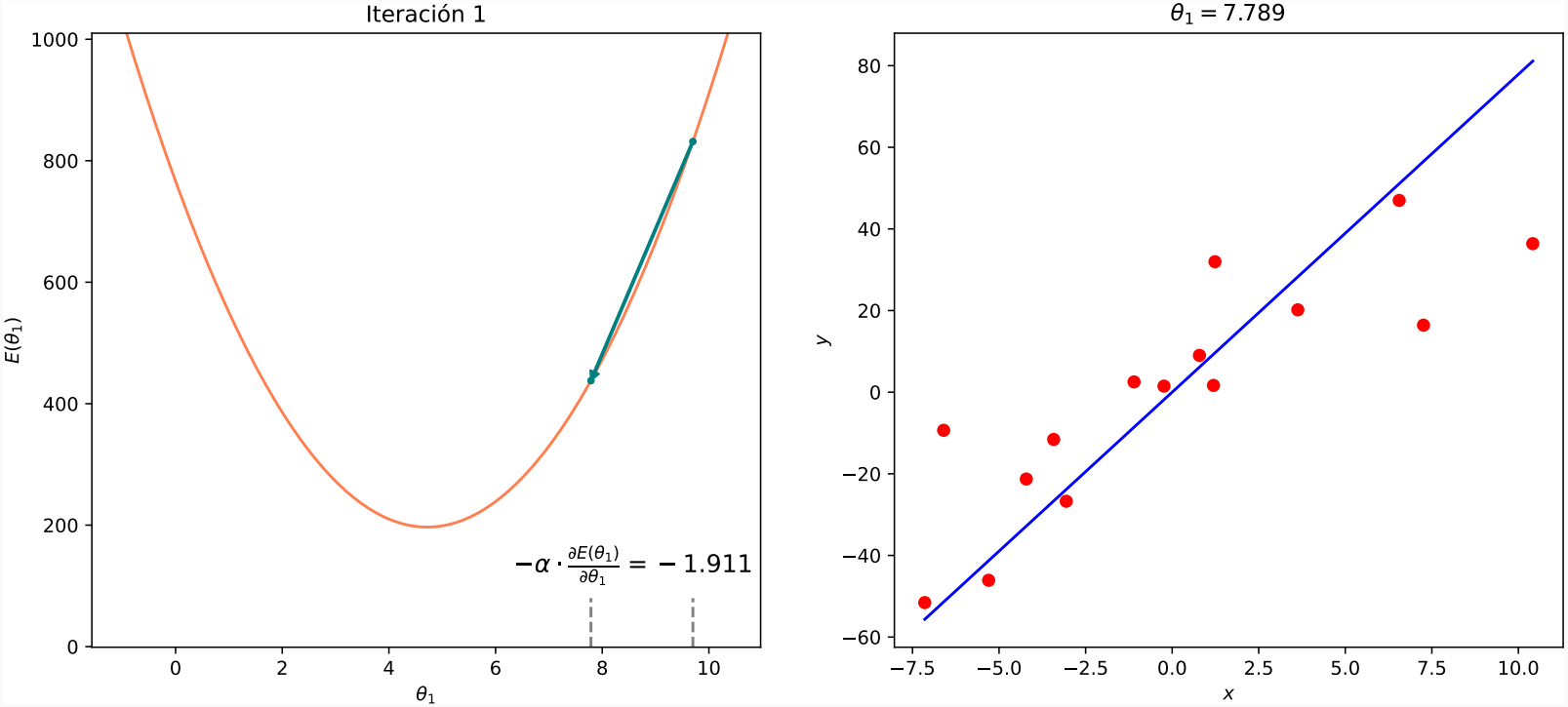
Ejemplo del descenso por gradiente (GD)

- Inicializando θ_1 con un valor mayor al que minimiza la función de pérdida



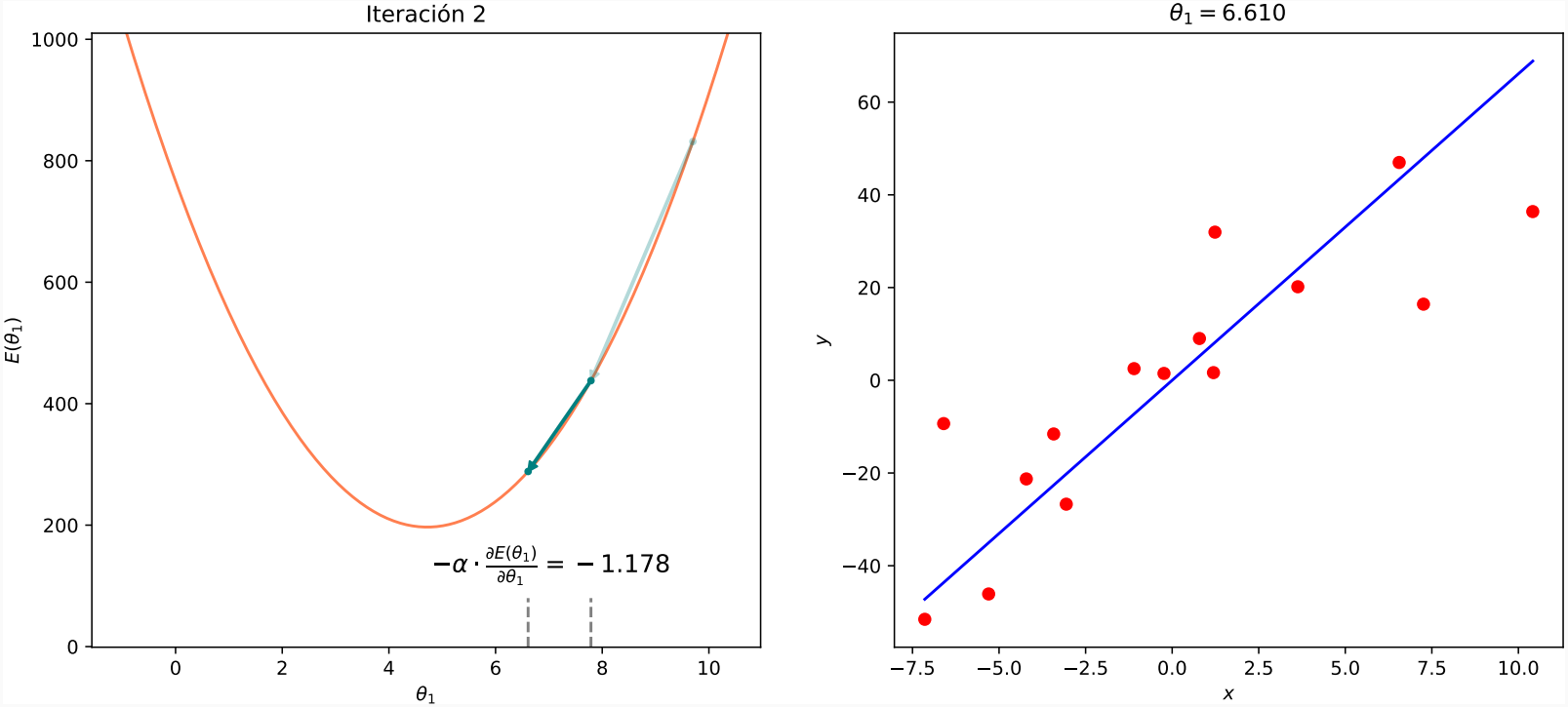
Ejemplo del descenso por gradiente (GD)

- Inicializando θ_1 con un valor mayor al que minimiza la función de pérdida



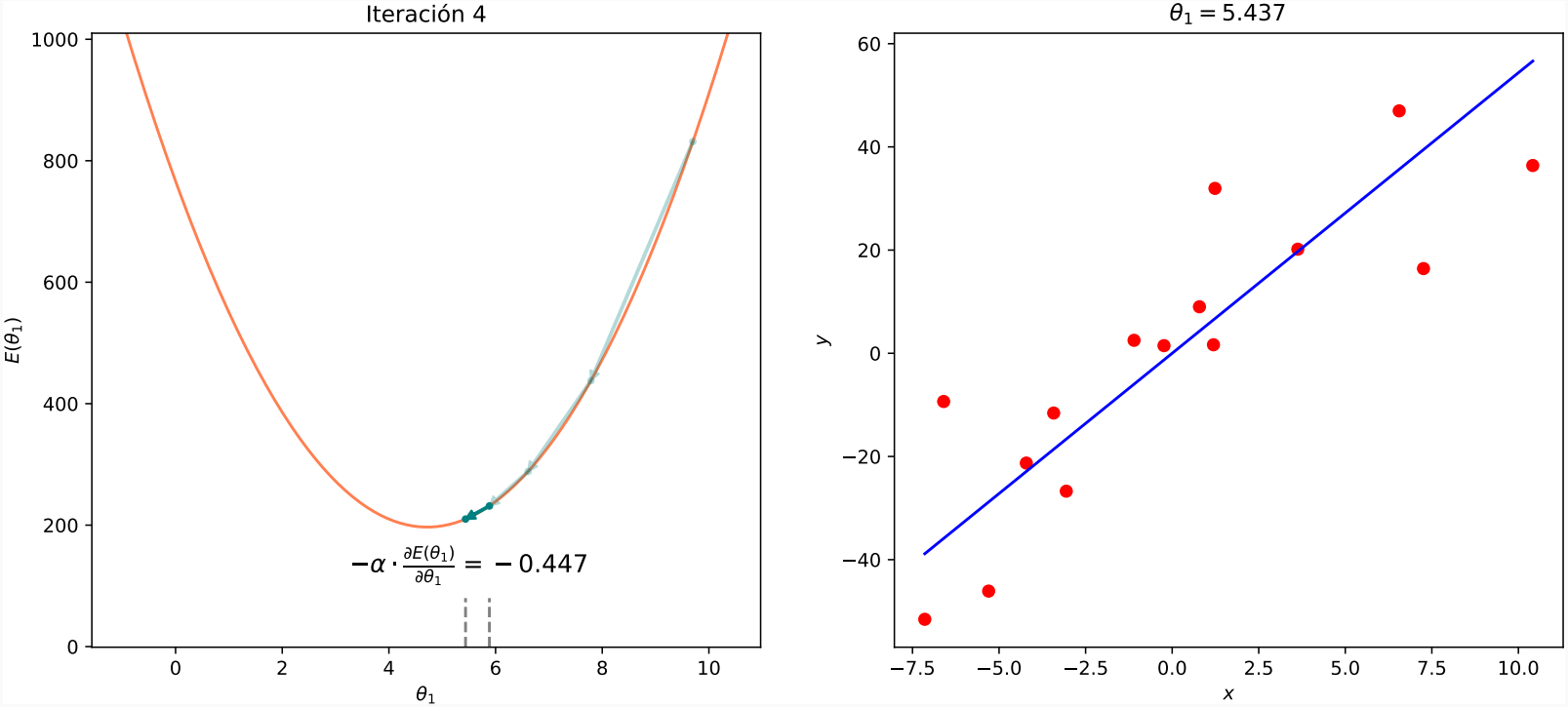
Ejemplo del descenso por gradiente (GD)

- Inicializando θ_1 con un valor mayor al que minimiza la función de pérdida



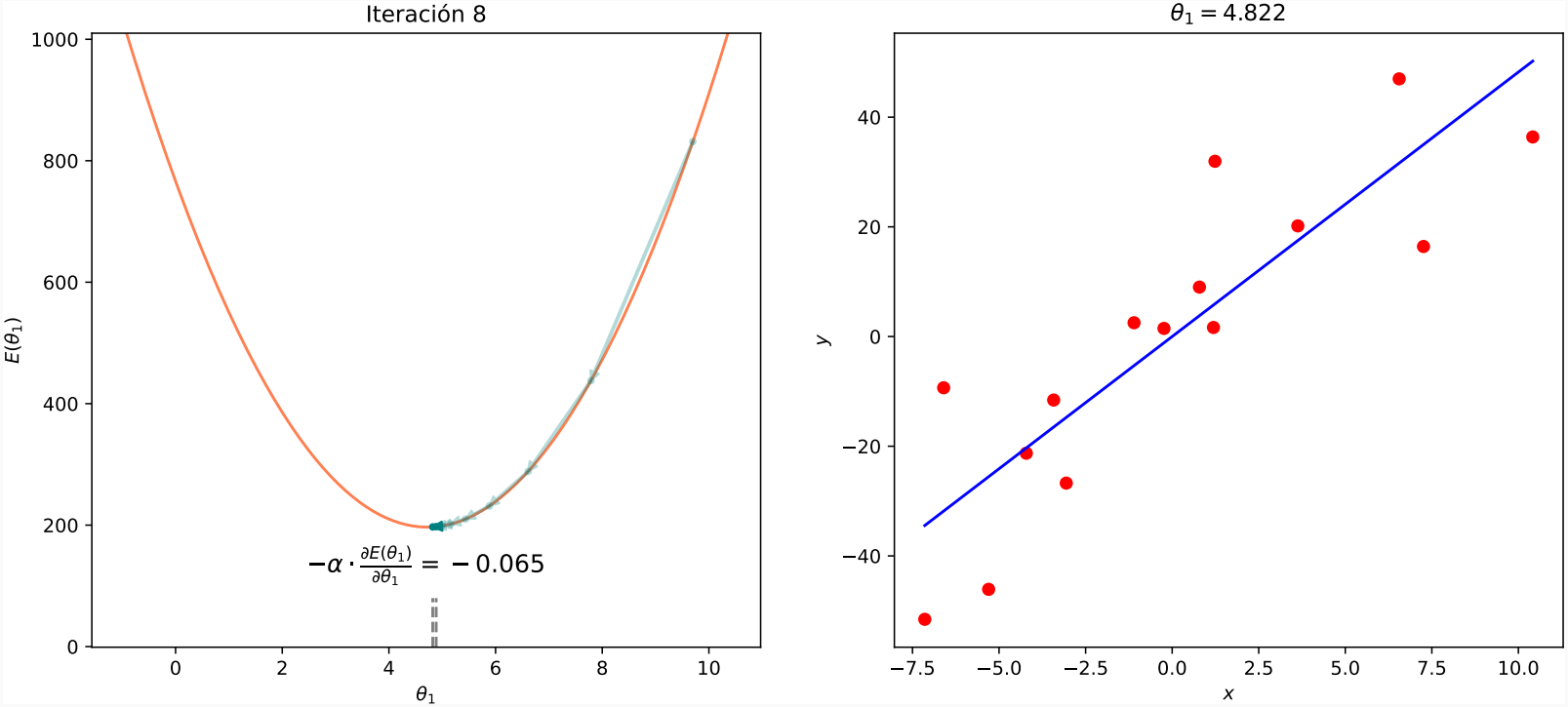
Ejemplo del descenso por gradiente (GD)

- Inicializando θ_1 con un valor mayor al que minimiza la función de pérdida

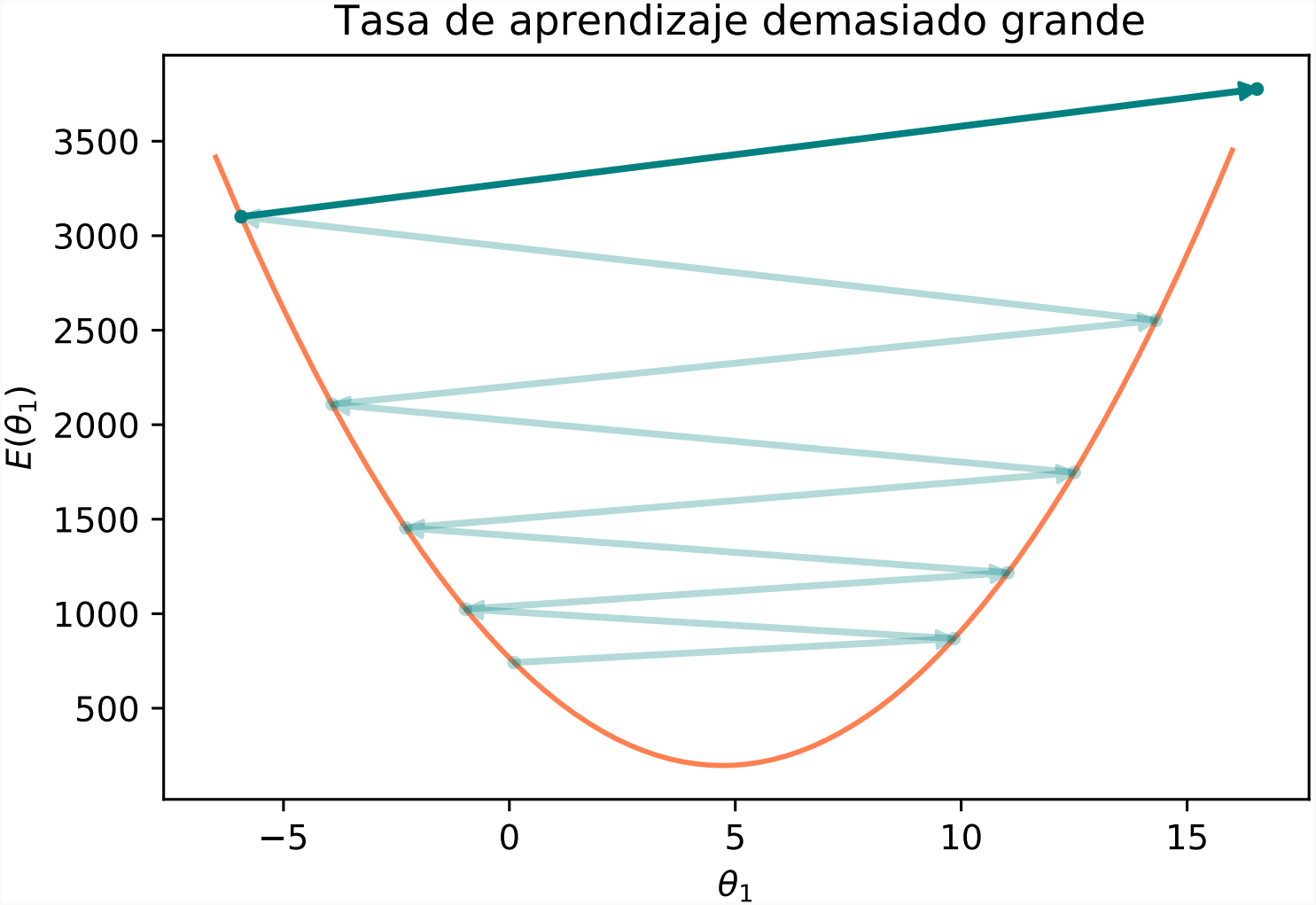


Ejemplo del descenso por gradiente (GD)

- Inicializando θ_1 con un valor mayor al que minimiza la función de pérdida

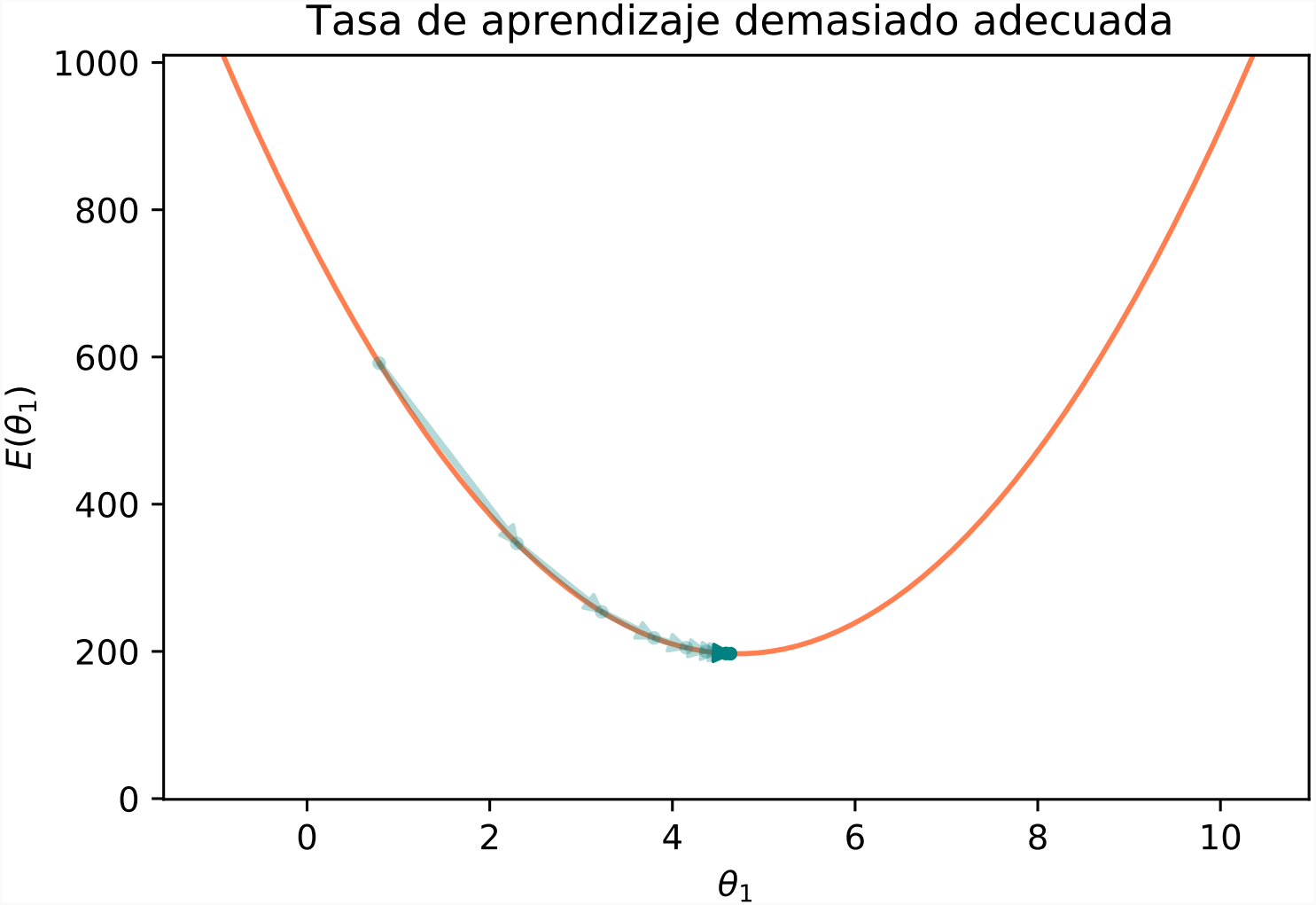


Sensibilidad a tasa de aprendizaje α

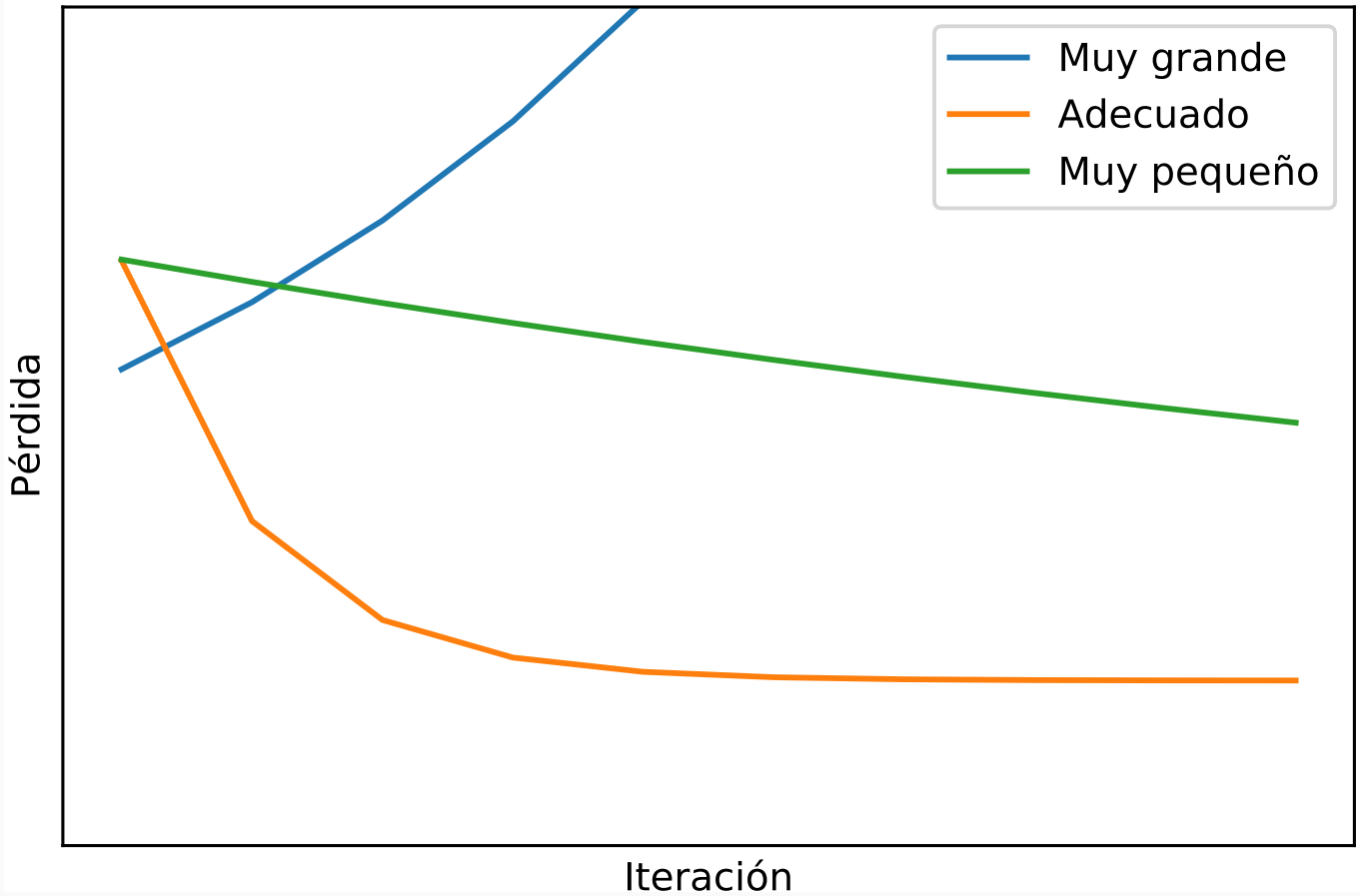


Sensibilidad a tasa de aprendizaje α

Sensibilidad a tasa de aprendizaje α



Sensibilidad a tasa de aprendizaje α



Escalando características

- **El problema**: los valores de las características pueden estar en rangos de valores muy diferentes

Escalando características

- **El problema**: los valores de las características pueden estar en rangos de valores muy diferentes
- **La estrategia**: Normalizar los rangos tal que todas las características contribuyan proporcionalmente a la distancia

Escalando características

- **El problema**: los valores de las características pueden estar en rangos de valores muy diferentes

- **La estrategia**: Normalizar los rangos tal que todas las características contribuyan proporcionalmente a la distancia

- **Diferentes métodos**:

$$x' = \frac{x - min(x_{1:n})}{max(x_{1:n}) - min(x_{1:n})} \quad \text{(Re-escalado)}$$

$$x' = \frac{x - \bar{x}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x^{(i)} - \bar{x})^2}} \quad \text{(Estandaziración)}$$

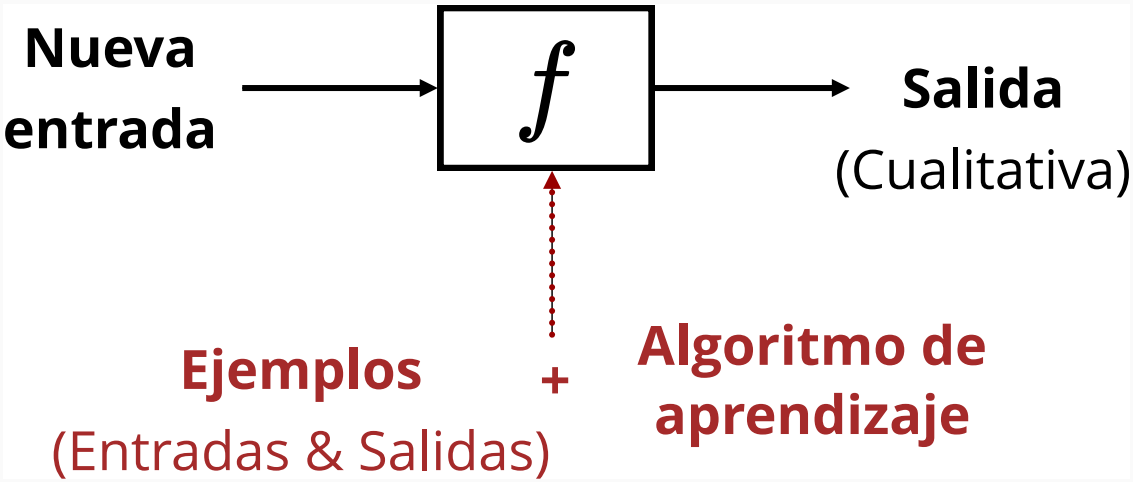
$$x' = \frac{x}{\|x\|} \quad \text{(Magnitud unitaria)}$$

Descenso por gradiente estocástico

- Aproximación estocástica de GD: estima $\nabla E(\boldsymbol{\theta}^{[t]})$ y actualiza parámetros con un subconjunto $\mathcal{B}$ de ejemplos de entrenamiento
  - $|\mathcal{B}|$ es un hiperparámetro
  - Es común dividir y ordenar aleatoriamente el conjunto de n ejemplos de entrenamiento en k minilotes ($|\mathcal{B}| \times k \approx n$)

Clasificación

- Salida discreta (cualitativa)
- Ejemplos: detección de spam, reconocimiento de rostros, etc.



Ejemplo de clasificación

- Clasificar sub-especias de la flor Iris basado en el ancho y largo de su pétalo

Ancho	Largo	Especie
1.4	0.2	Setosa
1.7	0.4	Setosa
1.5	0.1	Setosa
⋮	⋮	⋮
4.7	1.4	Versicolor
4.5	1.5	Versicolor
3.3	1.0	Versicolor
⋮	⋮	⋮

Características o atributo — Respuesta

Setosa

Versicolor

Tomada de `https://en.wikipedia.org/wiki/Iris_flower_data_set`

Clasificación: el caso binario

- En regresión lineal tenemos

$$P(y|\mathbf{x}, \boldsymbol{\theta}, \sigma^2) = \mathcal{N}(y|f_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\theta}), \sigma^2)$$

- ¿Cómo podemos extender este modelo para la clasificación binaria?

Clasificación: el caso binario

- En regresión lineal tenemos

$$P(y|\mathbf{x}, \boldsymbol{\theta}, \sigma^2) = \mathcal{N}(y|f_{\boldsymbol{\theta}}(\mathbf{x}, \boldsymbol{\theta}), \sigma^2)$$

- ¿Cómo podemos extender este modelo para la clasificación binaria?

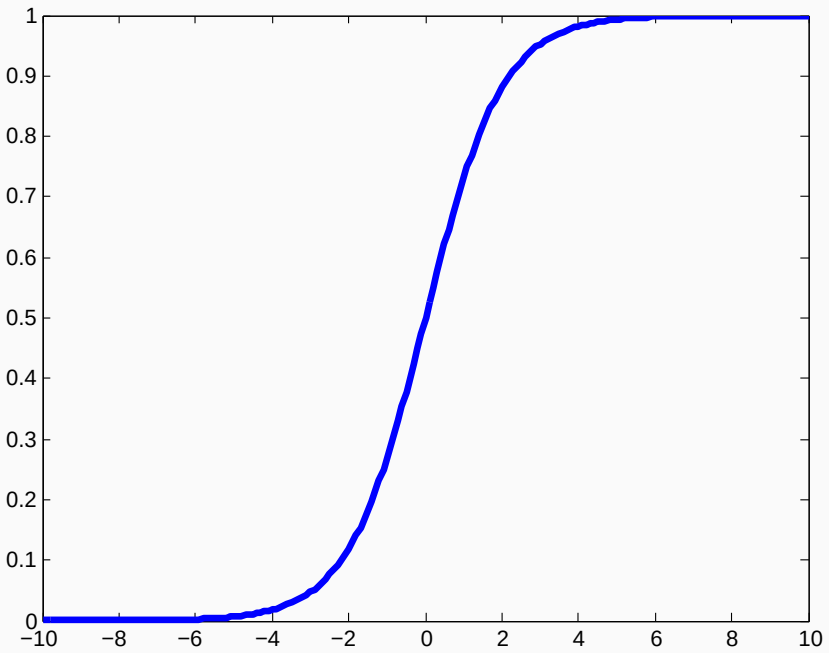
- Modelo de regresión logística

$$P(y|\mathbf{x}, \boldsymbol{\theta}) = Ber(y|sigm(\boldsymbol{\theta}^\top \mathbf{x}))$$

La función logística

- La función sigmoide o logística está dada por

$$sigm(z) = \frac{1}{1 + exp(-z)}$$



Estimador de máxima verosimilitud para regresión logística

- Tomando el negativo de la verosimilitud logarítmica

$$NVL(\boldsymbol{\theta}) = -\sum_{i=1}^{n} \{y^{(i)} \log q^{(i)} + (1 - y^{(i)}) \log(1 - q^{(i)})\} = E(\boldsymbol{\theta})$$

donde $E(\boldsymbol{\theta})$ se conoce como *entropía cruzada binaria* y $q^{(i)} = sigm(\boldsymbol{\theta}^\top \mathbf{x}^{(i)})$

Estimador de máxima verosimilitud para regresión logística

- Tomando el negativo de la verosimilitud logarítmica

$$NVL(\boldsymbol{\theta}) = -\sum_{i=1}^{n} \{y^{(i)} \log q^{(i)} + (1 - y^{(i)}) \log(1 - q^{(i)})\} = E(\boldsymbol{\theta})$$

  donde $E(\boldsymbol{\theta})$ se conoce como *entropía cruzada binaria* y $q^{(i)} = sigm(\boldsymbol{\theta}^\top \mathbf{x}^{(i)})$

- No hay solución cerrada, podemos entrenar usando descenso por gradiente

$$\nabla E(\boldsymbol{\theta}) = \sum_{i=1}^{n} (q^{(i)} - y^{(i)}) \cdot \mathbf{x}^{(i)} = \mathbf{X}^\top (\mathbf{q} - \mathbf{y})$$

Regularización en clasificación

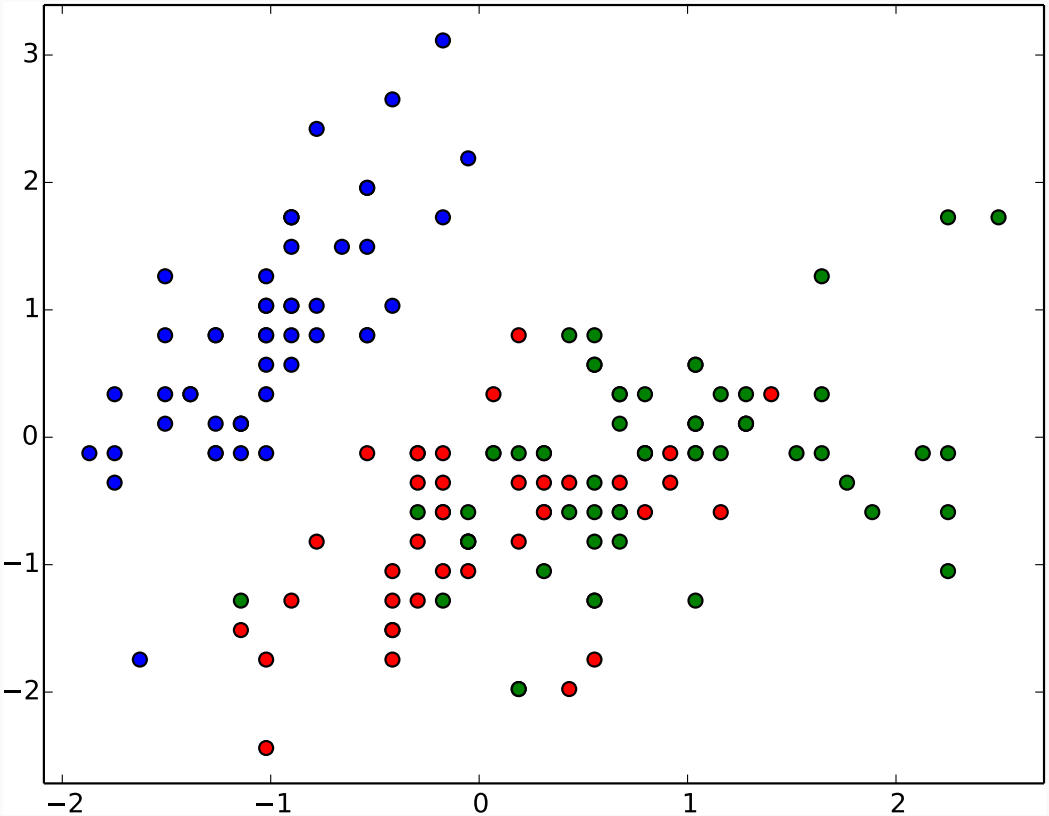
- Al igual que en regresión lineal la regularización puede ayudar a evitar el sobreajuste

- La función de error y el gradiente están dados por

$$\tilde{E}(\boldsymbol{\theta}) = E(\boldsymbol{\theta}) + \lambda \|\boldsymbol{\theta}\|_2^2$$
$$\nabla \tilde{E}(\boldsymbol{\theta}) = \nabla E(\boldsymbol{\theta}) + 2\lambda \boldsymbol{\theta}$$

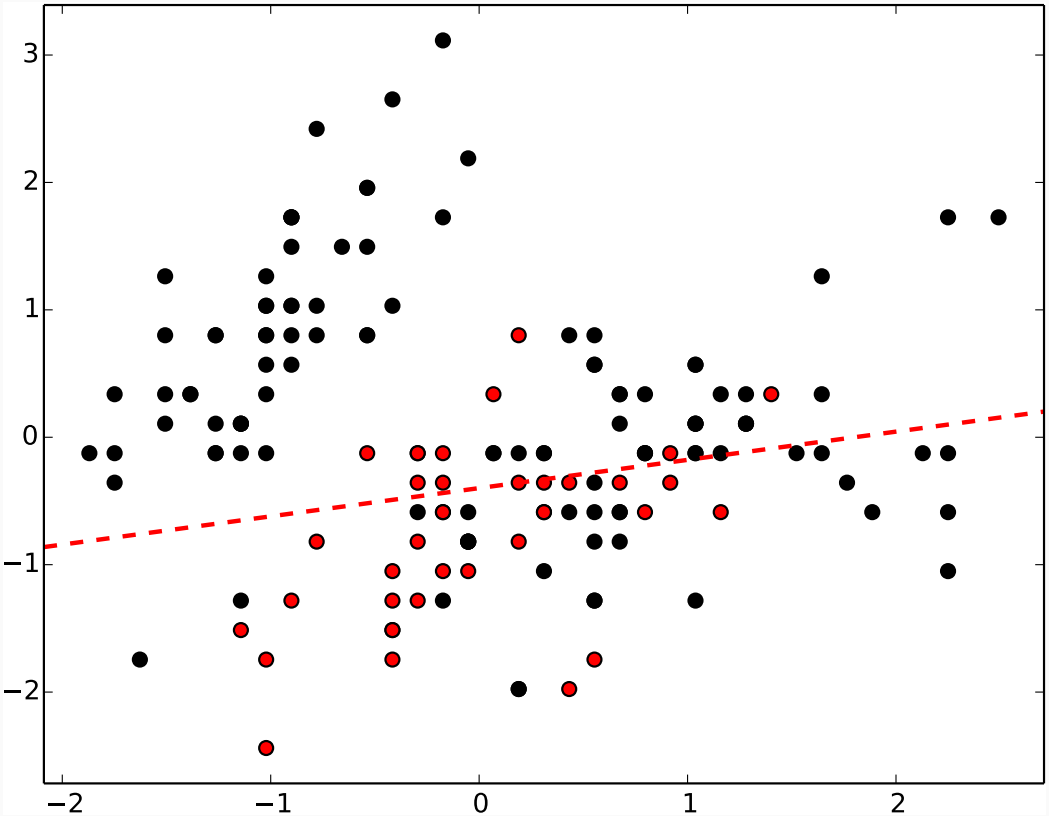
Clasificación multi-clase: uno vs el resto

- Un clasificador binario entre cada clase y el resto



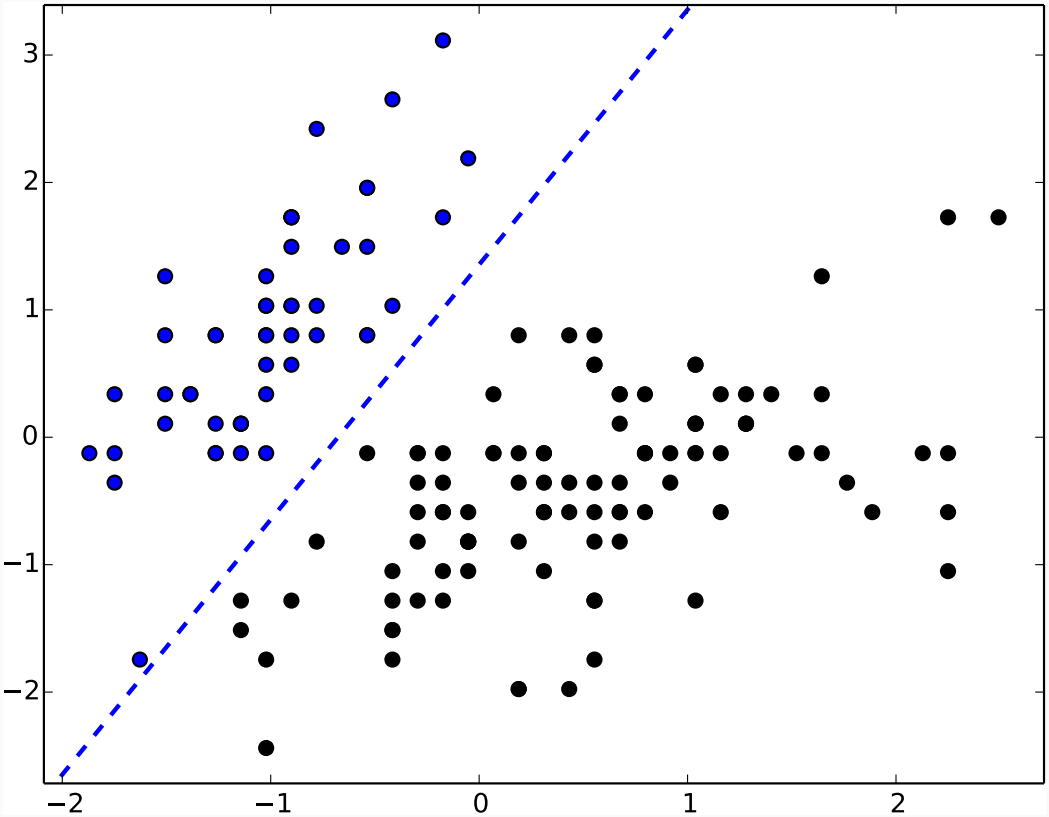
Clasificación multi-clase: uno vs el resto

- Un clasificador binario entre cada clase y el resto



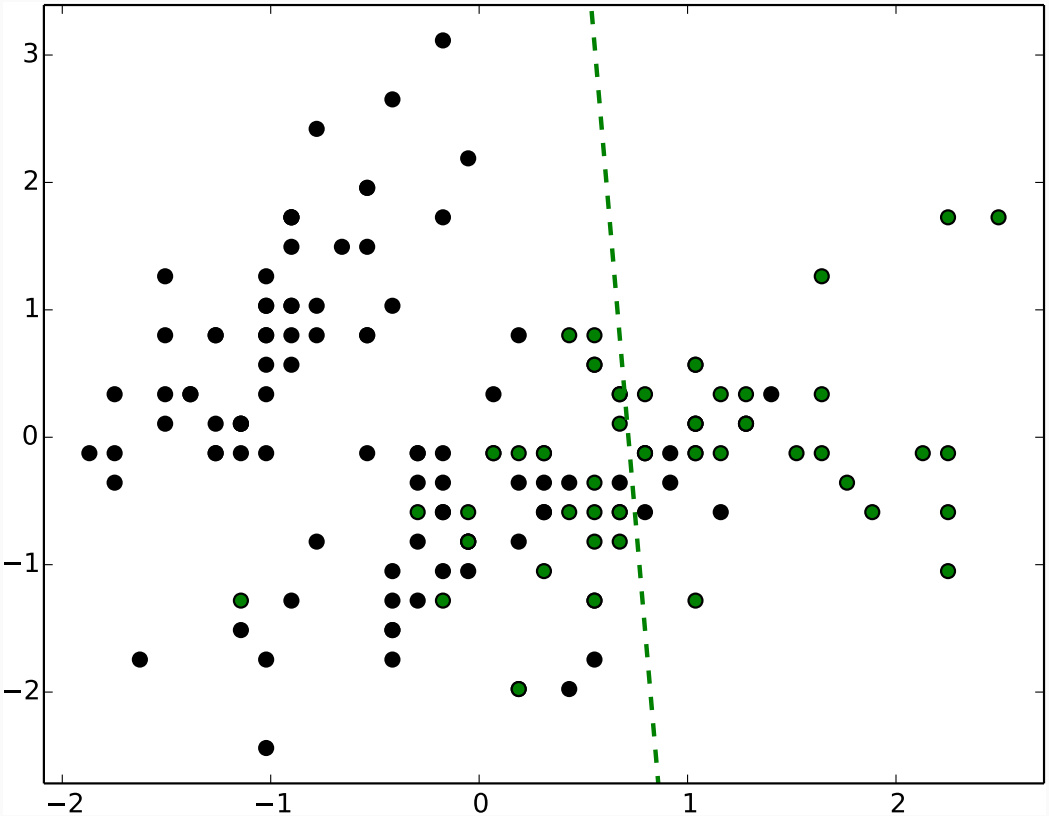
Clasificación multi-clase: uno vs el resto

- Un clasificador binario entre cada clase y el resto



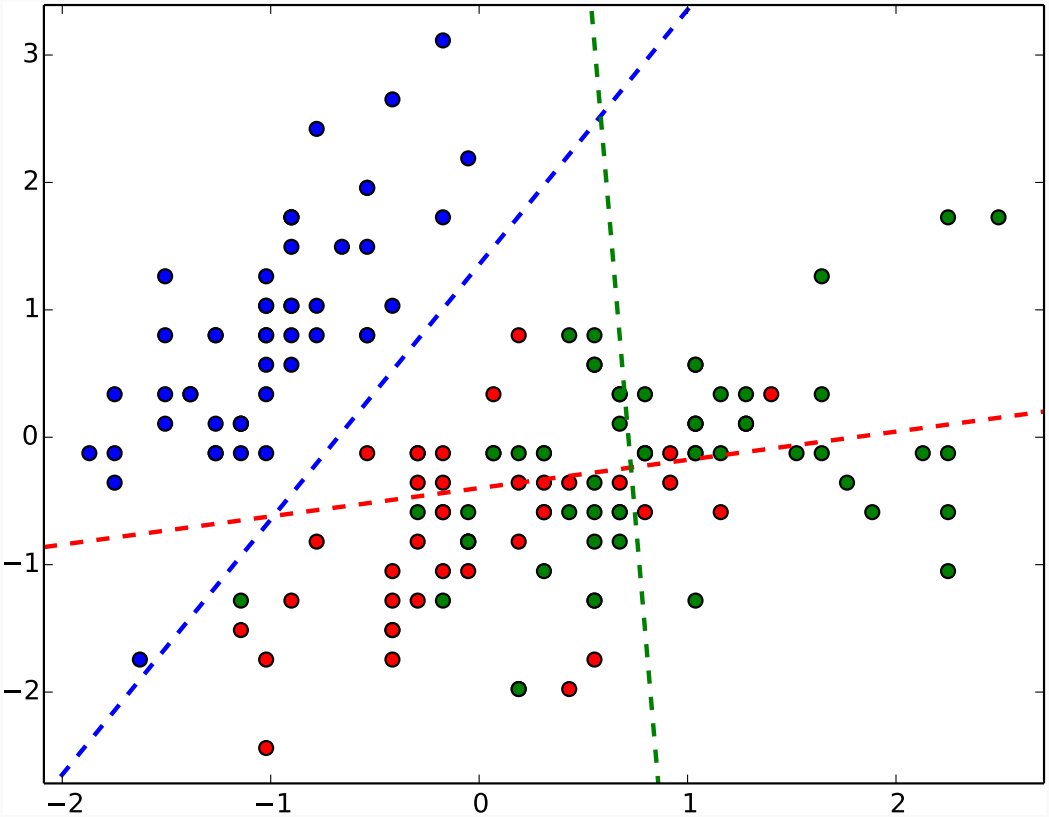
Clasificación multi-clase: uno vs el resto

- Un clasificador binario entre cada clase y el resto



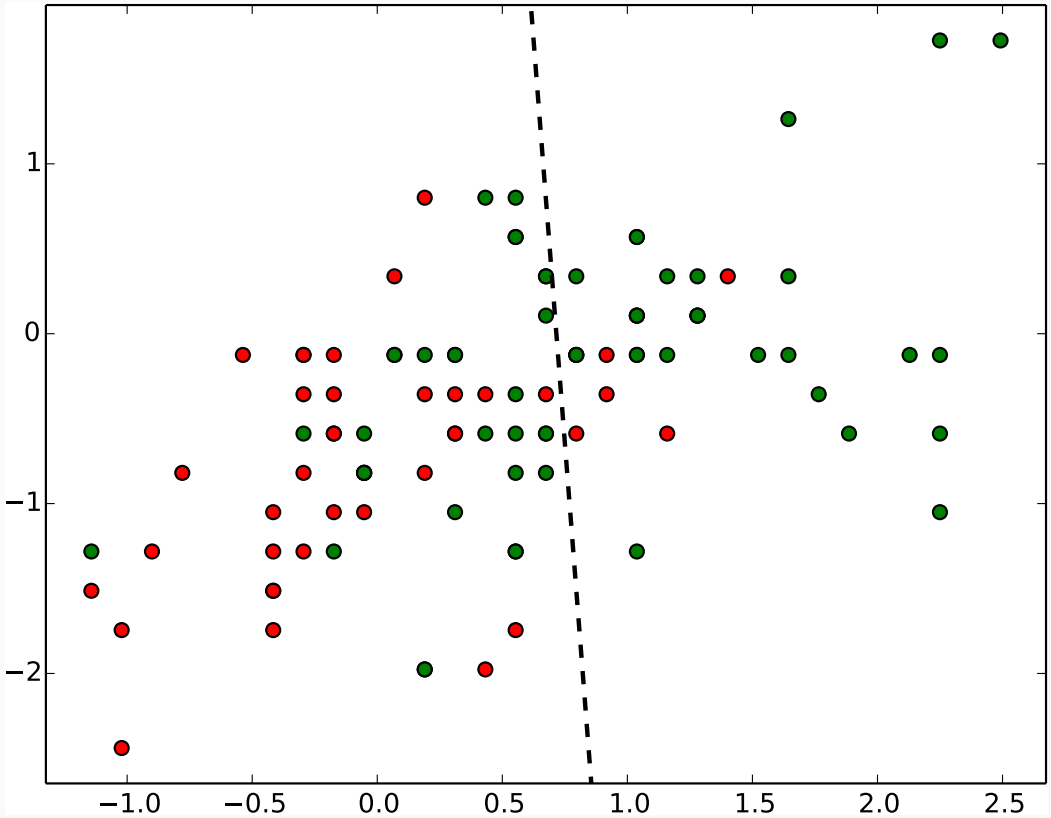
Clasificación multi-clase: uno vs el resto

- Un clasificador binario entre cada clase y el resto



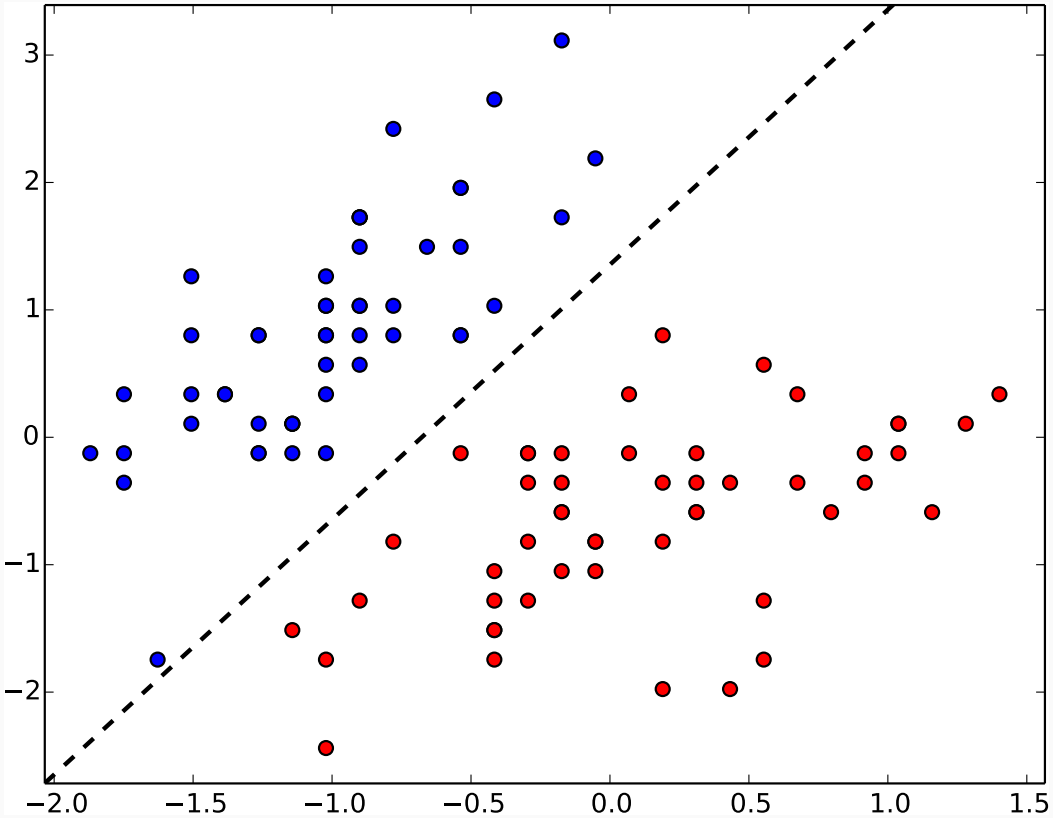
Clasificación multi-clase: uno vs uno

- Un clasificador binario entre cada par de clases



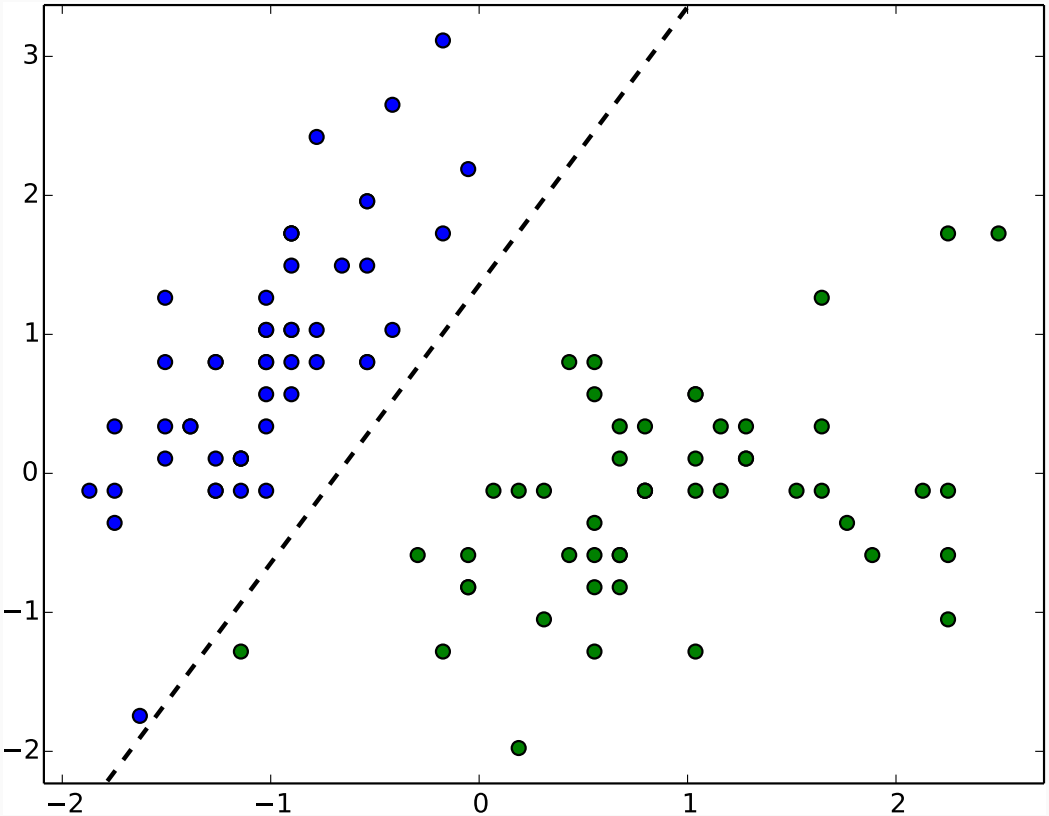
Clasificación multi-clase: uno vs uno

- Un clasificador binario entre cada par de clases



Clasificación multi-clase: uno vs uno

- Un clasificador binario entre cada par de clases



Clasificación multiclase: regresión logística multinomial

- Extensión de la regresión logística para múltiples clases

$$
\begin{aligned}
P(y|\mathbf{x}, \boldsymbol{\Theta}) &= Cat(y|softmax(\boldsymbol{\Theta}^\top \mathbf{x})_k) \\
&= \prod_{k=1}^{K} softmax(\boldsymbol{\Theta}^\top \mathbf{x})_k^{[y=k]}
\end{aligned}
$$

Clasificación multiclase: regresión logística multinomial

- Extensión de la regresión logística para múltiples clases

$$\begin{aligned} P(y|\mathbf{x}, \boldsymbol{\Theta}) &= Cat(y|softmax(\boldsymbol{\Theta}^\top \mathbf{x})_k) \\ &= \prod_{k=1}^{K} softmax(\boldsymbol{\Theta}^\top \mathbf{x})_k^{[y=k]} \end{aligned}$$

- donde $\mathbf{x} = [1, x_1, \ldots, x_d]$, $[y = k]$ son los corchetes de Iverson, $\boldsymbol{\Theta} \in \mathbb{R}^{d \times K}$, $\boldsymbol{\Theta}^\top \mathbf{x} \in \mathbb{R}^K$ y *softmax* es una generalización de la función logística

$$softmax(\mathbf{z})_k = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}} = \frac{e^{z_k - max(\mathbf{z})}}{\sum_{j=1}^{K} e^{z_j - max(\mathbf{z})}}$$

EMV para regresión logística multinomial

- Tomando el negativo de la verosimilitud logarítmica

$$NVL(\mathbf{\Theta}) = -\sum_{i=1}^{n}\sum_{k=1}^{K}[y^{(i)} = k] \log q_k^{(i)} = E(\mathbf{\Theta})$$

- donde

$$q_k^{(i)} = softmax(\mathbf{\Theta}^\top \mathbf{x}^{(i)})_k$$

- A $E(\mathbf{\Theta})$ se le como *entropía cruzada categórica*.

EMV para regresión logística multinomial

- Tomando el negativo de la verosimilitud logarítmica

$$NVL(\mathbf{\Theta}) = -\sum_{i=1}^{n}\sum_{k=1}^{K}[y^{(i)} = k] \log q_k^{(i)} = E(\mathbf{\Theta})$$

- donde

$$q_k^{(i)} = softmax(\mathbf{\Theta}^\top \mathbf{x}^{(i)})_k$$

- A $E(\mathbf{\Theta})$ se le como *entropía cruzada categórica*.

- Podemos entrenar modelos usando descenso por gradiente

$$\nabla E(\boldsymbol{\theta})_k = \sum_{i=1}^{n}(q_k^{(i)} - [y^{(i)} = k]) \cdot \mathbf{x}^{(i)}$$

¿Cómo representamos múltiples clases?

- **Sólo un valor**: se representa por una variable discreta y que puede tomar los valores $1, \ldots, K$. Por ej. si tenemos 4 clases, representamos la clase 2 por $y = 2$

¿Cómo representamos múltiples clases?

- **Sólo un valor**: se representa por una variable discreta y que puede tomar los valores $1, \ldots, K$. Por ej. si tenemos 4 clases, representamos la clase 2 por $y = 2$

- **1-de-K**: cada clase se representa por un vector binario $\mathbf{y}$ de K dimensiones con 1 sólo en la posición de la clase. Siguiendo el mismo ejemplo tenemos

$$\mathbf{y} = [0, 1, 0, 0]$$

Modelos generativos

- Modelan la probabilidad conjunta de los entradas y las salidas $P(\mathbf{x}, \mathbf{y}) = P(\mathbf{x}|\mathbf{y})P(\mathbf{y})$.
- La probabilidad condicional de las salidas dadas las entradas $P(\mathbf{y}|\mathbf{x})$ se obtiene a partir de la probabilidad conjunta.
- Ejemplos: clasificador bayesiano ingenuo, redes bayesianas, HMMs, etc.

Modelos discriminativos

- Modelan directamente la probabilidad condicional de las salidas dadas las entradas $P(\mathbf{y}|\mathbf{x})$.

- Ejemplos: regresión logística, SVMs, etc.

Generativos vs distriminativos: Entrenamiento

- **Generativo**: algunos modelos requieren sólo contar y promediar.
- **Discriminativo**: usualmente requieren resolver problemas de optimización convexo.

Generativos vs distriminativos: nuevas clases

- **Generativo**: las clases se entrenan por separado, por lo que no es necesario volver a entrenar si agregamos una nueva clase.

- **Discriminativo**: requiere volver a entrenar el modelo completo si agregamos una nueva clase.

Generativos vs distriminativos: datos faltantes

- **Generativo**: podemos ignorar datos faltantes en la etapa de prueba y calcular la probabilidad a posteriori con los disponibles.

- **Discriminativo**: no tienen una forma natural de lidiar con datos faltantes.

Generativos vs distriminativos: datos no etiquetados

- **Generativo**: es sencillo de incorporar datos no etiquetados (aprendizaje semi-supervisado).
- **Discriminativo**: difícil de incorporar datos no etiquetados.

Generativos vs distriminativos: simetría en entradas y salidas

- **Generativo**: es posible inferir entradas posibles dadas ciertas salidas.
- **Discriminativo**: no es posible inferir entradas posibles dadas ciertas salidas.

Generativos vs distriminativos: expansión por bases

- **Generativo**: difícil de incorporar debido a dependencias.
- **Discriminativo**: es fácil modelar entradas expandidas.

Generativos vs distriminativos: calibración de probabilidades

- **Generativo**: algunos modelos hacen presuposiciones de independencia que no se cumplen y esto puede hacer que las probabilidades estén en los extremos (cerca de 0 o 1).

- **Discriminativo**: usualmente mejor calibradas.