Las matemáticas del *Machine Learning*, explicadas con dibujos

De los sumatorios a la forma de Jordan

Eduardo C. Garrido Merchán

Profesor propio adjunto, Departamento de Métodos Cuantitativos
FCEE – ICADE ▪ Investigador del IIT
Universidad Pontificia Comillas

Cómo leer estas slides

- Cada concepto se presenta en tres pasos: **(1)** dibujo, **(2)** ecuación, **(3)** para qué sirve.
- Los *términos* de las ecuaciones están **coloreados** para que veas en el dibujo a qué se refieren.
- No hace falta saber matemáticas avanzadas: empezamos por sumar y acabamos en autovectores.

Convenio de colores

azul = entradas / variables

naranja = parámetros / áreas

verde = datos / muestras

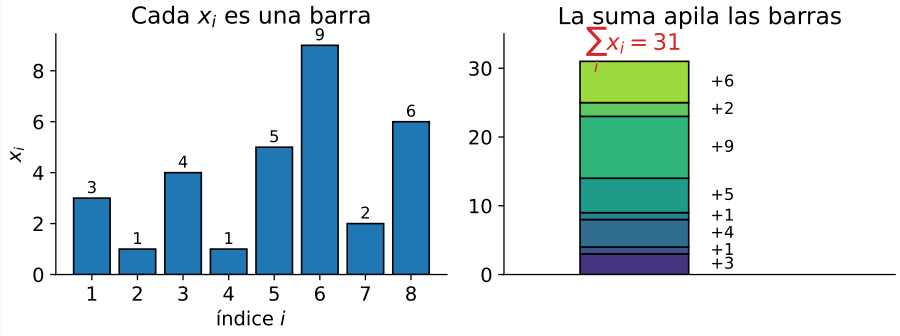
rojo = error / resultado

morado = derivada / cambio

1. Sumatorios y promedios

Sumatorios: $\sum$ es “apila estos números”

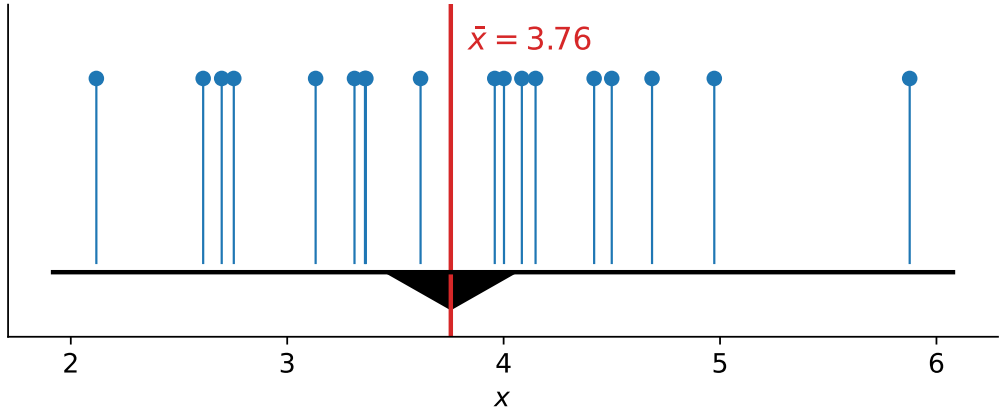
$$\sum_{i=1}^{n} x_i \;=\; x_1 + x_2 + x_3 + \cdots + x_n$$



Promedio: el centro de masa de los datos

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

El promedio es el punto donde la balanza queda en equilibrio



2. Funciones, derivadas y gradiente

Una función es una "máquina" $x \to f(x)$

Una función es una regla que a cada entrada x le asocia una salida f(x).

Ejemplos. $f(x) = 2x + 1$ (recta) $f(x) = x^2$ (parábola)

En ML, una red neuronal es *una función gigante* $f_\theta(x)$ con muchos parámetros θ que ajustamos.

Derivada = pendiente local

$$f'(x_0) = \lim_{dx \to 0} \frac{df}{dx}$$

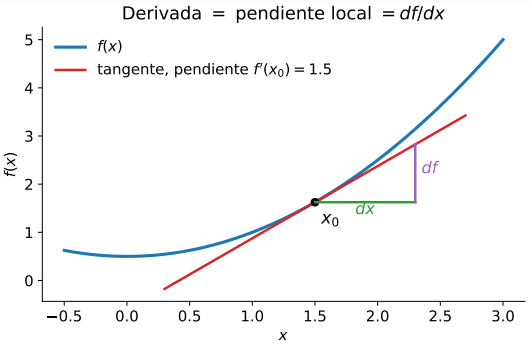
- dx: pasito muy pequeño en x.
- df: cuánto sube/baja f con ese pasito.
- Cociente df/dx = pendiente de la tangente roja.

Ejemplos.

$f(x) = x^2 \Rightarrow f'(x) = 2x.$

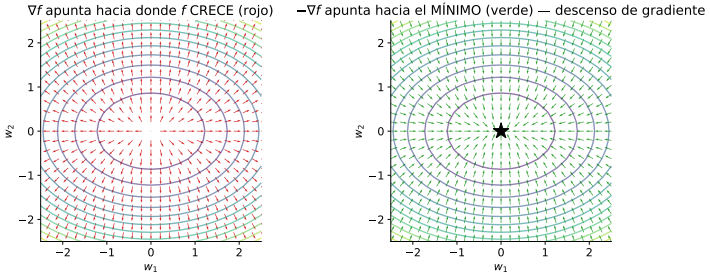
$f(x) = e^x \Rightarrow f'(x) = e^x.$

$f(x) = \sin x \Rightarrow f'(x) = \cos x.$



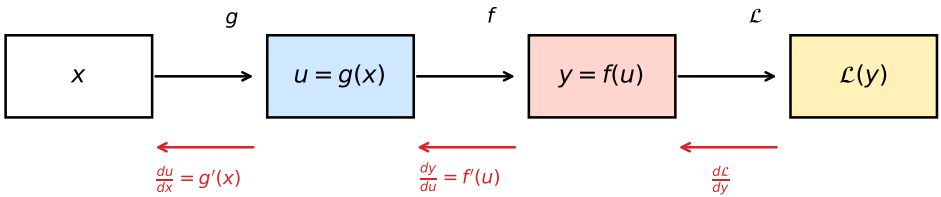
Gradiente: derivada en varias variables

$$\nabla f = \begin{pmatrix} \partial f/\partial w_1 \\ \partial f/\partial w_2 \\ \vdots \\ \partial f/\partial w_d \end{pmatrix} = \text{vector que apunta hacia donde } f \text{ crece más rápido.}$$



Ejemplo. $f(w_1, w_2) = w_1^2 + w_2^2 \Rightarrow \nabla f = (2w_1,\, 2w_2)$. En $(1, 2)$: $\nabla f = (2, 4)$.

Regla de la cadena (la base del *backpropagation*)

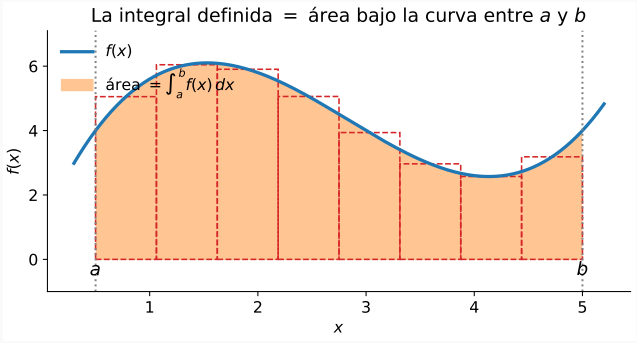


$$\frac{d\mathcal{L}}{dx} = \frac{d\mathcal{L}}{dy} \cdot \frac{dy}{du} \cdot \frac{du}{dx}$$ (regla de la cadena = backprop)

Intuición. Una red neuronal es una composición de funciones $\mathcal{L} \circ f_K \circ \cdots \circ f_1$. La regla de la cadena dice que el gradiente del coste respecto a un peso interior se obtiene multiplicando las derivadas locales hacia atrás. Eso es backpropagation.

3. Integrales paso a paso

Integral = área bajo la curva

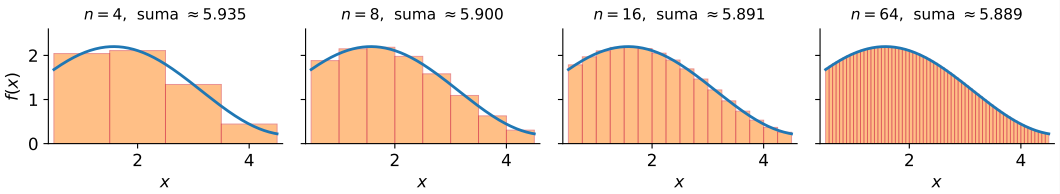


$$\int_a^b f(x)\,dx \approx \sum_{i=1}^{n} f(x_i)\cdot \Delta x$$

Cortamos $[a, b]$ en rectángulos de anchura Δx, sumamos sus áreas.

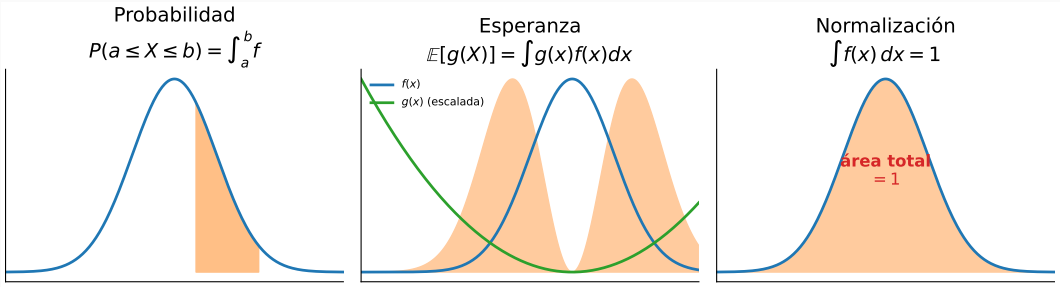
Más rectángulos ⇒ mejor aproximación

La integral $\int_a^b f(x)\,dx$ es el área bajo la curva: rectángulos cada vez más finos.



Intuición. La integral es la suma de infinitos rectángulos infinitamente finos. El símbolo $\int$ es de hecho una "S" alargada (de *summa*).

Para qué necesitamos integrales en ML

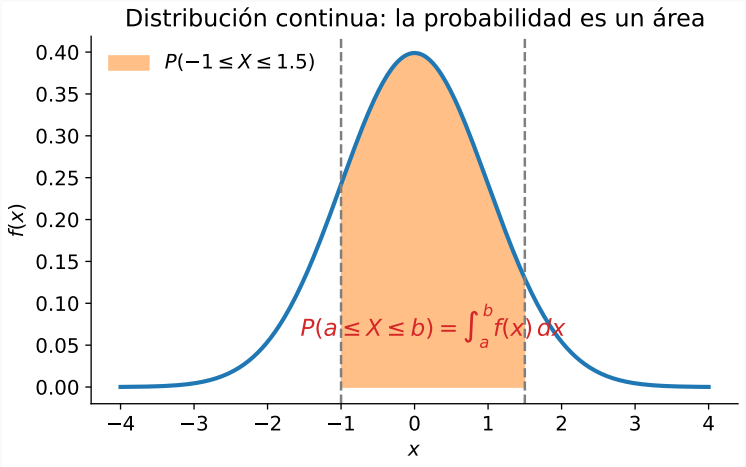


- Probabilidades de variables continuas.
- Esperanzas (medias teóricas): $\mathbb{E}[g(X)] = \int g(x)\, f(x)\, dx$.
- Normalización: una densidad debe integrar a 1.
- En la práctica casi siempre las aproximamos por *Monte Carlo*.

4. Distribuciones continuas

De histograma a densidad

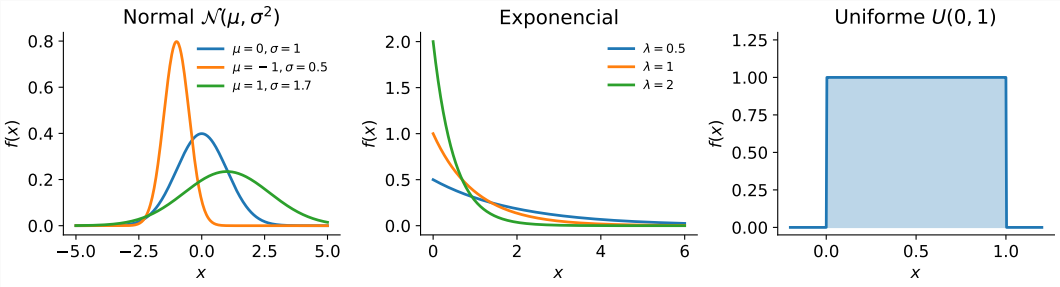
Una variable aleatoria continua X toma valores en un *intervalo*. Su densidad $f(x)$ cumple



$f(x) \geq 0$ y $\int f(x)\,dx = 1$.

$$P(a \leq X \leq b) \;=\; \int_a^b f(x)\,dx \quad \text{(el área naranja).}$$

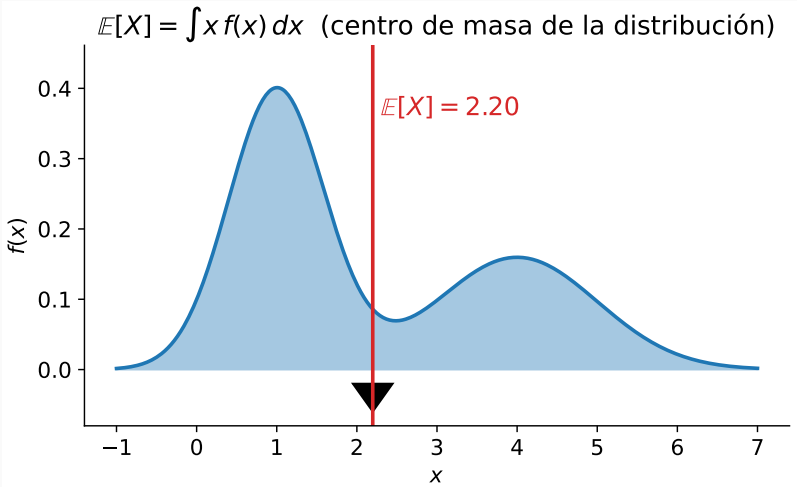
Tres distribuciones que aparecen siempre



- Normal $\mathcal{N}(\mu, \sigma^2)$: ruido, errores de medida, teorema central del límite.
- Exponencial: tiempo entre eventos (clientes que llegan, fallos, ...).
- Uniforme $U(0, 1)$: la fuente de aleatoriedad básica del ordenador.

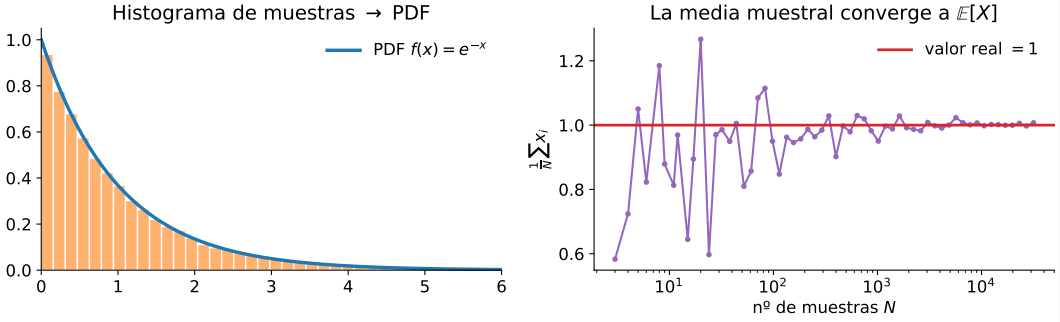
Esperanza: el “centro” de la distribución

$$\mathbb{E}[X] \;=\; \int_{-\infty}^{\infty} x \cdot f(x)\, dx$$



5. Muestreo (sampling)

¿Por qué muestrear? Porque la integral converge

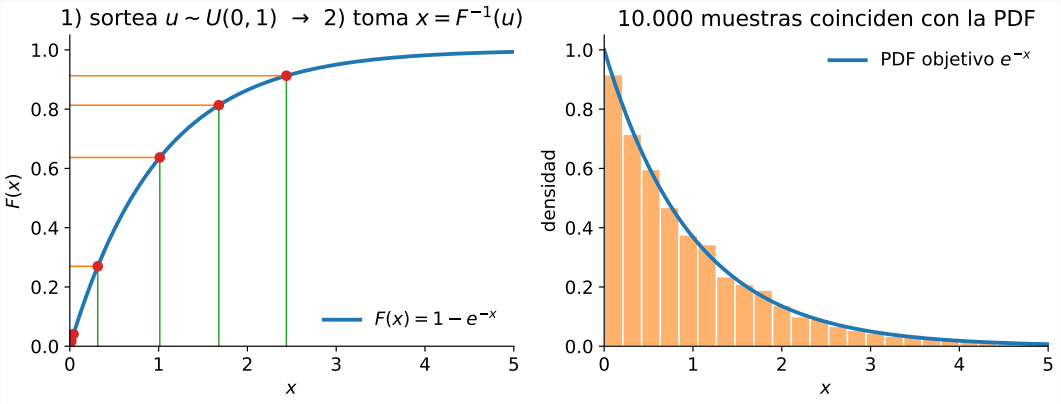


$$\mathbb{E}[g(X)] \;=\; \int g(x)\, f(x)\, dx \;\approx\; \frac{1}{N}\sum_{i=1}^{N} g(x_i) \quad \text{(Monte Carlo)}$$

Inverse-transform sampling

Receta. Si $F(x) = P(X \leq x)$ es la CDF:

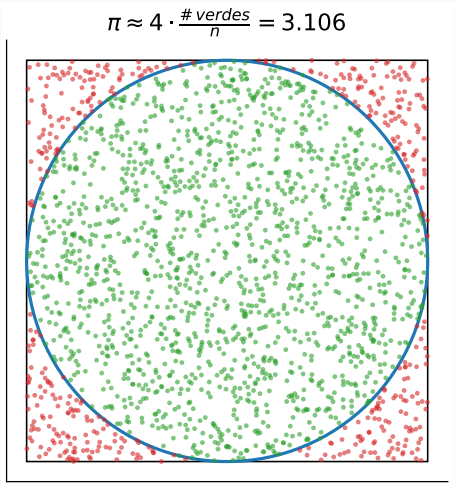
$$u \sim U(0,1) \quad \Longrightarrow \quad x = F^{-1}(u) \;\sim\; f(x)$$



Monte Carlo: estimar π tirando dardos

1. Generamos N puntos uniformes en $[-1, 1]^2$.
2. Contamos cuántos caen dentro del círculo de radio 1.
3. $$\pi \approx 4 \cdot \frac{\#\,\text{verdes}}{N}$$

Es exactamente la idea integral = área = fracción de muestras dentro.



6. Vectores y matrices

Producto escalar: ¿cuánto se parecen dos vectores?

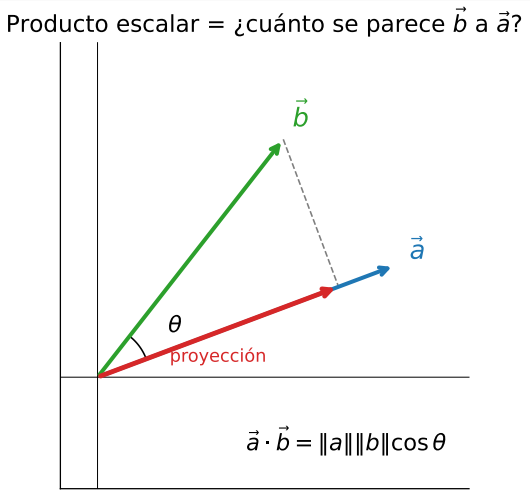
$$\vec{a} \cdot \vec{b} \;=\; \sum_i a_i b_i \;=\; \|\vec{a}\|\,\|\vec{b}\|\,\cos\theta$$

- Cero $\Leftrightarrow$ vectores perpendiculares.
- Positivo: apuntan al mismo lado.
- En ML: núcleo de cualquier capa lineal $y = \vec{w} \cdot \vec{x} + b$.

Ejemplo.

$\vec{a} = (3,4),\ \vec{b} = (1,2) \Rightarrow \vec{a} \cdot \vec{b} = 3 + 8 = 11.$

$\vec{a} = (1,0),\ \vec{b} = (0,1) \Rightarrow \vec{a} \cdot \vec{b} = 0$ (perpendiculares).

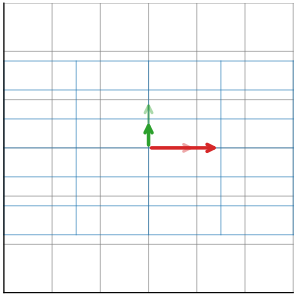


Una matriz *transforma* el plano

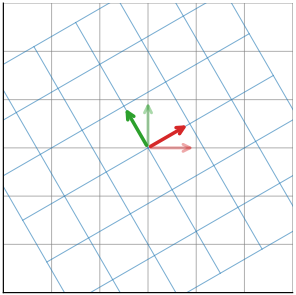
$$A\,\vec{v} \;=\; \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} a_{11}v_1 + a_{12}v_2 \\ a_{21}v_1 + a_{22}v_2 \end{pmatrix}$$

Una matriz A deforma el plano: gris = original, azul = transformado

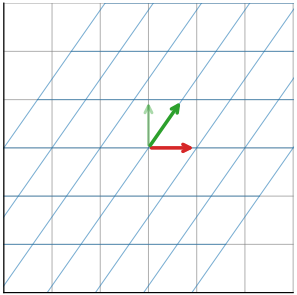
Escalado



Rotación 30°

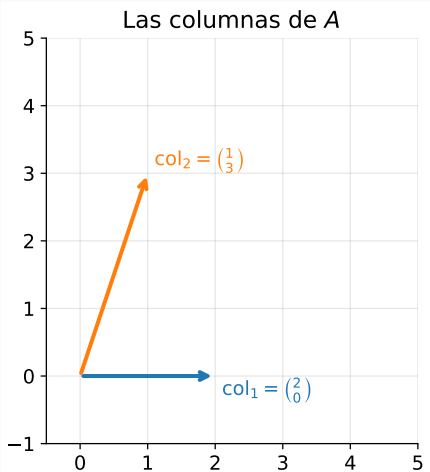


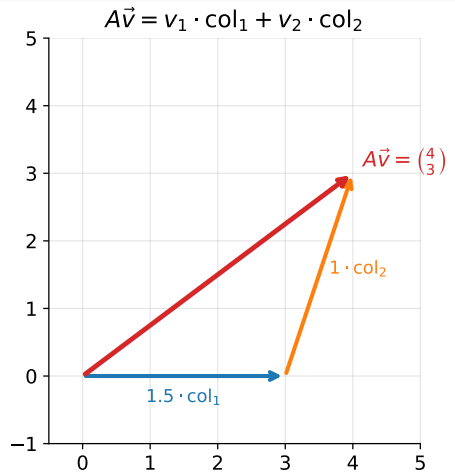
Cizalla (shear)



$A\vec{v}$ = combinación de las columnas de A

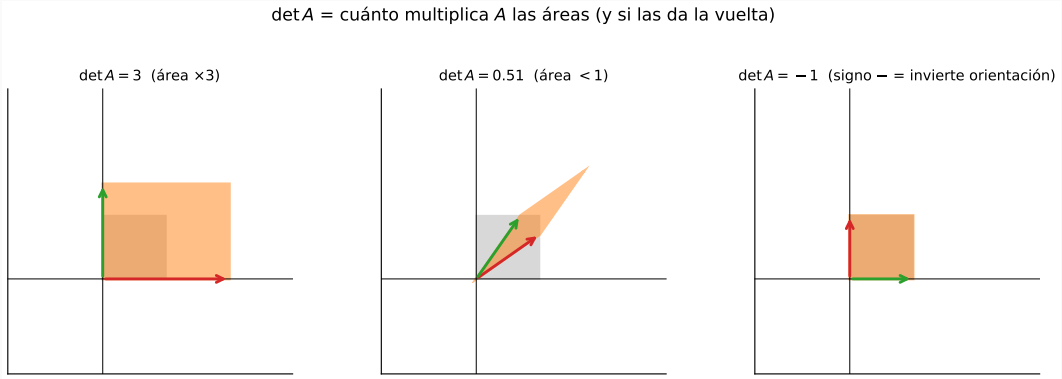
$$\begin{pmatrix} 2 & 1 \\ 0 & 3 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = v_1 \begin{pmatrix} 2 \\ 0 \end{pmatrix} + v_2 \begin{pmatrix} 1 \\ 3 \end{pmatrix}$$





Determinante: cuánto A multiplica las áreas

$\det A$ = factor por el que crecen (o se invierten) las áreas.



Intuición. $\det A = 0$ significa que A aplasta el plano (algunas direcciones desaparecen) $\Rightarrow$ la matriz no tiene inversa. En ML usamos $\det$ para *normalizing flows*, máxima verosimilitud Gaussiana, etc.

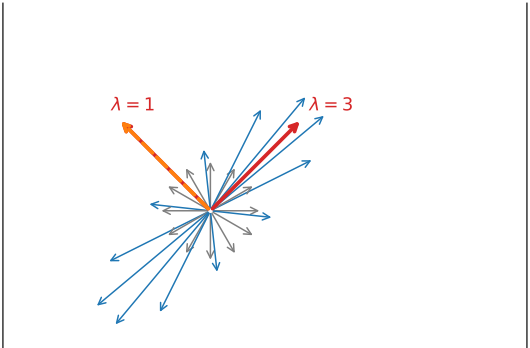
7. Autovalores, autovectores y forma de Jordan

La pregunta: ¿qué vectores no cambian de dirección?

$$A\,\vec{v} \;=\; \lambda\,\vec{v}$$

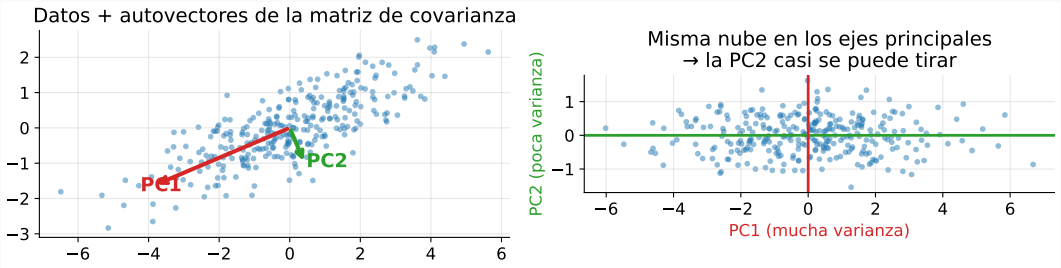
- $\vec{v}$: autovector; λ: autovalor asociado.
- Al aplicar A, el vector *sólo se estira/encoge* por λ.

Casi todo vector cambia de dirección al multiplicar por A...



...salvo los autovectores (rojos)

Aplicación estrella: PCA



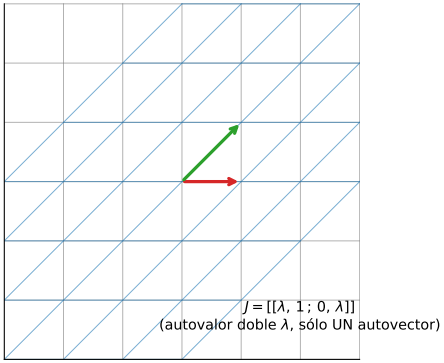
Los autovectores de la matriz de covarianza son las direcciones donde los datos varían más.

Proyectando sobre los primeros, comprimimos los datos perdiendo poca información.

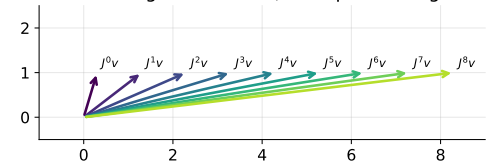
Cuando la diagonalización falla: forma de Jordan

$$J = \begin{pmatrix} \lambda & 1 \\ 0 & \lambda \end{pmatrix}$$

Bloque de Jordan: cizalla pura sobre el eigenespacio



Iterar J: el vector deriva en la dirección del autovector generalizado (no se puede diagonalizar)



- Tiene un autovalor doble λ pero un solo autovector.

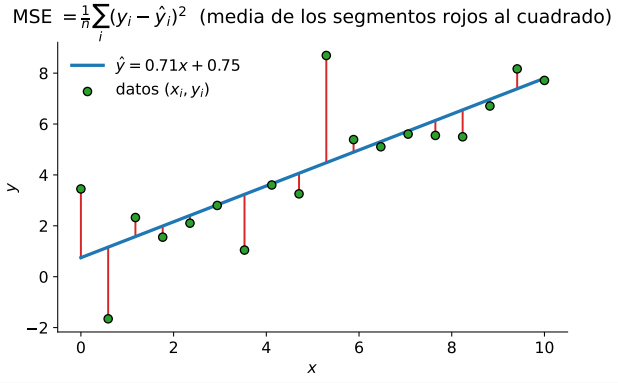
Por qué los autovalores importan en ML

- PCA y compresión de datos.
- Spectral clustering (autovectores del Laplaciano de un grafo).
- Estabilidad de redes profundas: si $|\lambda| > 1$ los gradientes *explotan*; si $|\lambda| < 1$ se *desvanecen*.
- PageRank (Google) $=$ autovector dominante de la matriz de enlaces.

8. Función de coste y descenso de gradiente

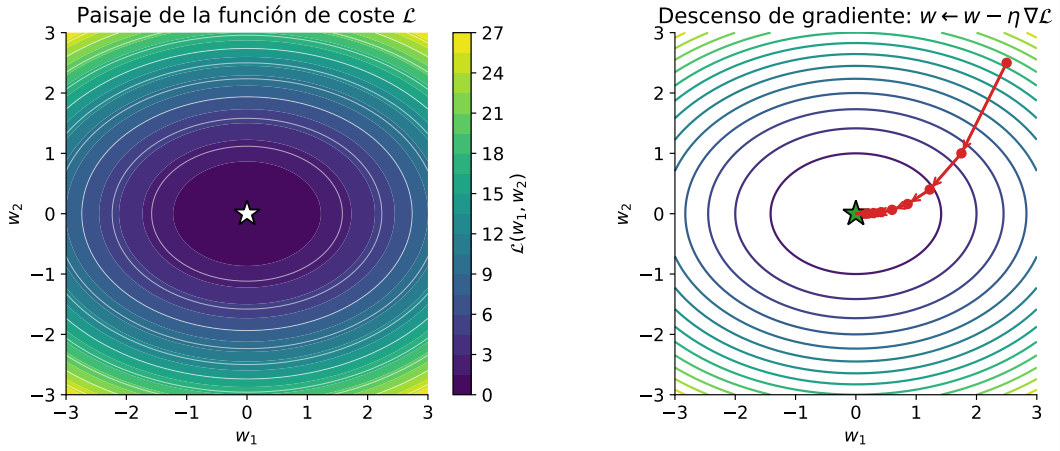
Mean Squared Error (MSE)

$$\mathcal{L}(\theta) \;=\; \frac{1}{n}\sum_{i=1}^{n}\big(y_i - \hat{y}_i(\theta)\big)^2$$



Ejemplo. Con $y = (2, 4, 6)$ y $\hat{y} = (2{,}5, 3, 7)$: $\mathcal{L} = \frac{1}{3}(0{,}25 + 1 + 1) = 0{,}75$.

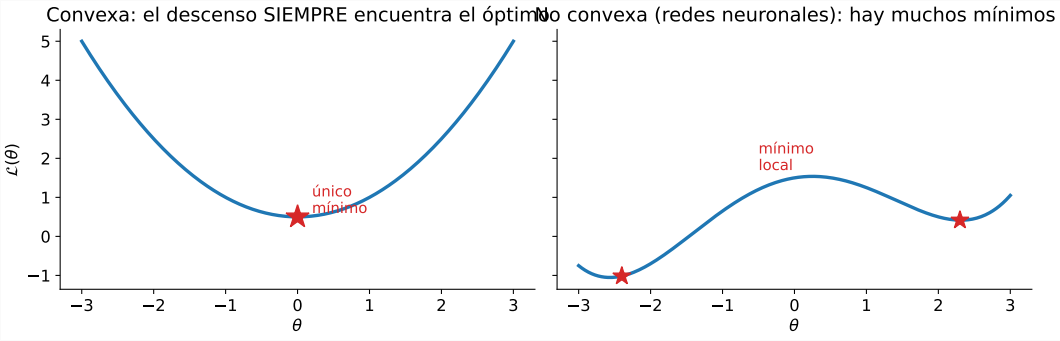
Paisaje del coste y descenso de gradiente



$$\theta_{t+1} = \theta_t - \eta \nabla \mathcal{L}(\theta_t)$$

η: *learning rate*. $\nabla \mathcal{L}$: dirección de máximo crecimiento $\rightarrow$ vamos al revés.

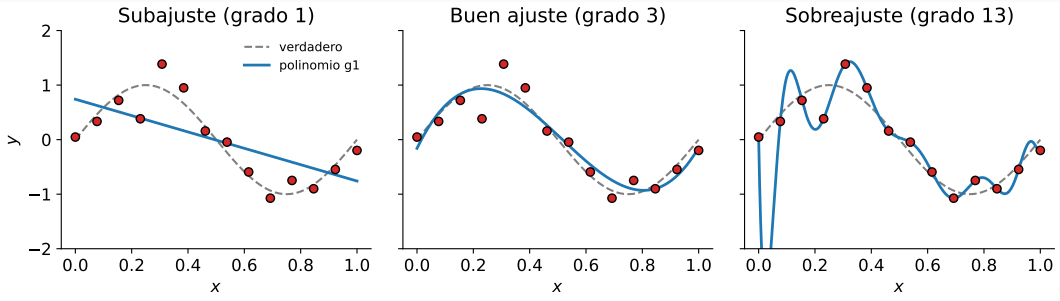
Convexa vs no convexa



Intuición. En problemas convexos (regresión lineal, regresión logística) el descenso de gradiente garantiza el óptimo global. En redes neuronales no, pero funciona sorprendentemente bien.

9. Sobreajuste y regularización

El problema: sobreajuste

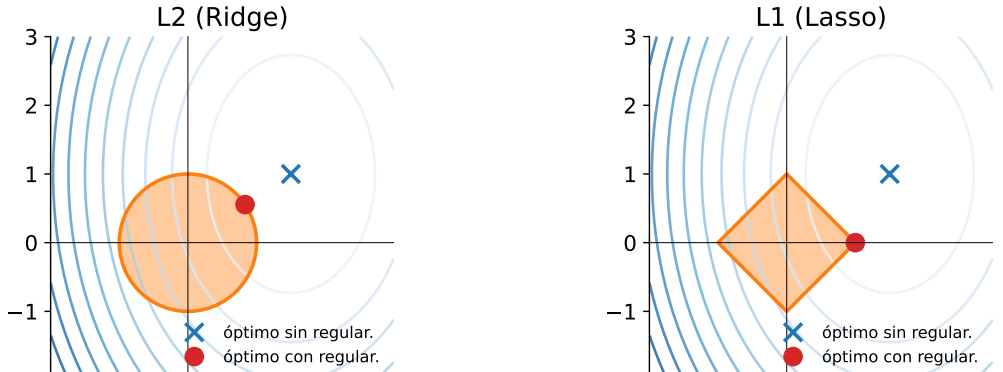


- **Subajuste:** modelo demasiado simple.
- **Buen ajuste:** captura la tendencia y *ignora el ruido*.
- **Sobreajuste:** pasa por todos los puntos pero predice fatal en datos nuevos.

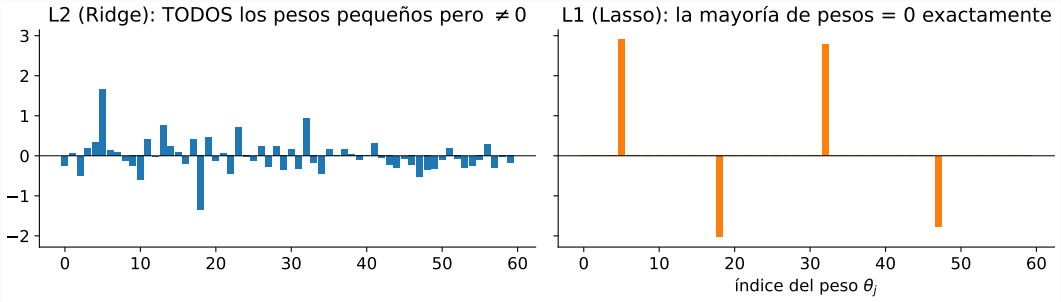
Regularización: añadir un “castigo” al coste

$$\mathcal{L}_{\text{total}}(\theta) = \underbrace{\mathcal{L}_{\text{datos}}(\theta)}_{\text{error en los datos}} + \lambda \underbrace{R(\theta)}_{\text{tamaño de los pesos}}$$

La regularización confina los pesos dentro de una bola: L2 → suaviza, L1 → cero exacto



L1 vs L2 en imágenes: el efecto sobre los pesos

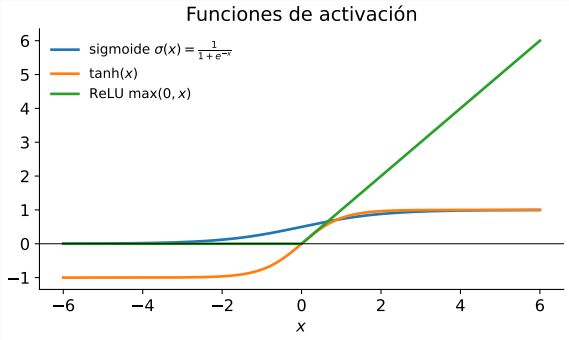


Intuición. La esquina puntiaguda de la bola L1 toca el óptimo en un eje, donde algunos pesos valen exactamente 0 (*sparsity*). La bola redonda L2 sólo los hace pequeños.

λ se elige por validación cruzada: probamos varios y nos quedamos con el que mejor predice.

10. Activaciones, log-loss y softmax

Funciones de activación: meter no-linealidad

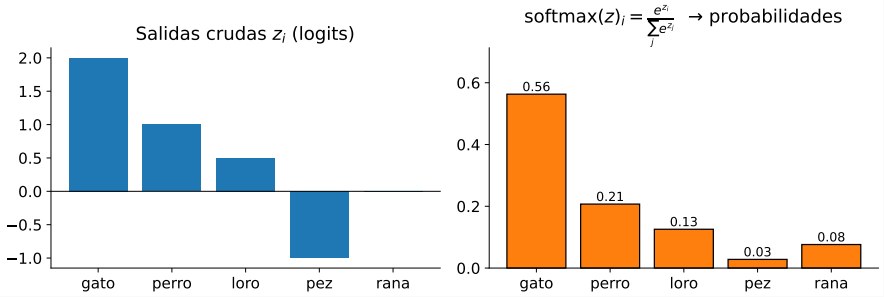


$$\sigma(x) = \frac{1}{1+e^{-x}} \qquad \tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \qquad \mathrm{ReLU}(x) = \max(0, x)$$

Sin no-linealidades, una red sería *una sola transformación lineal*.

Softmax: convertir números en probabilidades

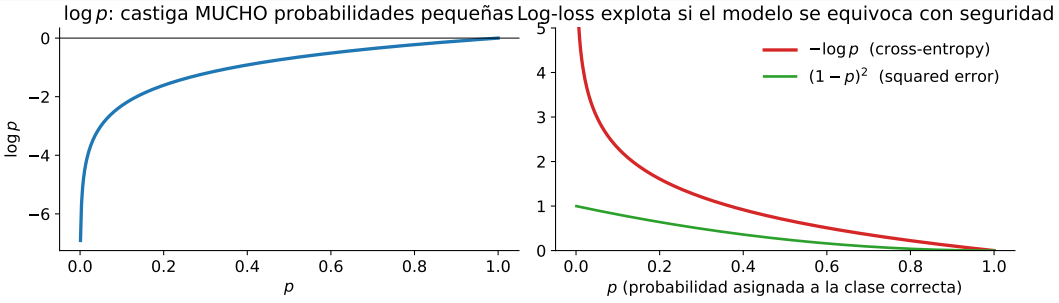
$$\text{softmax}(z)_i \;=\; \frac{e^{z_i}}{\sum_j e^{z_j}}$$



Ejemplo. $z = (2, 1, 0) \Rightarrow e^z \approx (7{,}39, 2{,}72, 1)$; suma $= 11{,}11$; $\text{softmax}(z) \approx (0{,}67, 0{,}24, 0{,}09)$.

Log-verosimilitud / cross-entropy

$$\mathcal{L}_{\text{CE}} \;=\; -\sum_{i} y_i \,\log p_i \quad (\text{con } p_i = \text{softmax}(z)_i)$$

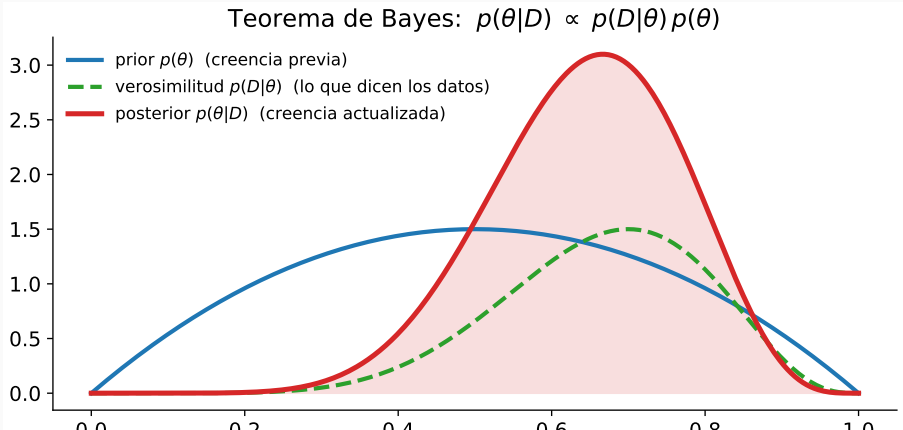


Intuición. Equivocarse con seguridad (asignar $p \to 0$ a la clase verdadera) cuesta infinito. Por eso la cross-entropy entrena mucho mejor que el MSE en clasificación.

11. Bayes, KL y Taylor: tres herramientas más

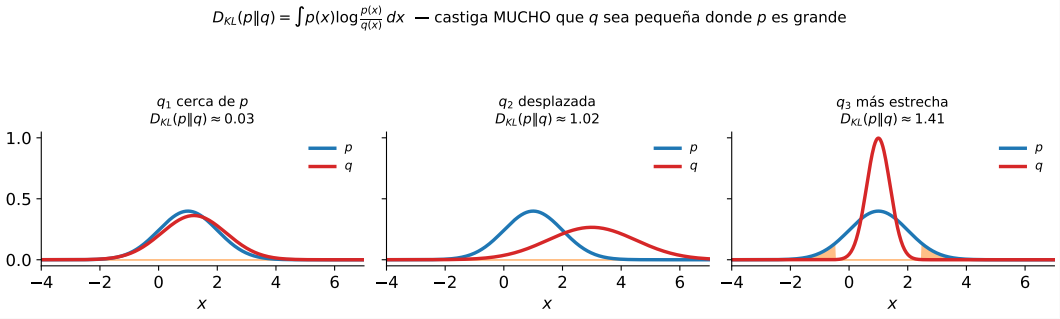
Teorema de Bayes: actualizar creencias con datos

$$\underbrace{p(\theta \mid D)}_{\text{posterior}} = \frac{\overbrace{p(D \mid \theta)}^{\text{verosimilitud}} \cdot \overbrace{p(\theta)}^{\text{prior}}}{p(D)}$$



KL-divergence: ¿cuánto se diferencian dos distribuciones?

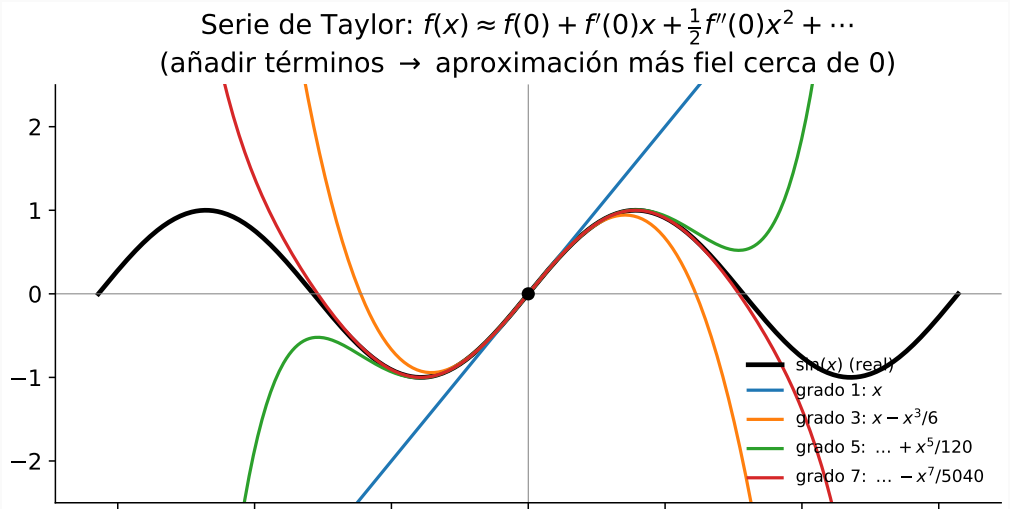
$$D_{KL}(p \,\|\, q) \;=\; \int p(x)\, \log \frac{p(x)}{q(x)}\, dx \;\;\geq\;\; 0$$



Intuición. No es una distancia simétrica. Castiga muy fuerte que q sea pequeña donde p es grande (zona naranja). En ML aparece en VAEs, modelos de difusión, fitting variacional y como justificación teórica de la cross-entropy.

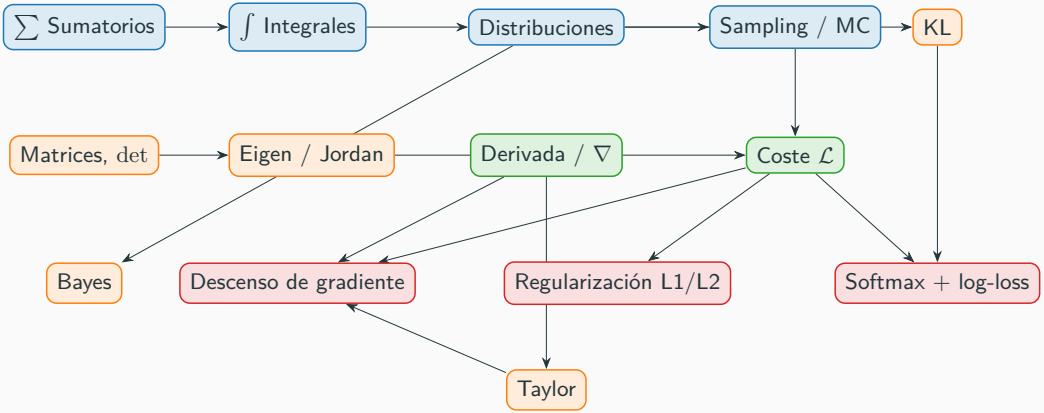
Serie de Taylor: aproximar funciones por polinomios

$$f(x) \approx f(x_0) + f'(x_0)\,(x - x_0) + \tfrac{1}{2} f''(x_0)\,(x - x_0)^2 + \cdots$$



Recapitulación

Mapa mental: cómo encajan todas las piezas



- Las ideas básicas (sumar, integrar, probabilidad) se combinan para definir un coste.
- Minimizamos ese coste con descenso de gradiente, controlando la complejidad con regularización.
- El álgebra lineal (matrices, eigen, Jordan) describe *cómo se mueve la información* dentro del modelo.

¿Preguntas?