Aprendizaje automatizado

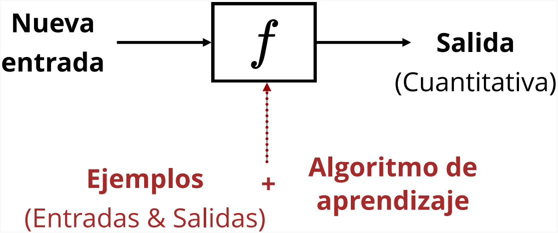
Métodos lineales de regresión y clasificación

Gibran Fuentes Pineda

Febrero 2020

Regresión

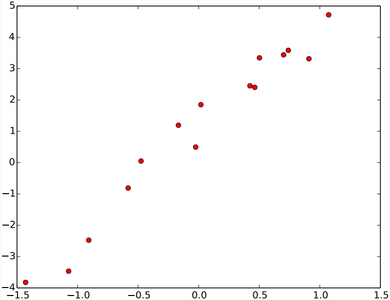
- Salida contínua (cuantitativa)
- Ejemplos: predicción de temperatura de un cuarto, etc.



Prediciendo el precio de casas

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

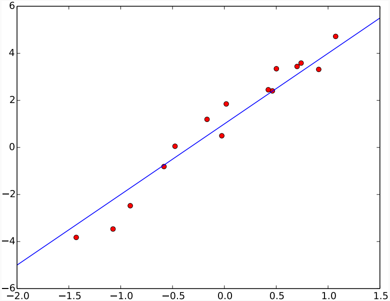
Tamaño (m^2)	Precio (USD)
489.59	489.59
556.08	556.08
570.35	570.35
772.84	772.84
970.95	970.95
1162.00	1162.00
1263.10	1263.10
$\vdots$	$\vdots$



Prediciendo el precio de casas

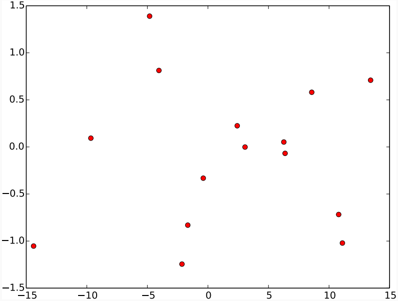
- Podemos hacer presuposiciones sobre f, por ejemplo que la relación es lineal:

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



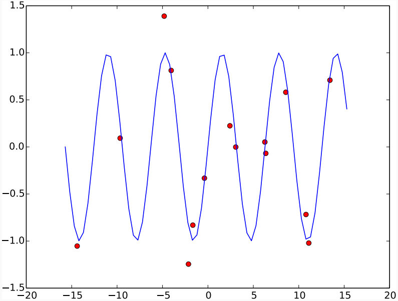
Modelando relaciones no lineales

- ¿Qué función se ajusta a estos datos?



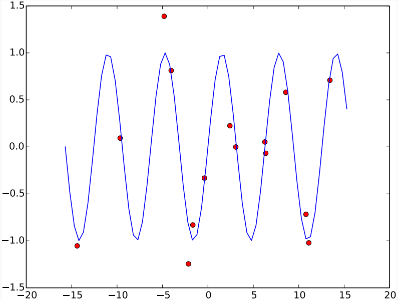
Modelando relaciones no lineales

- ¿Qué tal una función seno?



- Podemos usar polinomios para aproximarla

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



Regresión lineal

- Modelo lineal

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

Regresión lineal

- Modelo lineal

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

- Con expansión de funciones base

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

Regresión lineal

- Modelo lineal

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

- Con expansión de funciones base

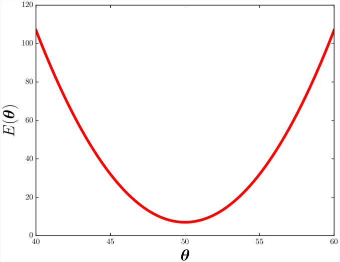
$$f_{\boldsymbol{\theta}}(\mathsf{x}, \boldsymbol{\theta}) = \boldsymbol{\theta}^T \phi(\mathsf{x}) = \sum_{i=1}^{d} \theta_i \cdot \phi(\mathsf{x})_i$$

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

¿Cómo medimos la calidad del ajuste?

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

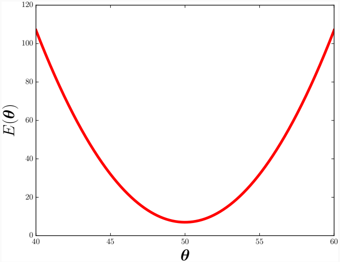
$$E(\boldsymbol{\theta}) = \frac{1}{2}\sum_{i=1}^{n}\{f_{\boldsymbol{\theta}}(\mathrm{x}^{(i)}, \boldsymbol{\theta}) - y^{(i)}\}^2$$



¿Cómo medimos la calidad del ajuste?

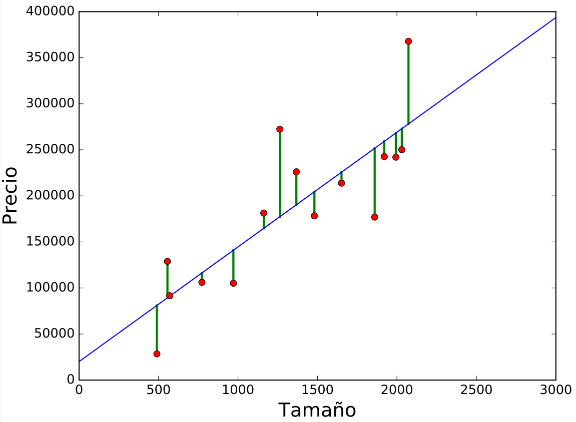
- Definimos una función de error, por ejemplo la suma de errores cuadráticos:

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

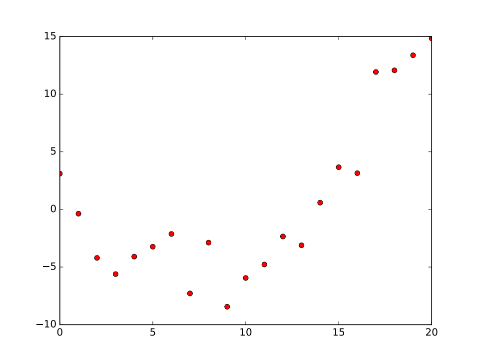


- Objetivo: encontrar el valor de $\boldsymbol{\theta}$ que minimice $E(\boldsymbol{\theta})$

¿Cómo medimos la calidad del ajuste?



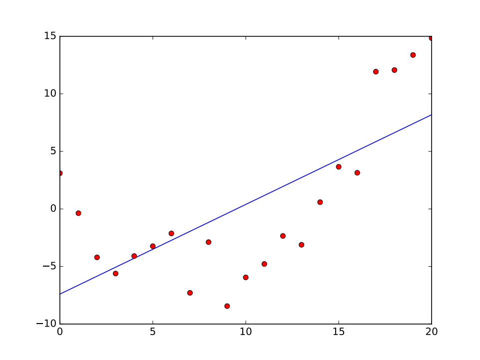
¿Qué grado del polinomio es adecuado?



¿Qué grado del polinomio es adecuado?

- Podemos usar uno lineal nuevamente

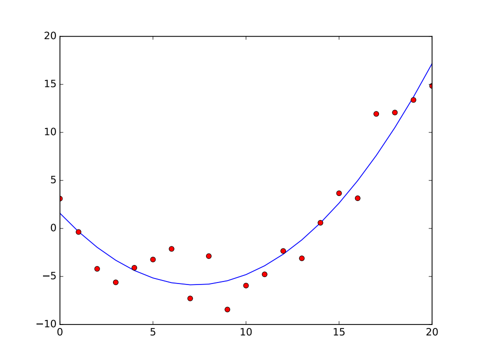
$$f_{\boldsymbol{\theta}}(x) = \theta_0 + \theta_1 \cdot x$$



¿Qué grado del polinomio es adecuado?

- O uno cuadrático

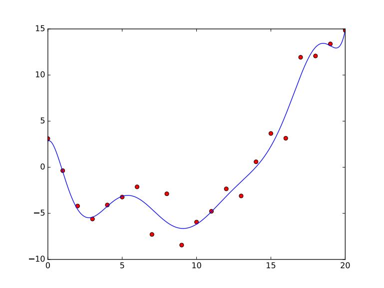
$$f_{\boldsymbol{\theta}}(x) = \theta_0 + \theta_1 \cdot x + \theta_2 \cdot x^2$$



¿Qué grado del polinomio es adecuado?

- ¿Qué tal uno de grado 10?

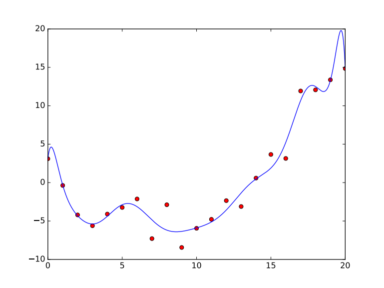
$$f_{\boldsymbol{\theta}}(x) = \theta_0 + \theta_1 + \theta_2 \cdot x^2 + \cdot x + \cdots + \theta_{10} \cdot x^{10}$$



¿Qué grado del polinomio es adecuado?

- Grado 14

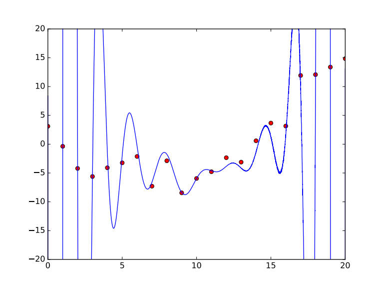
$$f_{\boldsymbol{\theta}}(x) = \theta_0 + \theta_1 + \theta_2 \cdot x^2 + \cdot x + \cdots + \theta_{14} \cdot x^{14}$$



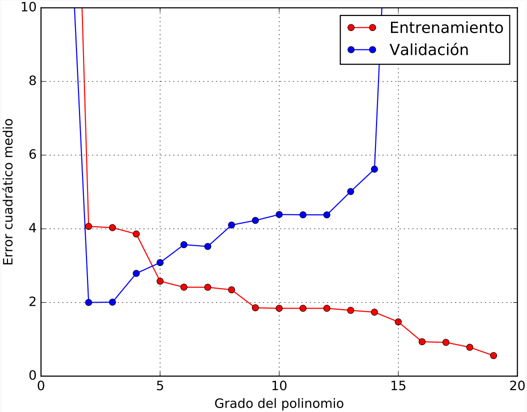
¿Qué grado del polinomio es adecuado?

- O grado 20

$$f_{\boldsymbol{\theta}}(x) = \theta_0 + \theta_1 + \theta_2 \cdot x^2 + \cdot x + \cdots + \theta_{20} \cdot x^{20}$$



El problema de la generalización



¿Por qué está sobreajustando?

	$d = 0$	$d = 1$	$d = 3$	$d = 9$
θ_0	0.19	0.82	0.31	0.35
θ_1		-1.27	7.99	232.37
θ_2			-25.43	-5321.83
θ_3			17.37	48568
θ_4				-231639.30
θ_5				640042.26
θ_6				-1061800.52
θ_7				1042400.18
θ_8				-557682.99
θ_9				125201.43

¿Cómo evito el sobreajuste?

- Penalizando parámetros con valores grandes

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

- λ determina el peso dado a la penalización

¿Cómo evito el sobreajuste?

	$\log \lambda = -\infty$	$\log \lambda = -18$	$\log \lambda = 0$
θ_0	0.35	0.35	0.13
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

Interpretación probabilística

- Asumiendo ruido ϵ con distribución normal en el modelo

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

Interpretación probabilística

- Asumiendo ruido ϵ con distribución normal en el modelo

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

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

$$P(\hat{y}|\mathsf{x}, \boldsymbol{\theta}, \sigma^2) = \mathcal{N}(\hat{y}|\theta_0 + \boldsymbol{\theta}_{1:d}^T \phi(\mathsf{x}), \sigma^2)$$

Interpretación probabilística

- Asumiendo ruido ϵ con distribución normal en el modelo

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

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

$$P(\hat{y}|\mathsf{x}, \boldsymbol{\theta}, \sigma^2) = \mathcal{N}(\hat{y}|\theta_0 + \boldsymbol{\theta}_{1:d}^T \phi(\mathsf{x}), \sigma^2)$$

- $\phi(\mathsf{x})$ es una función base (por ej. polinomial)

Obteniendo el estimador de máxima verosimilitud

- Se busca minimizar el negativo de la verosimilitud logarítmica

$$\begin{aligned}
NVL(\boldsymbol{\theta}) &= -\sum_{i=1}^{n} \log P(\hat{y}^{(i)}|\mathsf{x}^{(i)}, \boldsymbol{\theta}) \\
&= -\sum_{i=1}^{n} \log \mathcal{N}(\hat{y}^{(i)}|\theta_0 + \boldsymbol{\theta}_{1:d}^{T}\phi(\mathsf{x}^{(\mathrm{i})}), \sigma^2) \\
&= -\frac{1}{2\sigma^2}\sum_{i=1}^{n}(\theta_0 + \boldsymbol{\theta}_{1:d}^{T}\phi(\mathsf{x}^{(i)}) - \hat{y}^{(i)})^2 - \frac{n}{2} log 2\pi\sigma^2
\end{aligned}$$

Obteniendo el estimador de máxima verosimilitud

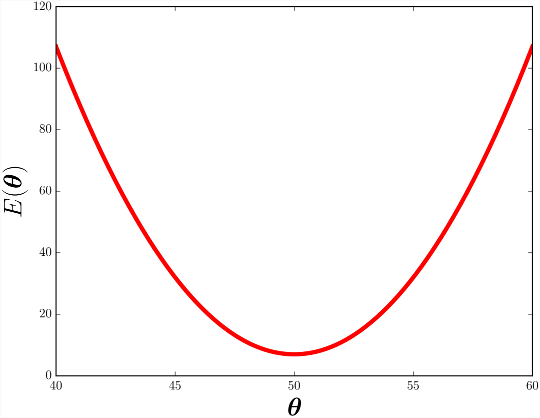
- Se busca minimizar el negativo de la verosimilitud logarítmica

$$
\begin{aligned}
NVL(\boldsymbol{\theta}) &= -\sum_{i=1}^{n} \log P(\hat{y}^{(i)}|\mathsf{x}^{(i)}, \boldsymbol{\theta}) \\
&= -\sum_{i=1}^{n} \log \mathcal{N}(\hat{y}^{(i)}|\theta_0 + \boldsymbol{\theta}_{1:d}^{T}\phi(\mathsf{x}^{(\mathrm{i})}), \sigma^2) \\
&= -\frac{1}{2\sigma^2}\sum_{i=1}^{n}(\theta_0 + \boldsymbol{\theta}_{1:d}^{T}\phi(\mathsf{x}^{(i)}) - \hat{y}^{(i)})^2 - \frac{n}{2} log 2\pi\sigma^2
\end{aligned}
$$

- Equivalente a minimizar suma de errores cuadráticos

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

Función de error para mínimos cuadrados



Obteniendo el estimador de máxima verosimilitud

- Reformulando NVL

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

Obteniendo el estimador de máxima verosimilitud

- Reformulando NVL

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

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

$$\mathrm{X}^T\mathrm{X}\boldsymbol{\theta} = \mathrm{X}^T\mathrm{y}$$

Obteniendo el estimador de máxima verosimilitud

- Reformulando NVL

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

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

$$\mathrm{X}^T\mathrm{X}\boldsymbol{\theta} = \mathrm{X}^T\mathrm{y}$$

- El estimador de máxima verosimilitud es

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

¿Y si tenemos múltiples variables de salida?

- Solución de mínimos cuadrados

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

- Equivalente a

$$\hat{\theta}_{k\,EMV} = (X^T X)^{-1} X^T y_k$$

Obteniendo el estimador de máximo a posteriori

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

$$\hat{\boldsymbol{\theta}}_{MAP} = \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{n} log\mathcal{N}(\hat{y}^{(i)}|\theta_0 + \boldsymbol{\theta}^T\phi(\mathrm{x}^{(i)}), \sigma^2)$$

$$+ \sum_{j=0}^{d} log\mathcal{N}(\theta_j|0, \tau^2)$$

Obteniendo el estimador de máximo a posteriori

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

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

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

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

Obteniendo el estimador de máximo a posteriori

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

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

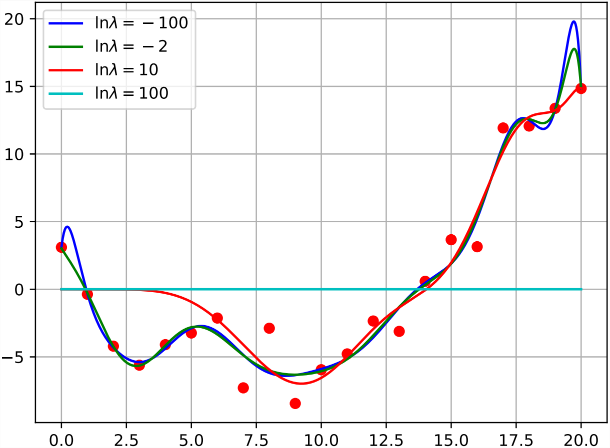
- Equivalente a minimizar suma de errores cuadráticos con los parámeros penalizados con la norma ℓ_2

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

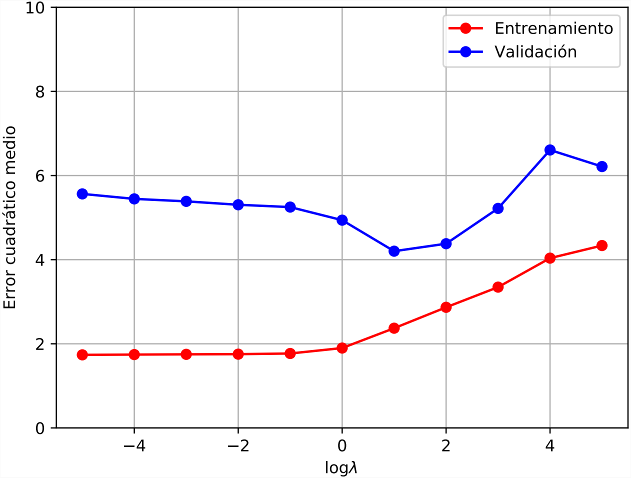
- Derivando $\tilde{E}(\boldsymbol{\theta})$ con respecto a $\boldsymbol{\theta}$ e igualando a cero

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

Mínimos cuadrados penalizados



Mínimos cuadrados penalizados



Regularización con norma ℓ_1

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

$$\hat{\boldsymbol{\theta}}_{\text{LASSO}} = \arg\min_{\boldsymbol{\theta}} \{ \frac{1}{2} \sum_{i=1}^{n} \{ f_{\boldsymbol{\theta}}(\mathrm{x}^{(i)}, \boldsymbol{\theta}) - y^{(i)} \}^2 + \frac{\lambda}{2} \|\boldsymbol{\theta}\|_1 \}$$

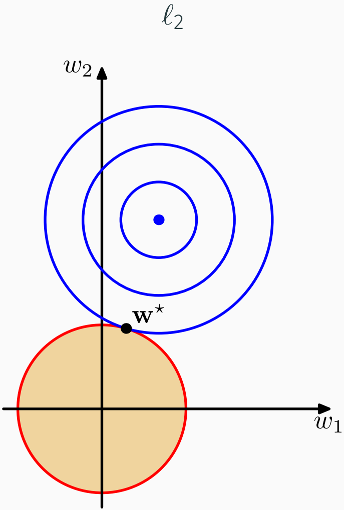
Regularización con norma ℓ_1

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

$$\hat{\boldsymbol{\theta}}_{\text{LASSO}} = \arg\min_{\boldsymbol{\theta}} \{ \frac{1}{2} \sum_{i=1}^{n} \{ f_{\boldsymbol{\theta}}(\mathrm{x}^{(i)}, \boldsymbol{\theta}) - y^{(i)} \}^2 + \frac{\lambda}{2} \|\boldsymbol{\theta}\|_1 \}$$

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

Regularización con diferentes normas



Método alternativo: descenso por gradiente

- Modifica parámetros iterativamente de acuerdo al gradiente de la función de suma de errores cuadráticos

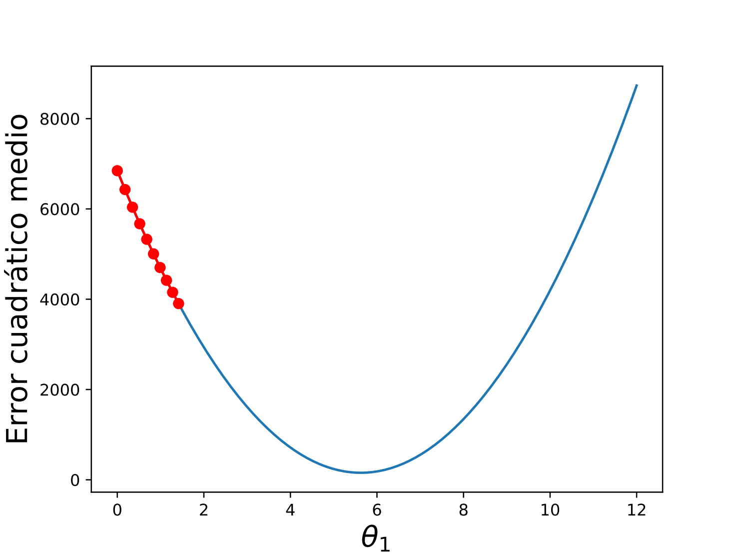
$$\boldsymbol{\theta}^{\{k+1\}} = \boldsymbol{\theta}^{\{k\}} - \alpha \mathbf{g}(\boldsymbol{\theta}^{\{k\}})$$

- donde

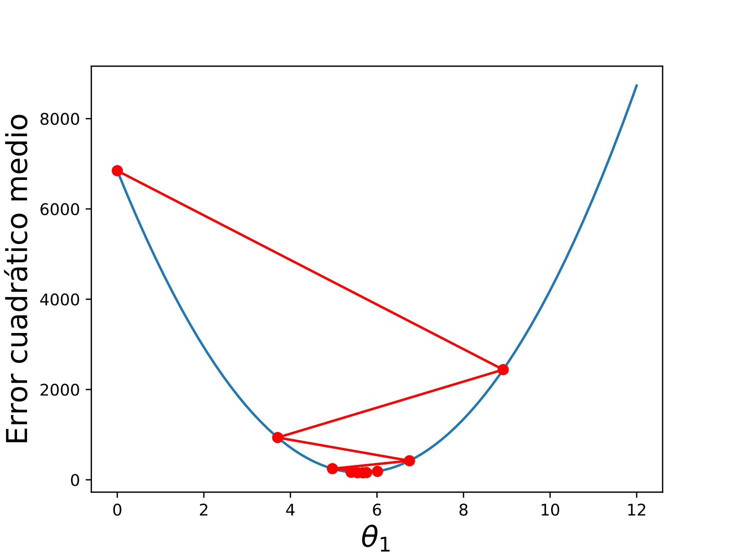
$$\mathbf{g}(\boldsymbol{\theta}) = \nabla E(\boldsymbol{\theta}) = \nabla \left[\frac{1}{2} \sum_{i=1}^{n} \{ f_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \boldsymbol{\theta}) - y^{(i)} \}^2 \right] = \mathbf{X}^T (f_{\boldsymbol{\theta}}(\mathbf{X}, \boldsymbol{\theta}) - \mathbf{y})$$

- 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

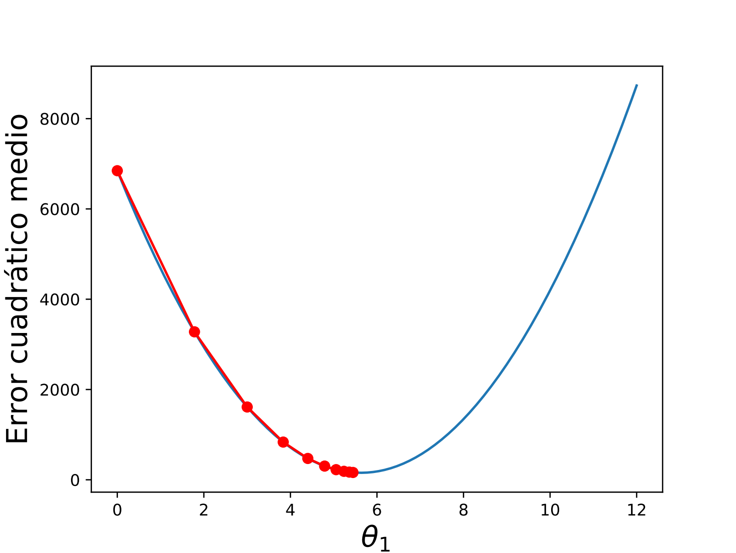
¿Qué tasa de aprendizaje usamos?



¿Qué tasa de aprendizaje usamos?



¿Qué tasa de aprendizaje usamos?



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**:

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

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

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

Descenso por gradiente estocástico

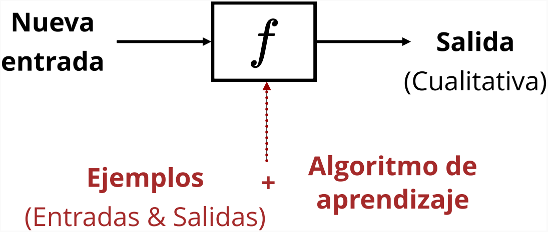
- Se nos presentan los ejemplos uno tras otro

$$\boldsymbol{\theta}^{\{k+1\}} = \boldsymbol{\theta}^{\{k\}} + \alpha \mathbf{g}(\boldsymbol{\theta}^{\{k\}})$$

- Se repite la actualización por cada nuevo dato
- Puede escapar de mínimos locales debido a que incorpora de alguna forma "ruido"

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

$$P(\hat{y}|\mathsf{x}, \boldsymbol{\theta}) = \mathcal{N}(\hat{y}|\mu(\mathsf{x}), \sigma^2(\mathsf{x}))$$

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

Clasificación: el caso binario

- En regresión lineal tenemos

$$P(\hat{y}|\mathrm{x}, \boldsymbol{\theta}) = \mathcal{N}(\hat{y}|\mu(\mathrm{x}), \sigma^2(\mathrm{x}))$$

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

- Modelo de regresión logística

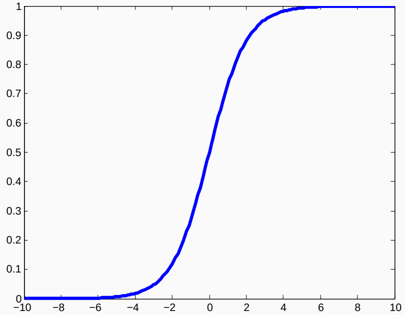
$$P(\hat{y}|\mathrm{x}, \boldsymbol{\theta}) = Ber(\hat{y}|\mu(\mathrm{x}, \boldsymbol{\theta}))$$

$$\mu(\mathrm{x}, , \boldsymbol{\theta}) = sigm(\boldsymbol{\theta}^T \mathrm{x})$$

La función logística

- La función sigmoidal 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

$$
\begin{aligned}
NVL(\boldsymbol{\theta}) &= -\sum_{i=1}^{n} \log\{p^{(i)y^{(i)}}(1 - p^{(i)})^{1-y^{(i)}}\} \\
&= -\sum_{i=1}^{n}\{y^{(i)} \log p^{(i)} + (1 - y^{(i)}) \log(1 - p^{(i)})\} = E(\boldsymbol{\theta})
\end{aligned}
$$

- donde $E(\boldsymbol{\theta})$ se conoce como **entropía cruzada categótica** y

$$p^{(i)} = sigm(\boldsymbol{\theta}^T \mathrm{x})$$

Estimador de máxima verosimilitud para regresión logística

- Tomando el negativo de la verosimilitud logarítmica

$$\begin{aligned} NVL(\boldsymbol{\theta}) &= -\sum_{i=1}^{n} \log\{p^{(i)y^{(i)}}(1 - p^{(i)})^{1-y^{(i)}}\} \\ &= -\sum_{i=1}^{n}\{y^{(i)} \log p^{(i)} + (1 - y^{(i)}) \log(1 - p^{(i)})\} = E(\boldsymbol{\theta}) \end{aligned}$$

- donde $E(\boldsymbol{\theta})$ se conoce como **entropía cruzada categótica** y

$$p^{(i)} = sigm(\boldsymbol{\theta}^T \mathrm{x})$$

- No hay solución cerrada, podemos usar descenso por gradiente

$$\mathrm{g}(\boldsymbol{\theta}) = \frac{d}{d\boldsymbol{\theta}} E(\boldsymbol{\theta}) = \sum_{i=1}^{n} (p^{(i)} - y^{(i)})\mathrm{x}^{(i)} = \mathrm{X}^T(\mathrm{p} - \mathrm{y})$$

Regularización en clasificación

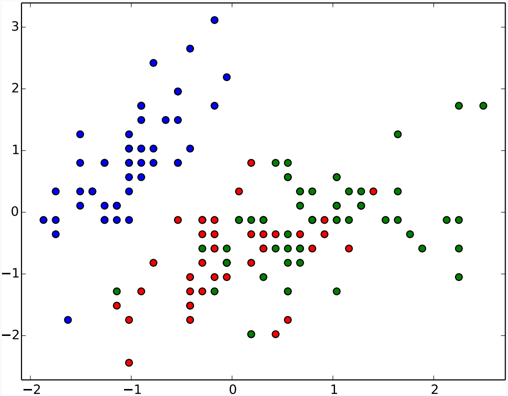
- Al igual que en regresión lineal la regularización puede ayudar a evitar el sobreajuste

- La función de error, el gradiente y el Hessiano están dados por

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

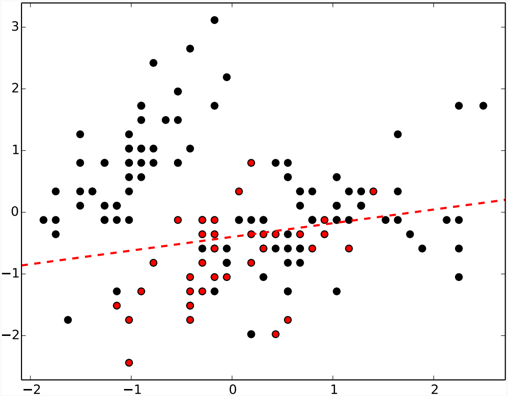
Clasificación multi-clase: uno vs el resto

- Un clasificador binario entre cada clase y el resto



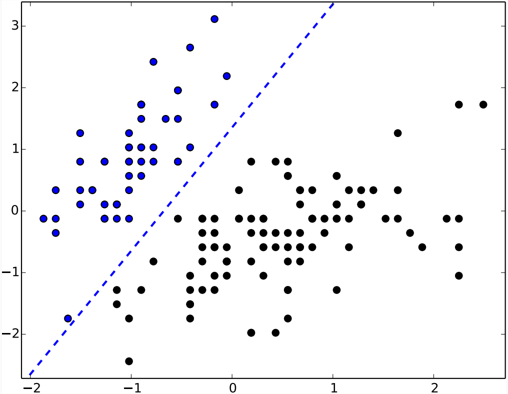
Clasificación multi-clase: uno vs el resto

- Un clasificador binario entre cada clase y el resto



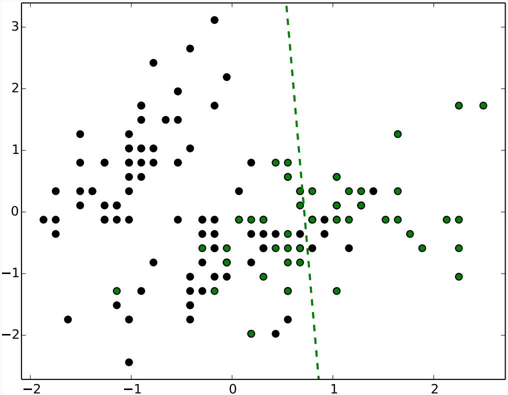
Clasificación multi-clase: uno vs el resto

- Un clasificador binario entre cada clase y el resto



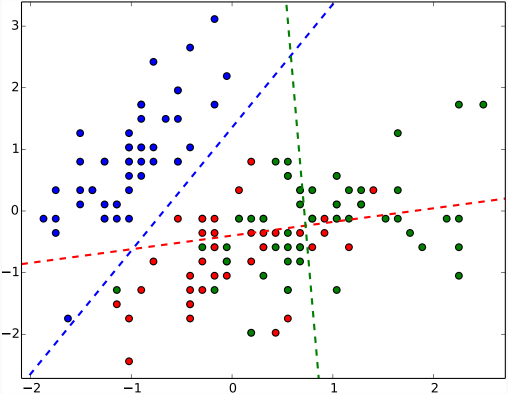
Clasificación multi-clase: uno vs el resto

- Un clasificador binario entre cada clase y el resto



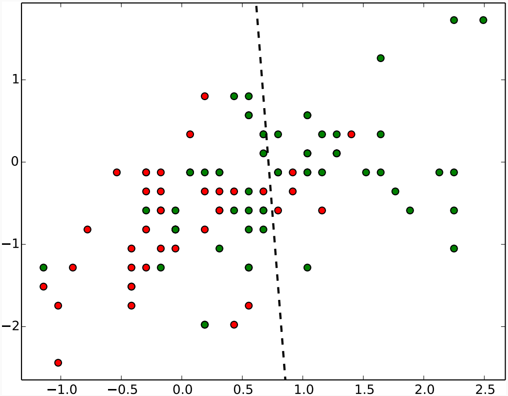
Clasificación multi-clase: uno vs el resto

- Un clasificador binario entre cada clase y el resto



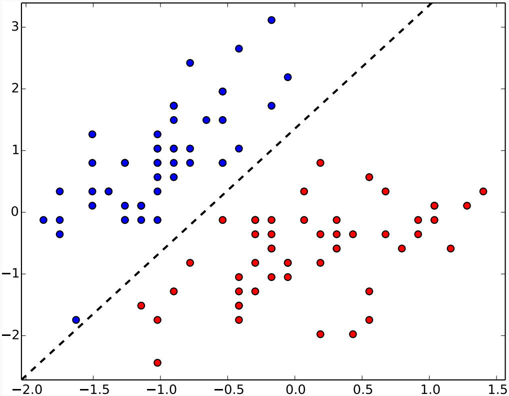
Clasificación multi-clase: uno vs uno

- Un clasificador binario entre cada par de clases



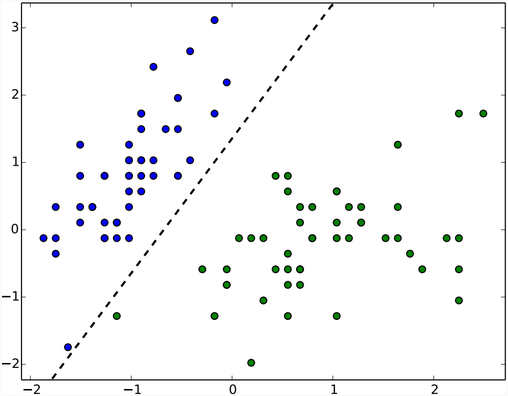
Clasificación multi-clase: uno vs uno

- Un clasificador binario entre cada par de clases



Clasificación multi-clase: uno vs uno

- Un clasificador binario entre cada par de clases



Clasificación multi-clase: regresión logística multinomial o softmax

- Extensión de la regresión logística para múltiples clases

$$P(y|\mathbf{x}, \mathbf{\Theta}) = Cat(y|\mu(\mathbf{x}, \mathbf{\Theta})) = \prod_{k=1}^{K} \mu(\mathbf{x}, \mathbf{\Theta})_k^{[y=k]}$$

$$\mu(\mathbf{x}, \mathbf{\Theta})_k = softmax(\mathbf{\Theta}^T \mathbf{x})_k, \mathbf{x} = [1, x_1, \ldots, x_d]$$

(entropía cruzada categótica)

Clasificación multi-clase: regresión logística multinomial o softmax

- Extensión de la regresión logística para múltiples clases

$$P(y|\mathbf{x}, \mathbf{\Theta}) = Cat(y|\mu(\mathbf{x}, \mathbf{\Theta})) = \prod_{k=1}^{K} \mu(\mathbf{x}, \mathbf{\Theta})_k^{[y=k]}$$

$$\mu(\mathbf{x}, \mathbf{\Theta})_k = softmax(\mathbf{\Theta}^T \mathbf{x})_k, \mathbf{x} = [1, x_1, \ldots, x_d]$$

(**entropía cruzada categótica**)

- donde $\mathbf{\Theta} \in \mathbb{R}^{d \times K}$, $\mathbf{\Theta}^T \mathbf{x} \in \mathbb{R}^K$ *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

$$
\begin{aligned}
NVL(\boldsymbol{\theta}) &= -\sum_{i=1}^{n}\sum_{k=1}^{K} \log p_{ik}^{[y^{(i)}=k]} = -\sum_{i=1}^{n}\sum_{k=1}^{K} [y^{(i)} = k] \log p_{ik} \\
&= E(\boldsymbol{\theta})
\end{aligned}
$$

- donde

$$
p_{ik} = softmax(\boldsymbol{\theta}^T \mathbf{x}^{(i)})_k
$$

EMV para regresión logística multinomial

- Tomando el negativo de la verosimilitud logarítmica

$$
\begin{aligned}
NVL(\boldsymbol{\theta}) &= -\sum_{i=1}^{n}\sum_{k=1}^{K}\log p_{ik}^{[y^{(i)}=k]} = -\sum_{i=1}^{n}\sum_{k=1}^{K}[y^{(i)} = k]\log p_{ik} \\
&= E(\boldsymbol{\theta})
\end{aligned}
$$

- donde

$$p_{ik} = softmax(\boldsymbol{\theta}^T \mathrm{x}^{(i)})_k$$

- Parámetros se estiman por descenso por gradiente

$$\mathrm{g}(\boldsymbol{\theta}_k) = \frac{d}{d\boldsymbol{\theta}_k}E(\boldsymbol{\theta}_k) = \sum_{i=1}^{n}(p_{ik} - [y^{(i)} = k])\mathrm{x}^{(i)}$$

¿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 $P(\mathrm{X}, \mathrm{y}) = P(\mathrm{X}|\mathrm{y})P(\mathrm{y})$.

- La probabilidad condicional de las salidas dadas las clases $P(\mathrm{y}|\mathrm{X})$ 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 clases $P(\mathrm{y}|\mathrm{X})$.
- 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

- **Generativo**: podemos ignorar los atributos faltantes en la etapa de prueba y calcular
- **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**: usalmente mejor calibradas