El A”alto Diccionario de Aprendizaje Automático

Alexander Jung and Konstantina Olioumtsevits

May 12, 2025

please cite as: A. Jung and K. Olioumtsevits, *The Aalto Dictionary of Machine Learning*. Espoo, Finland: Aalto University, 2025.

Acknowledgements

Este diccionario de aprendizaje automático evolucionó a través del desarrollo y la enseñanza de varios cursos, incluyendo CS-E3210 Machine Learning: Basic Principles, CS-C3240 Machine Learning, CS-E4800 Artificial Intelligence, CS-EJ3211 Machine Learning with Python, CS-EJ3311 Deep Learning with Python, CS-E4740 Federated Learning, y CS-E407507 Human-Centered Machine Learning. Estos cursos se ofrecieron en Aalto University `https://www.aalto.fi/en`, a estudiantes adultos a travez de The Finnish Institute of Technology (FITech) `https://fitech.io/en/`, y a estudiantes internacionales a través de European University Alliance Unite! `https://www.aalto.fi/en/unite`.

Agradecemos a los estudiantes que brindaron valiosos comentarios que ayudaron a dar forma a este diccionario. Un agradecimiento especial a Mikko Seesto por su meticulosa corrección.

Lists of Symbols

Conjuntos y Funciones

$a \in \mathcal{A}$	El objeto a es un elemento del conjunto $\mathcal{A}$.
$a := b$	Utilizamos a como una abreviatura para b.
$\lvert\mathcal{A}\rvert$	La cardinalidad (es decir, el número de elementos) de un conjunto finito $\mathcal{A}$.
$\mathcal{A} \subseteq \mathcal{B}$	$\mathcal{A}$ es un subconjunto de $\mathcal{B}$.
$\mathcal{A} \subset \mathcal{B}$	$\mathcal{A}$ es un subconjunto propio de $\mathcal{B}$.
$\mathbb{N}$	Los números naturales $1, 2, \ldots$.
$\mathbb{R}$	Los números reales x [1].
$\mathbb{R}_+$	Los números reales no negativos $x \geq 0$.
$\mathbb{R}_{++}$	Los números reales positivos $x > 0$.

$\{0,1\}$	El conjunto que consta de los dos números reales 0 y 1.
$[0,1]$	El intervalo cerrado de números reales x con $0 \leq x \leq 1$.
$\operatorname{argmin}_{\mathbf{w}} f(\mathbf{w})$	El conjunto de minimizadores para una función de valor real $f(\mathbf{w})$.
$\mathbb{S}^{(n)}$	El conjunto de vectores de norma unitaria en $\mathbb{R}^{n+1}$.
$\log a$	El logaritmo del número positivo $a \in \mathbb{R}_{++}$.
$h(\cdot): \mathcal{A} \to \mathcal{B} : a \mapsto h(a)$	Una función (aplicación) que acepta cualquier elemento $a \in \mathcal{A}$ de un conjunto $\mathcal{A}$ como entrada y entrega un elemento bien definido $h(a) \in \mathcal{B}$ de un conjunto $\mathcal{B}$. El conjunto $\mathcal{A}$ es el dominio de la función h y el conjunto $\mathcal{B}$ es el codominio de h. El aprendizaje automático (aprendizaje automático) tiene como objetivo encontrar (o aprender) una función h (es decir, una hipótesis) que lea las features $\mathbf{x}$ de un punto de datos y entregue una predicción $h(\mathbf{x})$ para su etiqueta y.
$\nabla f(\mathbf{w})$	El gradiente de una función diferenciable de valor real $f : \mathbb{R}^d \to \mathbb{R}$ es el vector $\nabla f(\mathbf{w}) = \left(\frac{\partial f}{\partial w_1}, \ldots, \frac{\partial f}{\partial w_d}\right)^T \in \mathbb{R}^d$ [2, Ch. 9].

Matrices y Vectores

$\mathbf{x} = \big(x_1, \ldots, x_d\big)^T$	Un vector de longitud d, con su j-ésima entrada siendo x_j.
$\mathbb{R}^d$	El conjunto de vectores $\mathbf{x} = \big(x_1, \ldots, x_d\big)^T$ que consiste en d entradas de valor real $x_1, \ldots, x_d \in \mathbb{R}$.
$\mathbf{I}_{l \times d}$	Una matriz identidad generalizada con l filas y d columnas. Los elementos de $\mathbf{I}_{l \times d} \in \mathbb{R}^{l \times d}$ son iguales a 1 en la diagonal principal y iguales a 0 en los demás casos.
$\mathbf{I}_d, \mathbf{I}$	Una matriz identidad cuadrada de tamaño $d \times d$. Si el tamaño es claro por el contexto, omitimos el subíndice.
$\|\mathbf{x}\|_2$	La norma euclidiana (o ℓ_2) del vector $\mathbf{x} = \big(x_1, \ldots, x_d\big)^T \in \mathbb{R}^d$ definida como $\|\mathbf{x}\|_2 := \sqrt{\sum_{j=1}^{d} x_j^2}$.
$\|\mathbf{x}\|$	Alguna norma del vector $\mathbf{x} \in \mathbb{R}^d$ [3]. A menos que se especifique lo contrario, nos referimos a la norma euclidiana $\|\mathbf{x}\|_2$.
$\mathbf{x}^T$	La traspuesta de una matriz que tiene el vector $\mathbf{x} \in \mathbb{R}^d$ como su única columna.
$\mathbf{X}^T$	La traspuesta de una matriz $\mathbf{X} \in \mathbb{R}^{m \times d}$. Una matriz cuadrada de valores reales $\mathbf{X} \in \mathbb{R}^{m \times m}$ se denomina simétrica si $\mathbf{X} = \mathbf{X}^T$.
$\mathbf{0} = \big(0, \ldots, 0\big)^T$	El vector en $\mathbb{R}^d$ con cada entrada igual a cero.
$\mathbf{1} = \big(1, \ldots, 1\big)^T$	El vector en $\mathbb{R}^d$ con cada entrada igual a uno.

$\left(\mathbf{v}^T, \mathbf{w}^T\right)^T$	El vector de longitud $d + d'$ obtenido al concatenar las entradas del vector $\mathbf{v} \in \mathbb{R}^d$ con las entradas de $\mathbf{w} \in \mathbb{R}^{d'}$.
$\text{span}\{\mathbf{B}\}$	El espacio generado (span) por una matriz $\mathbf{B} \in \mathbb{R}^{a \times b}$, que es el subespacio de todas las combinaciones lineales de las columnas de $\mathbf{B}$, $\text{span}\{\mathbf{B}\} = \left\{\mathbf{Ba} : \mathbf{a} \in \mathbb{R}^b\right\} \subseteq \mathbb{R}^a$.
$\det(\mathbf{C})$	El determinante de la matriz $\mathbf{C}$.
$\mathbf{A} \otimes \mathbf{B}$	El producto de Kronecker de $\mathbf{A}$ y $\mathbf{B}$ [4].

Teoría de la Probabilidad

$\mathbb{E}_p\{f(\mathbf{z})\}$	La expectation de una función $f(\mathbf{z})$ de una variable aleatoria (RV) $\mathbf{z}$ cuya probability distribution es $p(\mathbf{z})$. Si la probability distribution es clara por el contexto, simplemente escribimos $\mathbb{E}\{f(\mathbf{z})\}$.
$p(\mathbf{x}, y)$	Una probability distribution conjunta de una RV cuyas realizaciónes son punto de datoss con features $\mathbf{x}$ y etiqueta y.
$p(\mathbf{x}\|y)$	Una probability distribution condicional de una RV $\mathbf{x}$ dado el valor de otra RV y [5, Sec. 3.5].
$p(\mathbf{x}; \mathbf{w})$	Una probability distribution parametrizada de una RV $\mathbf{x}$. La probability distribution depende de un vector de parámetros $\mathbf{w}$. Por ejemplo, $p(\mathbf{x}; \mathbf{w})$ podría ser una distribución normal multivariante con el vector de parámetros $\mathbf{w}$ dado por las entradas del vector de media $\mathbb{E}\{\mathbf{x}\}$ y la covariance matrix $\mathbb{E}\Big\{\big(\mathbf{x} - \mathbb{E}\{\mathbf{x}\}\big)\big(\mathbf{x} - \mathbb{E}\{\mathbf{x}\}\big)^T\Big\}$.
$\mathcal{N}(\mu, \sigma^2)$	La probability distribution de una variable aleatoria gaussiana (VA gaussiana) $x \in \mathbb{R}$ con media (o expectation) $\mu = \mathbb{E}\{x\}$ y varianza $\sigma^2 = \mathbb{E}\big\{(x-\mu)^2\big\}$.
$\mathcal{N}(\boldsymbol{\mu}, \mathbf{C})$	La distribución normal multivariante de un vector de valores VA gaussiana $\mathbf{x} \in \mathbb{R}^d$ con media (o expectation) $\boldsymbol{\mu} = \mathbb{E}\{\mathbf{x}\}$ y covariance matrix $\mathbf{C} = \mathbb{E}\big\{\big(\mathbf{x} - \boldsymbol{\mu}\big)\big(\mathbf{x} - \boldsymbol{\mu}\big)^T\big\}$.

Aprendizaje Automático

r	Un índice $r = 1, 2, \ldots$ que enumeran los punto de datoss.
m	El número de punto de datoss en (es decir, el tamaño de) un conjunto de datos.
$\mathcal{D}$	Un conjunto de datos $\mathcal{D} = \{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}\}$ es una lista de punto de datoss individuales $\mathbf{z}^{(r)}$, para $r = 1, \ldots, m$.
d	El número de features que caracterizan un punto de datos.
x_j	La j-ésima característica (feature) de un punto de datos. La primera feature se denota como x_1, la segunda como x_2, y así sucesivamente.
$\mathbf{x}$	El vector de características (feature vector) $\mathbf{x} = \big(x_1, \ldots, x_d\big)^T$ de un punto de datos cuyas entradas son las características individuales de un punto de datos.
$\mathcal{X}$	El espacio de características $\mathcal{X}$ es el conjunto de todos los valores posibles que las features $\mathbf{x}$ de un punto de datos pueden tomar.
$\mathbf{z}$	En lugar del símbolo $\mathbf{x}$, a veces usamos $\mathbf{z}$ como otro símbolo para denotar un vector cuyas entradas son las features individuales de un punto de datos. Necesitamos dos símbolos diferentes para distinguir entre características (features) crudas y características aprendidas [6, Ch. 9].
$\mathbf{x}^{(r)}$	El vector de características de el r-ésimo punto de datos dentro de un conjunto de datos.
$x_j^{(r)}$	La j-ésima feature del r-ésimo punto de datos dentro de un conjunto de datos.

$\mathcal{B}$	Un mini-lote (o subconjunto) de punto de datoss seleccionados aleatoriamente.
B	El tamaño de (es decir, el número de punto de datoss en) un mini-lote.
y	La etiqueta (o cantidad de interés) de un punto de datos.
$y^{(r)}$	La etiqueta del r-ésimo punto de datos.
$\left(\mathbf{x}^{(r)}, y^{(r)}\right)$	Las features y la etiqueta del r-ésimo punto de datos.
$\mathcal{Y}$	El espacio de etiquetas $\mathcal{Y}$ de un método de ML consiste en todos los valores potenciales de etiqueta que un punto de datos puede tener. El espacio de etiquetas nominal puede ser más grande que el conjunto de diferentes valores de etiqueta que surgen en un conjunto de datos dado (por ejemplo, un conjunto de entrenamiento). Los problemas (o métodos) de ML que utilizan un espacio de etiquetas numérico, como $\mathcal{Y} = \mathbb{R}$ o $\mathcal{Y} = \mathbb{R}^3$, se conocen como problemas de regresión. Los problemas (o métodos) de ML que utilizan un espacio de etiquetas discreto, como $\mathcal{Y} = \{0, 1\}$ o $\mathcal{Y} = \{gato, perro, ratón\}$, se conocen como problemas de clasificación.
η	La learning rate (o step size) utilizada por los métodos de gradient-based methods.
$h(\cdot)$	Un mapa de hipótesis que toma como entrada las features $\mathbf{x}$ de un punto de datos y entrega una predicción $\hat{y} = h(\mathbf{x})$ para su etiqueta y.
$\mathcal{Y}^{\mathcal{X}}$	Dado dos conjuntos $\mathcal{X}$ y $\mathcal{Y}$, denotamos por $\mathcal{Y}^{\mathcal{X}}$ el conjunto de todos los posibles mapas de hipótesis $h : \mathcal{X} \to \mathcal{Y}$.

$\mathcal{H}$	Un hypothesis space o model utilizado por un método de ML. El hypothesis space consiste en diferentes mapas de hipótesis $h : \mathcal{X} \to \mathcal{Y}$, entre los cuales el método de ML debe elegir.
$d_{\text{eff}}(\mathcal{H})$	La dimensión efectiva de un hypothesis space $\mathcal{H}$.
B^2	El sesgo cuadrado de una hipótesis aprendida $\hat{h}$ producida por un método de ML. El método se entrena con punto de datoss que se modelan como las realizaciónes de RVs. Dado que los datos son una realización de RVs, la hipótesis aprendida $\hat{h}$ también es una realización de una RV.
V	La varianza de los (parameters de la) hipótesis aprendida producida por un método de ML. El método se entrena con punto de datoss que se modelan como las realizaciónes de RVs. Dado que los datos son una realización de RVs, la hipótesis aprendida $\hat{h}$ también es una realización de una RV.
$L((\mathbf{x}, y), h)$	La loss incurrida al predecir la etiqueta y de un punto de datos utilizando la predicción $\hat{y} = h(\mathbf{x})$. La predicción $\hat{y}$ se obtiene al evaluar la hipótesis $h \in \mathcal{H}$ para la vector de características $\mathbf{x}$ del punto de datos.
E_v	El error de validación de una hipótesis h, que es su loss promedio incurrida en un conjunto de validación.
$\widehat{L}\big(h\|\mathcal{D}\big)$	El riesgo empírico o loss promedio incurrido por la hipótesis h en un conjunto de datos $\mathcal{D}$.

E_t	El error de entrenamiento de una hipótesis h, que es su loss promedio incurrido en un conjunto de entrenamiento.
t	Un índice de tiempo discreto $t = 0, 1, \ldots$ utilizado para enumerar eventos secuenciales (o instantes de tiempo).
t	Un índice que enumera las tarea de aprendizajes dentro de un problema de aprendizaje multitarea.
α	Un parámetro de regularization que controla la cantidad de regularization.
$\lambda_j\big(\mathbf{Q}\big)$	El j-ésimo eigenvalue (ordenado en forma ascendente o descendente) de una matriz positive semi-definite (psd) $\mathbf{Q}$. También usamos la abreviatura λ_j si la matriz correspondiente es clara por el contexto.
$\sigma(\cdot)$	La función de activación utilizada por una neurona artificial dentro de una artificial neural network (ANN).
$\mathcal{R}_{\hat{y}}$	Una decision region dentro de un espacio de características.
$\mathbf{w}$	Un vector de parámetros $\mathbf{w} = \big(w_1, \ldots, w_d\big)^T$ de un model, por ejemplo, los weights de un modelo lineal o en una ANN.
$h^{(\mathbf{w})}(\cdot)$	Un mapa de hipótesis que involucra model parameters ajustables $w_1, \ldots, w_d$ apilados en el vector $\mathbf{w} = \big(w_1, \ldots, w_d\big)^T$.
$\phi(\cdot)$	Un feature map $\phi : \mathcal{X} \to \mathcal{X}' : \mathbf{x} \mapsto \mathbf{x}' := \phi\big(\mathbf{x}\big) \in \mathcal{X}'$.
$K\big(\cdot, \cdot\big)$	Dado un espacio de características $\mathcal{X}$, un kernel es un mapa $K : \mathcal{X} \times \mathcal{X} \to \mathbb{C}$ que es psd.

Aprendizaje Federado

$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	Un graph no dirigido cuyos nodos $i \in \mathcal{V}$ representan devices dentro de un red de aprendizaje federado (red FL). Los bordes ponderados no dirigidos $\mathcal{E}$ representan conectividad entre devices y similitudes estadísticas entre sus conjunto de datoss y tarea de aprendizajes.
$i \in \mathcal{V}$	Un nodo que representa un device dentro de un red de aprendizaje federado (red FL). El dispositivo puede acceder a un local dataset y entrenar un local model.
$\mathcal{G}^{(\mathcal{C})}$	El subgrafo inducido de $\mathcal{G}$ utilizando los nodos en $\mathcal{C} \subseteq \mathcal{V}$.
$\mathbf{L}^{(\mathcal{G})}$	La Laplacian matrix de un graph $\mathcal{G}$.
$\mathbf{L}^{(\mathcal{C})}$	La Laplacian matrix del graph inducido $\mathcal{G}^{(\mathcal{C})}$.
$\mathcal{N}^{(i)}$	La entorno de un nodo i en un graph $\mathcal{G}$.
$d^{(i)}$	El grado ponderado $d^{(i)} := \sum_{i' \in \mathcal{N}^{(i)}} A_{i,i'}$ de un nodo i en un graph $\mathcal{G}$.
$d_{\max}^{(\mathcal{G})}$	El máximo grado ponderado de nodo en un graph $\mathcal{G}$.
$\mathcal{D}^{(i)}$	El local dataset $\mathcal{D}^{(i)}$ que posee el nodo $i \in \mathcal{V}$ de un red de aprendizaje federado (red FL).
m_i	El número de punto de datoss (es decir, el sample size) contenidos en el local dataset $\mathcal{D}^{(i)}$ en el nodo $i \in \mathcal{V}$.

$\mathbf{x}^{(i,r)}$	Las features del r-ésimo punto de datos en el local dataset $\mathcal{D}^{(i)}$.
$y^{(i,r)}$	La etiqueta del r-ésimo punto de datos en el local dataset $\mathcal{D}^{(i)}$.
$\mathbf{w}^{(i)}$	Los model parameters locales del device i dentro de un red de aprendizaje federado (red FL).
$L_i\left(\mathbf{w}\right)$	La loss function local utilizada por el device i para medir la utilidad de alguna elección $\mathbf{w}$ para los model parameters locales.
$L^{(\mathrm{d})}\left(\mathbf{x}, h\big(\mathbf{x}\big), h'\big(\mathbf{x}\big)\right)$	La loss incurrida por una hipótesis h' en un punto de datos con features $\mathbf{x}$ y etiqueta $h\big(\mathbf{x}\big)$ obtenida de otra hipótesis.
$\mathrm{stack}\big\{\mathbf{w}^{(i)}\big\}_{i=1}^{n}$	El vector $\left(\big(\mathbf{w}^{(1)}\big)^T, \ldots, \big(\mathbf{w}^{(n)}\big)^T\right)^T \in \mathbb{R}^{dn}$ que se obtiene apilando verticalmente los model parameters locales $\mathbf{w}^{(i)} \in \mathbb{R}^d$.

Machine Learning Concepts

k-fold cross-validation (k-fold CV) k-fold CV is a method for learning and validating a hipótesis using a given conjunto de datos. This method divides the conjunto de datos evenly into k subsets or folds and then executes k repetitions of model training (e.g., via minimización empírica del riesgo (ERM)) and validación. Each repetition uses a different fold as the conjunto de validación and the remaining $k-1$ folds as a conjunto de entrenamiento. The final output is the average of the error de validacións obtained from the k repetitions.

k-means The k-medias algorithm is a hard clustering method which assigns each punto de datos of a conjunto de datos to precisely one of k different clusters. The method alternates between updating the cluster assignments (to the cluster with the nearest media) and, given the updated cluster assignments, re-calculating the cluster medias [6, Ch. 8].

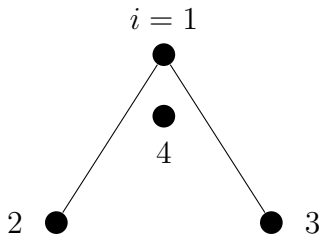
absolute error loss Consider a punto de datos with features $\mathbf{x} \in \mathcal{X}$ and numeric etiqueta $y \in \mathbb{R}$. The absolute error loss incurred by a hipótesis $h : \mathcal{X} \to \mathbb{R}$ is defined as $|y - h(\mathbf{x})|$, i.e., the absolute difference between the predicción $h(\mathbf{x})$ and the true etiqueta y.

agrupamiento basado en flujo El clustering basado en flujo agrupa los nodos de un graph no dirigido aplicando el algoritmo de k-means sobre vector de características específicos para cada nodo. Estos vector de

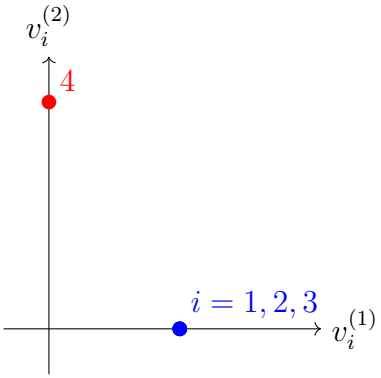
característicass se construyen a partir de flujos de red entre nodos fuente y destino seleccionados cuidadosamente [7].

agrupamiento en grafos El clustering en graphs tiene como objetivo agrupar punto de datoss que están representados como nodos de un graph $\mathcal{G}$. Las aristas del $\mathcal{G}$ representan similitudes por pares entre los punto de datoss. En algunos casos, es posible cuantificar el grado de estas similitudes mediante un edge weight [7, 8].

agrupamiento espectral El clustering espectral es una instancia particular del agrupamiento en grafos, es decir, agrupa punto de datoss representados como los nodos $i = 1, \ldots, n$ de un graph $\mathcal{G}$. El clustering espectral utiliza los eigenvectors de la Laplacian matrix $\mathbf{L}^{(\mathcal{G})}$ para construir vector de característicass $\mathbf{x}^{(i)} \in \mathbb{R}^d$ para cada nodo (es decir, para cada punto de datos) $i = 1, \ldots, n$. Podemos utilizar estos vector de característicass como entrada para métodos de clustering en Euclidean space,como k-means o soft clustering mediante Gaussian mixture model (GMM). Mas o menos, los vector de característicass de los nodos que pertenecen a un subconjunto bien conectado (o cluster) de nodos en $\mathcal{G}$ estan ubicados cerca en el Euclidean space $\mathbb{R}^d$ (Vea la Figura 1).



$$\mathbf{L}^{(\mathcal{G})} = \begin{pmatrix} 2 & -1 & -1 & 0 \\ -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$$



$$\mathbf{V} = \left(\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \mathbf{v}^{(3)}, \mathbf{v}^{(4)}\right)$$

$$\mathbf{v}^{(1)} = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}, \; \mathbf{v}^{(2)} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

Figure 1: **Arriba.** Izquierda: Un graph no dirigido $\mathcal{G}$ con cuatro nodos $i = 1, 2, 3, 4$, donde cada nodo representa un punto de datos. Derecha: La Laplacian matrix $\mathbf{L}^{(\mathcal{G})} \in \mathbb{R}^{4\times 4}$ y su EVD. **Abajo.** Izquierda: Un scatterplot de los punto de datoss usando los vector de característicass $\mathbf{x}^{(i)} = \left(v_i^{(1)}, v_i^{(2)}\right)^T$. Derecha: Dos eigenvectors $\mathbf{v}^{(1)}, \mathbf{v}^{(2)} \in \mathbb{R}^d$ correspondientes al eigenvalue $\lambda = 0$ de la Laplacian matrix $\mathbf{L}^{(\mathcal{G})}$.

algorithm An algorithm is a precise, step-by-step specification for how to produce an output from a given input within a finite number of computational steps [9]. For example, an algorithm for training a modelo lineal explicitly describes how to transform a given conjunto de entrenamiento into model parameters through a sequence of paso de gradientes. This informal characterization can be formalized rigorously via different mathematical models [10]. One very simple model of an algorithm is a collection of possible executions. Each execution is a sequence:

$$\text{input}, s_1, s_2, \ldots, s_T, \text{output}$$

that respects the constraints inherent to the computer executing the algorithm. Algorithms may be deterministic, where each input results uniquely in a single execution, or randomized, where executions can vary probabilistically. Randomized algorithms can thus be analyzed by modeling execution sequences as outcomes of random experiments, viewing the algorithm as a stochastic process [5, 11, 12]. Crucially, an algorithm encompasses more than just a mapping from input to output; it also includes the intermediate computational steps $s_1, \ldots, s_T$.

algoritmo distribuido Un algoritmo distribuido distributed es un algorithm diseñado para un tipo especial de computadora: una colección de dispositivos de cómputo interconectados (o nodos). Estos dispositivos se comunican y coordinan sus cálculos locales intercambiando mensajes a través de una red [13, 14]. A diferencia de un algorithm clásico, que se ejecuta en un solo device, un algoritmo distribuido algorithm se ejecuta de forma concurrente en múltiples devices con capacidades de

cómputo. Cada ejecución involucra tanto cálculos locales como eventos de intercambio de mensajes. Una ejecución genérica podría verse así:

$$\begin{aligned}
&\text{Node 1: } \text{input}_1, s_1^{(1)}, s_2^{(1)}, \ldots, s_{T_1}^{(1)}, \text{output}_1; \\
&\text{Node 2: } \text{input}_2, s_1^{(2)}, s_2^{(2)}, \ldots, s_{T_2}^{(2)}, \text{output}_2; \\
&\quad \vdots \\
&\text{Node N: } \text{input}_N, s_1^{(N)}, s_2^{(N)}, \ldots, s_{T_N}^{(N)}, \text{output}_N.
\end{aligned}$$

Cada device i inicia con su entrada local y ejecuta una secuencia de de cálculos intermedios $s_k^{(i)}$ en instantes de tiempo discretos $k = 1, \ldots, T_i$. Estos cálculos pueden depender tanto de cálculos previos locales como de mensajes recibidos de otros dispositivos. Uno de los usos clave de los algoritmos distribuidos es en federated learning (FL), donde una red de devices colabora para entrenar un model personalizado por dispositivo..

algoritmo en línea Un algoritmo en línea es un algorithm que procesa datos de forma incremental, recibiendo elementos de datos uno por uno y tomando decisiones o generando salidas inmediatamente, sin tener acceso a toda la entrada desde el inicio [15, 16]. A diferencia de un algoritmo fuera de línea, que dispone de toda la entrada desde el comienzo, un algoritmo en línea debe lidiar con la incertidumbre del futuro y no puede cambiar decisiones pasadas. Puede modelarse como una ejecución del tipo:

$$\text{init}, s_1, \text{out}_1, \text{in}_2, s_2, \text{out}_2, \ldots, \text{in}_T, s_T, \text{out}_T.$$

Cada ejecución comienza en un estado inicial y alterna entre cálculos, salidas y nuevas entradas. Un ejemplo importante en ML es el online

gradient descent (online GD), que actualiza model parameters conforme llegan nuevos punto de datoss.

application programming interface (API) An API is a formal mechanism for enabling software components to interact in a structured manner [17]. In the context of ML, APIs are frequently used to make a trained ML model accessible to different types of users. These users, which can be other computers or humans, can request a predicción for the etiqueta of a punto de datos by providing its features. The internal structure of the ML model remains hidden from the user. For instance, consider a trained ML model $\widehat{h}(x) := 2x + 1$. An API enables a user to submit the feature value $x = 3$ and obtain the response $\widehat{h}(3) = 7$ without knowledge of the detailed structure of the ML model or its training. In practice, the ML model is typically hosted on a computer (i.e., a server) connected to the internet. Another computer (i.e., a client) sends the features of a punto de datos to the server, which then computes $\widehat{h}(\mathbf{x})$ and returns the result to the external system. APIs help to modularize the development of ML applications by decoupling specific tasks. For instance, one team can focus on developing and training the model, while another team handles user interaction and integration of the model into applications.

aprendizaje automático (ML) El aprendizaje automático tiene como objetivo predecir una etiqueta a partir de las features de un punto de datos. Los métodos de ML logran esto aprendiendo una hipótesis de un hypothesis space (o model) mediante la minimización de una loss

function [6, 18]. Una formulación precisa de este principio es el ERM. Diferentes métodos de ML se obtienen de distintas elecciones de diseño para los punto de datoss (sus features y etiqueta), el model, y la loss function [6, Cap. 3].

aprendizaje de características Consideremos una aplicación de ML con punto de datoss caracterizados por features crudas $\mathbf{x} \in \mathcal{X}$. El aprendizaje de características se refiere a la tarea de aprender un mapeo

$$\mathbf{\Phi} : \mathcal{X} \to \mathcal{X}' : \mathbf{x} \mapsto \mathbf{x}',$$

que recibe como entrada las features crudas $\mathbf{x} \in \mathcal{X}$ de un punto de datos y entrega nuevas features $\mathbf{x}' \in \mathcal{X}'$ de un nuevo espacio de características $\mathcal{X}'$. Se obtienen diferentes métodos de aprendizaje de características a partir de diferentes elecciones de $\mathcal{X}, \mathcal{X}'$, de un hypothesis space $\mathcal{H}$ de posibles mapeos $\mathbf{\Phi}$, y de una medida cuantitativa de la utilidad de un mapeo específico $\mathbf{\Phi} \in \mathcal{H}$. Por ejemplo, principal component analysis (PCA) utiliza $\mathcal{X} := \mathbb{R}^d$, $\mathcal{X}' := \mathbb{R}^{d'}$ con $d' < d$, y un hypothesis space

$$\mathcal{H} := \left\{ \mathbf{\Phi} : \mathbb{R}^d \to \mathbb{R}^{d'} : \mathbf{x}' := \mathbf{F}\mathbf{x} \text{ con alguna } \mathbf{F} \in \mathbb{R}^{d' \times d} \right\}.$$

PCA mide la utilidad de un mapeo específico $\mathbf{\Phi}(\mathbf{x}) = \mathbf{F}\mathbf{x}$ por el mínimo error de reconstrucción lineal incurrido sobre un conjunto de datos,

$$\min_{\mathbf{G} \in \mathbb{R}^{d \times d'}} \sum_{r=1}^{m} \left\| \mathbf{G}\mathbf{F}\mathbf{x}^{(r)} - \mathbf{x}^{(r)} \right\|_2^2.$$

aprendizaje federado en red (NFL) El aprendizaje federado en red (NFL) se refiere a métodos que aprenden models personalizados de manera distribuida. Estos métodos aprenden a partir de local datasets que están relacionados por una estructura de red intrínseca.

aprendizaje federado horizontal (horizontal FL) El aprendizaje federado horizontal utiliza local datasets constituidos por diferentes punto de datoss, pero emplea las mismas features para caracterizarlos [19]. Por ejemplo, la predicción meteorológica utiliza una red de estaciones meteorológicas (observación) distribuidas espacialmente. Cada estación mide las mismas cantidades, como la temperatura diaria, la presión atmosférica y la precipitación. Sin embargo, distintas estaciones miden las características o features de diferentes regiones espaciotemporales. Cada región espaciotemporal representa un punto de datos individual, caracterizado por las mismas features (por ejemplo, temperatura diaria o presión atmosférica).

aprendizaje federado vertical (FL vertical) El aprendizaje federado vertical utiliza local datasets formados por los mismos punto de datoss, pero caracterizados mediante diferentes features [20]. Por ejemplo, diferentes proveedores de salud podrían contener información sobre la misma población de pacientes. Sin embargo, diferentes proveedores de salud recopilan distintas mediciones (por ejemplo, valores sanguíneos, electrocardiogramas, radiografías de tórax) para los mismos pacientes.

aprendizaje multitarea El aprendizaje multitarea tiene como objetivo aprovechar las relaciones entre diferentes tarea de aprendizajes. Considere dos tarea de aprendizajes obtenidas del mismo conjunto de datos de capturas de webcam. La primera tarea consiste en predecir la presencia de un ser humano, mientras que la segunda consiste en predecir la presencia de un automóvil. Podría ser útil utilizar la misma estructura

de red profunda para ambas tareas y permitir que solo los weights de la capa de salida final sean diferentes.

aprendizaje semi-supervisado (SSL) El aprendizaje semi-supervisado (SSL) utiliza punto de datoss no etiquetados para apoyar el aprendizaje de una hipótesis a partir de punto de dato etiquetados [21]. Este enfoque es particularmente útil para aplicaciones de ML que ofrecen una gran cantidad de punto de datoss no etiquetados, pero solo un número limitado de punto de dato etiquetados.

arrepentimiento (regret) El arrepentimiento de una hipótesis h en relación con otra hipótesis h', que sirve como referencia referencia (baseline), es la diferencia entre la loss incurrida por h y la loss incurrida por h' [16]. La referencia (baseline) hipótesis de referencia h' también se denomina experto.

artificial intelligence (AI) AI refers to systems that behave rationally in the sense of maximizing a long-term reward. The ML-based approach to AI is to train a model for predicting optimal actions. These predictions are computed from observations about the state of the environment. The choice of loss function sets AI applications apart from more basic ML applications. AI systems rarely have access to a labeled conjunto de entrenamiento that allows the average loss to be measured for any possible choice of model parameters. Instead, AI systems use observed reward signals to obtain a (point-wise) estimate for the loss incurred by the current choice of model parameters.

artificial neural network (ANN) An ANN is a graphical (signal-flow)

representation of a function that maps features of a punto de datos at its input to a predicción for the corresponding etiqueta at its output. The fundamental unit of an ANN is the artificial neuron, which applies an función de activación to its weighted inputs. The outputs of these neurons serve as inputs for other neurons, forming interconnected layers.

aspectos computacionales Por aspectos computacionales de un método de ML, nos referimos principalmente a los recursos computacionales requeridos para su implementación. Por ejemplo, si un método de ML utiliza técnicas de optimización iterativas para resolver ERM, sus aspectos computacionales incluyen: 1) cuántas many operaciones aritméticas se necesitan para implementar una sola iteración(paso de gradiente); y 2) cuántas iteraciones se requieren para obtener model parameters útiles. Un ejemplo importante de técnica de optimización iterativa es el gradient descent (GD).

aspectos estadísticos Por aspectos estadísticos de un método de ML, nos referimos a las propiedades de la probability distribution de su salida bajo un modelo probabilístico para los datos introducidos en el método.

atributo sensible La ML busca aprender una hipótesis que prediga la etiqueta de un punto de datos a partir de sus features. En algunas aplicaciones, es crucial garantizar que la salida del sistema no permita inferir atributos sensibles de los punto de datoss. Qué se considera atributo sensible depende del dominio de aplicación y debe definirse explícitamente.

autoencoder Un autoencoder es un método de ML que aprende simultánea-

mente un mapeo codificador $h(\cdot) \in \mathcal{H}$ y un mapeo decodificador $h^*(\cdot) \in \mathcal{H}^*$. Es una instancia de ERM que utiliza una loss calculada a partir del error de reconstrucción $\mathbf{x} - h^*\big(h\big(\mathbf{x}\big)\big)$.

backdoor A backdoor attack refers to the intentional manipulation of the training process underlying an ML method. This manipulation can be implemented by perturbing the conjunto de entrenamiento (datos poisoning) or the optimization algorithm used by an ERM-based method. The goal of a backdoor attack is to nudge the learned hipótesis $\hat{h}$ towards specific prediccións for a certain range of feature values. This range of feature values serves as a key (or trigger) to unlock a backdoor in the sense of delivering anomalous prediccións. The key $\mathbf{x}$ and the corresponding anomalous predicción $\hat{h}(\mathbf{x})$ are only known to the attacker.

bagging Bagging (or bootstrap aggregation) is a generic technique to improve (the robustness of) a given ML method. The idea is to use the bootstrap to generate perturbed copies of a given conjunto de datos and then to learn a separate hipótesis for each copy. We then predict the etiqueta of a punto de datos by combining or aggregating the individual prediccións of each separate hipótesis. For hipótesis maps delivering numeric etiqueta values, this aggregation could be implemented by computing the average of individual prediccións.

Bayes estimator Consider a modelo probabilístico with a joint probability distribution $p(\mathbf{x}, y)$ for the features $\mathbf{x}$ and etiqueta y of a punto de datos. For a given loss function $L\left(\cdot,\cdot\right)$, we refer to a hipótesis h as a Bayes estimator if its riesgo $\mathbb{E}\{L\left(\left(\mathbf{x}, y\right), h\right)\}$ is the mínimo [22]. Note that

the property of a hipótesis being a Bayes estimator depends on the underlying probability distribution and the choice for the loss function $L(\cdot, \cdot)$.

Bayes risk Consider a modelo probabilístico with a joint probability distribution $p(\mathbf{x}, y)$ for the features $\mathbf{x}$ and etiqueta y of a punto de datos. The Bayes riesgo is the mínimo possible riesgo that can be achieved by any hipótesis $h : \mathcal{X} \to \mathcal{Y}$. Any hipótesis that achieves the Bayes risk is referred to as a Bayes estimator [22].

bootstrap Para el análisis de métodos de ML methods, es a menudo útil interpretar un conjunto dado de punto de datoss $\mathcal{D} = \left\{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}\right\}$ como realizaciónes de RVsindependent and identically distributed (i.i.d.) con una probability distribution común $p(\mathbf{z})$.En general, no conocemos $p(\mathbf{z})$ exactamente, por lo que necesitamos estimarla. El método bootstrap utiliza el histograma de $\mathcal{D}$ como un estimador para la probability distribution subyacente $p(\mathbf{z})$.

clasificación La clasificación es la tarea de determinar una etiqueta y con valor discreto para un punto de datos, basado únicamente en sus features $\mathbf{x}$. La etiqueta y pertenece a un conjunto finito, como por ejemplo $y \in \{-1, 1\}$ o $y \in \{1, \ldots, 19\}$, y representa la categoría a la que pertenece el punto de datos.

clasificación multi-etiqueta Los problemas y métodos de clasificación multi-etiqueta utilizan punto de datoss caracterizados por varias etiquetas. Por ejemplo, considere un punto de datos que representa una

imagen con dos etiquetas. Una etiqueta indica la presencia de un ser humano en la imagen y otra etiqueta indica la presencia de un aútomovil.

clasificador Un clasificador es una hipótesis (función) $h(\mathbf{x})$ usada para predecir una etiqueta que toma valores de un espacio de etiquetas finito. Podemos usar directamente el valor $h(\mathbf{x})$ como predicción $\hat{y}$ para la etiqueta. pero normalmente se usa una función $h(\cdot)$ que entrega una cantidad numérica. La predicción es obtenida a travez de un paso de umbral. Por ejemplo, en un problema de clasificación binaria con $\mathcal{Y} \in \{-1, 1\}$, podríamos usar una hipótesis de valores reales $h(\mathbf{x}) \in \mathbb{R}$ como clasificador. Una predicción $\hat{y}$ puede obtenerse mediante:

$$\hat{y} = 1 \text{ for } h(\mathbf{x}) \geq 0 \text{ and } \hat{y} = -1 \text{ otherwise.} \tag{1}$$

Podemos caracterizar un clasificador mediante sus decision regiones $\mathcal{R}_a$, para cada valor posible de etiqueta $a \in \mathcal{Y}$.

clasificador lineal Consideremos punto de datoss caracterizados por features numéricos $\mathbf{x} \in \mathbb{R}^d$ y una etiqueta $y \in \mathcal{Y}$ de algún espacio de etiquetas finito $\mathcal{Y}$. Un clasificador lineal se caracteriza por tener decision regions que están separadas por hiperplanos en $\mathbb{R}^d$ [6, Ch. 2].

cluster A cluster is a subset of punto de datoss that are more similar to each other than to the punto de datoss outside the cluster. The quantitative measure of similarity between punto de datoss is a design choice. If punto de datoss are characterized by Euclidean vector de características $\mathbf{x} \in \mathbb{R}^d$, we can define the similarity between two punto de datoss via the Euclidean distance between their vector de características.

clustered federated learning (CFL) Clustered FL assumes that local datasets are naturally grouped into clusters. The local datasets belonging to the same cluster have similar statistical properties. Clustered FL aggregates local datasets in the same cluster to obtain a conjunto de entrenamiento for the training of a cluster-specific model. Minimización de variación total generalizada (GTVMin) facilitates this clustering implicitly by enforcing approximate similarity of model parameters across well-connected subsets of the red de aprendizaje federado (red FL).

clustering Clustering methods decompose a given set of punto de datoss into a few subsets, which are referred to as clusters. Each cluster consists of punto de datoss that are more similar to each other than to punto de datoss outside the cluster. Different clustering methods use different measures for the similarity between punto de datoss and different forms of cluster representations. The clustering method k-means uses the average feature vector (cluster media) of a cluster as its representative. A popular soft clustering method based on GMM represents a cluster by a distribución normal multivariante.

clustering assumption The clustering assumption postulates that punto de datoss in a conjunto de datos form a (small) number of groups or clusters. Punto de datoss in the same cluster are more similar to each other than those outside the cluster [21]. We obtain different clustering methods by using different notions of similarity between punto de datoss.

confusion matrix Consider punto de datoss characterized by features $\mathbf{x}$ and etiqueta y having values from the finite espacio de etiquetas $\mathcal{Y} =$

$\{1, \ldots, k\}$. The confusion matrix is a $k \times k$ matrix with rows representing different values c of the true label of a punto de datos. The columns of a confusion matrix correspond to different values c' delivered by a hypothesis $h(\mathbf{x})$. The (c, c')-th entry of the confusion matrix is the fraction of punto de datoss with the etiqueta $y = c$ and the predicción $\hat{y} = c'$ assigned by the hipótesis h.

conjunto de datos Un conjunto de datos se refiere a una colección de punto de datoss. Estos punto de datoss contienen información sobre alguna cantidad de interés (o etiqueta) dentro de una aplicación de ML. Los métodos de ML utilizan conjuntos de datos para el entrenamiento de model (por ejemplo, a través de ERM) y para la validación de models. Nuestra noción de conjunto de datos es muy flexible, ya que permite diferentes tipos de punto de datoss. De hecho, los punto de datoss pueden ser objetos físicos concretos (como humanos o animales) o objetos abstractos (como números). Como ejemplo, la Figura 2 muestra un conjunto de datos que consiste en vacas como punto de datoss.

Figure 2: “Cows in the Swiss Alps” by User:Huhu Uet is licensed under [CC BY-SA 4.0](https://creativecommons.org/licenses/by-sa/4.0/)

Con frecuencia, un ingeniero de ML no tiene acceso directo a un conjunto de datos. De hecho, acceder al conjunto de datos en la Figura requeriría visitar el rebaño de vacas en los Alpes. En su lugar, necesitamos utilizar una aproximación (o representación) del conjunto de datos que sea más conveniente para trabajar. Se han desarrollado diferentes modelos matemáticos para la representación (o aproximación) de conjuntos de datos [23], [24], [25], [26]. Uno de los modelos de datos más adoptados es el modelo relacional, model, que organiza los datos en una tabla (o relación) [27], [23]. Una tabla se compone de filas y columnas:

- Cada fila de la tabla representa un solo punto de datos.
- Cada columna de la tabla corresponde a un atributo específico del punto de datos. Los métodos de ML pueden utilizar estos atributos como features y etiquetas del punto de datos.

Por ejemplo, la Tabla 1 muestra una representación del conjunto de datos en la Figura 2. En el modelo relacional, el orden de las filas es irrelevante, y cada atributo (es decir, columna) debe estar definido de manera precisa con un dominio, que especifica el conjunto de valores posibles. En las aplicaciones de ML, estos dominios de atributos se convierten en el espacio de características y el espacio de etiquetas.

Name	**Weight**	**Age**	**Height**	**Stomach temp**
Zenzi	100	4	100	25
Berta	140	3	130	23
Resi	120	4	120	31

Table 1: Una relación (o tabla) que representa el conjunto de datos en la Figura .

Si bien el modelo relacional es útil para el estudio de muchas aplicaciones de ML, puede ser insuficiente para cumplir con los requisitos de inteligencia artificial confiable (IA confiable). Enfoques modernos como las hojas de datos para conjuntos de datos proporcionan una documentación más completa, incluyendo detalles sobre el proceso de recolección del conjunto de datos, su uso previsto y otra información contextual [28].

conjunto de entrenamiento Un conjunto de entrenamiento es un conjunto de datos $\mathcal{D}$ que consiste en algunos punto de datoss usados en ERM para aprender unahipótesis $\hat{h}$. La loss promedio de $\hat{h}$ en el conjunto de entrenamiento se denomina error de entrenamiento. La comparación

del error de entrenamiento con el error de validación de $\hat{h}$ nos permite diagnosticar el método de ML e informa cómo mejorar el error de validación (por ejemplo, utilizando un hypothesis space diferente o recopilando más punto de datoss) [6, Sec. 6.6].

conjunto de prueba Un conjunto de punto de datoss que no ha sido utilizado ni para entrenar un model (por ejemplo, mediante ERM) ni en un conjunto de validación para elegir entre diferentes models.

conjunto de validación Un conjunto de punto de datoss usado para estimar el riesgo de una hipótesis $\hat{h}$ que ha sido aprendida mediante algún método de ML (por ejemplo, resolviendo ERM). El loss promedio de $\hat{h}$ en el conjunto de validación validación se denomina error de validación y puede utilizarse para diagnosticar un método de ML (vea [6, Sec. 6.6]). La comparación entre el error de entrenamiento y el error de validación puede proporcionar información sobre cómo mejorar el método de ML method (como usar unhypothesis space diferente).

convex A subset $\mathcal{C} \subseteq \mathbb{R}^d$ of the Euclidean space $\mathbb{R}^d$ is referred to as convex if it contains the line segment between any two points $\mathbf{x}, \mathbf{y} \in \mathcal{C}$ in that set. A function $f: \mathbb{R}^d \to \mathbb{R}$ is convex if its epigraph $\left\{ \left(\mathbf{w}^T, t \right)^T \in \mathbb{R}^{d+1} : t \geq f(\mathbf{w}) \right\}$ is a convex set [29]. We illustrate one example of a convex set and a convex function in Figure 3.

Figure 3: Left: A convex set $\mathcal{C} \subseteq \mathbb{R}^d$. Right: A convex function $f : \mathbb{R}^d \to \mathbb{R}$.

convex clustering Consider a conjunto de datos $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)} \in \mathbb{R}^d$. Convex clustering learns vectors $\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(m)}$ by minimizing

$$\sum_{r=1}^{m} \left\| \mathbf{x}^{(r)} - \mathbf{w}^{(r)} \right\|_2^2 + \alpha \sum_{i,i' \in \mathcal{V}} \left\| \mathbf{w}^{(i)} - \mathbf{w}^{(i')} \right\|_p .$$

Here, $\|\mathbf{u}\|_p := \left(\sum_{j=1}^{d} |u_j|^p \right)^{1/p}$ denotes the p-norma (for $p \geq 1$). It turns out that many of the optimal vectors $\widehat{\mathbf{w}}^{(1)}, \ldots, \widehat{\mathbf{w}}^{(m)}$ coincide. A cluster then consists of those punto de datoss $r \in \{1, \ldots, m\}$ with identical $\widehat{\mathbf{w}}^{(r)}$ [30, 31].

Courant–Fischer–Weyl min-max characterization Consider a psd matrix $\mathbf{Q} \in \mathbb{R}^{d \times d}$ with EVD (or spectral decomposition),

$$\mathbf{Q} = \sum_{j=1}^{d} \lambda_j \mathbf{u}^{(j)} \left(\mathbf{u}^{(j)} \right)^T .$$

Here, we use the ordered (in increasing fashion) eigenvalues

$$\lambda_1 \leq \ldots \leq \lambda_n .$$

The Courant–Fischer–Weyl min-max characterization [3, Th. 8.1.2] represents the eigenvalues of $\mathbf{Q}$ as the solutions to certain optimization problems.

covariance matrix The covariance matrix of an RV $\mathbf{x} \in \mathbb{R}^d$ is defined as $\mathbb{E}\Big\{\big(\mathbf{x} - \mathbb{E}\{\mathbf{x}\}\big)\big(\mathbf{x} - \mathbb{E}\{\mathbf{x}\}\big)^T\Big\}$.

data augmentation Datos augmentation methods add synthetic punto de datoss to an existing set of punto de datoss. These synthetic punto de datoss are obtained by perturbations (e.g., adding noise to physical measurements) or transformations (e.g., rotations of images) of the original punto de datoss. These perturbations and transformations are such that the resulting synthetic punto de datoss should still have the same etiqueta. As a case in point, a rotated cat image is still a cat image even if their vector de características (obtained by stacking pixel color intensities) are very different (see Figure 4). Datos augmentation can be an efficient form of regularization.

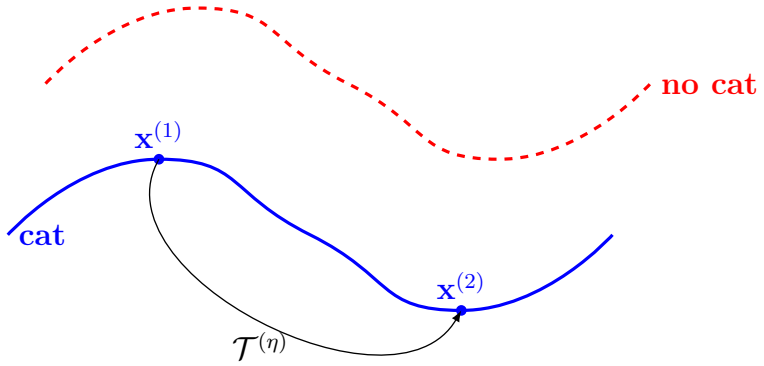


Figure 4: Datos augmentation exploits intrinsic symmetries of punto de datoss in some espacio de características $\mathcal{X}$. We can represent a symmetry by an operator $\mathcal{T}^{(\eta)} : \mathcal{X} \to \mathcal{X}$, parametrized by some number $\eta \in \mathbb{R}$. For example, $\mathcal{T}^{(\eta)}$ might represent the effect of rotating a cat image by η degrees. A punto de datos with vector de características $\mathbf{x}^{(2)} = \mathcal{T}^{(\eta)}\left(\mathbf{x}^{(1)}\right)$ must have the same etiqueta $y^{(2)} = y^{(1)}$ as a punto de datos with vector de características $\mathbf{x}^{(1)}$.

data minimization principle European datos protection regulation includes a datos minimization principle. This principle requires a datos controller to limit the collection of personal information to what is directly relevant and necessary to accomplish a specified purpose. The datos should be retained only for as long as necessary to fulfill that purpose [32, Article 5(1)(c)], [33].

data normalization Datos normalization refers to transformations applied to the vector de característicass of punto de datoss to improve the ML method's aspectos estadísticos or aspectos computacionales. For example, in linear regression with gradient-based methods using a fixed learning rate, convergence depends on controlling the norma of vector de

característicass in the conjunto de entrenamiento. A common approach is to normalize vector de característicass such that their norma does not exceed one [6, Ch. 5].

data poisoning Datos poisoning refers to the intentional manipulation (or fabrication) of punto de datoss to steer the training of an ML model [34, 35]. The protection against datos poisoning is particularly important in distributed ML applications where conjunto de datoss are decentralized.

datos Los datos se refieren a objetos que llevan información. Estos objetos pueden ser tanto objetos físicos concretos (como personas o animales) como conceptos abstractos (como números). A menudo, utilizamos representaciones (o aproximaciones) de los datos originales que son más convenientes para su procesamiento. Estas aproximaciones se basan en diferentes modelos de datos, siendo el modelo de datos relacional uno de los más utilizados [27].

datos en red Los datos en red consisten en local datasets relacionados por alguna noción de similitud por pares. Podemos representar los datos en red utilizando un graph cuyos nodos contienen local datasets y cuyas aristas codifican similitudes por pares. Un ejemplo de datos en red surge en las aplicaciones de FL donde los local datasets son generados por devices distribuidos espacialmente.

datos faltantes Considere un conjunto de datos constituido por punto de datoss recopilados a través de algún device físico. Debido a imperfecciones y fallas, algunos de los valores de feature o etiqueta de los punto de datoss podrían estar corruptos o simplemente faltar. La imputación

de Datos tiene como objetivo estimar estos valores faltantes [36]. Podemos interpretar la imputación de datos como un problema de MLdonde la etiqueta de un punto de datos es es el valor de la feature corrupta.

decision boundary Consider a hipótesis map h that reads in a feature vector $\mathbf{x} \in \mathbb{R}^d$ and delivers a value from a finite set $\mathcal{Y}$. The decision boundary of h is the set of vectors $\mathbf{x} \in \mathbb{R}^d$ that lie between different decision regions. More precisely, a vector $\mathbf{x}$ belongs to the decision boundary if and only if each entorno $\{\mathbf{x}' : \|\mathbf{x} - \mathbf{x}'\| \leq \varepsilon\}$, for any $\varepsilon > 0$, contains at least two vectors with different function values.

decision region Consider a hipótesis map h that delivers values from a finite set $\mathcal{Y}$. For each etiqueta value (category) $a \in \mathcal{Y}$, the hipótesis h determines a subset of feature values $\mathbf{x} \in \mathcal{X}$ that result in the same output $h(\mathbf{x}) = a$. We refer to this subset as a decision region of the hipótesis h.

decision tree A decision tree is a flow-chart-like representation of a hipótesis map h. More formally, a decision tree is a directed graph containing a root node that reads in the vector de características $\mathbf{x}$ of a punto de datos. The root node then forwards the punto de datos to one of its children nodes based on some elementary test on the features $\mathbf{x}$. If the receiving child node is not a leaf node, i.e., it has itself children nodes, it represents another test. Based on the test result, the punto de datos is forwarded to one of its descendants. This testing and forwarding of the punto de datos is continued until the punto de datos ends up in a leaf node (having no children nodes).

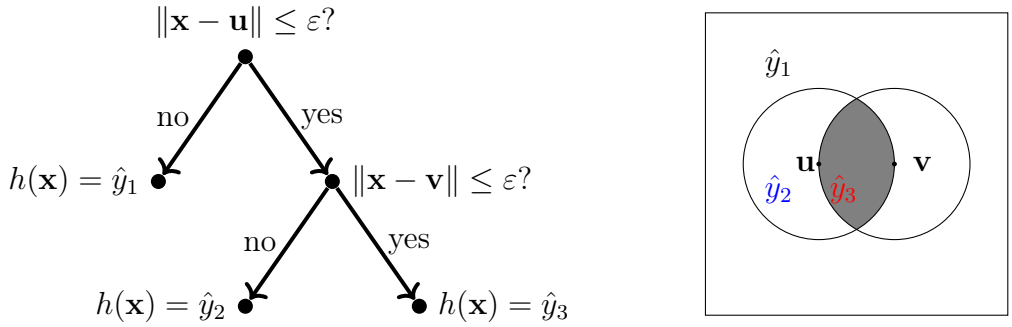


Figure 5: Left: A decision tree is a flow-chart-like representation of a piece-wise constant hipótesis $h : \mathcal{X} \to \mathbb{R}$. Each piece is a decision region $\mathcal{R}_{\hat{y}} := \big\{\mathbf{x} \in \mathcal{X} : h(\mathbf{x}) = \hat{y}\big\}$. The depicted decision tree can be applied to numeric vector de característicass, i.e., $\mathcal{X} \subseteq \mathbb{R}^d$. It is parametrized by the threshold $\varepsilon > 0$ and the vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$. Right: A decision tree partitions the espacio de características $\mathcal{X}$ into decision regions. Each decision region $\mathcal{R}_{\hat{y}} \subseteq \mathcal{X}$ corresponds to a specific leaf node in the decision tree.

denial-of-service attack A denial-of-service attack aims (e.g., via data poisoning) to steer the training of a model such that it performs poorly for typical punto de datoss.

density-based spatial clustering of applications with noise (DBSCAN) DBSCAN refers to a clustering algorithm for punto de datoss that are characterized by numeric vector de característicass. Like k-means and soft clustering via GMM, also DBSCAN uses the Euclidean distances between vector de característicass to determine the clusters. However, in contrast to k-means and GMM, DBSCAN uses a different notion of similarity between punto de datoss. DBSCAN considers two punto de datoss as similar if they are connected via a sequence (path) of close-by

intermediate punto de datoss. Thus, DBSCAN might consider two punto de datoss as similar (and therefore belonging to the same cluster) even if their vector de características have a large Euclidean distance.

descenso por gradiente proyectado (GD proyectado) Consideremos un método basado en ERM que utiliza un model parametrizado con un parameter space $\mathcal{W} \subseteq \mathbb{R}^d$. Aun si la función objetivo de ERM es smooth, no podemos usar el GD básico, ya ya que este no toma en cuenta las restricciones sobre la variable de optimización (es decir, los model parameters). El GD proyectado extiende el GD básico para controlar restricciones sobre la variable de optimización(es decir, los model parameters). Una sola iteración del GD proyectado consiste primero en realizar un paso de gradiente y luego proyectar el resultado sobre el parameter space.

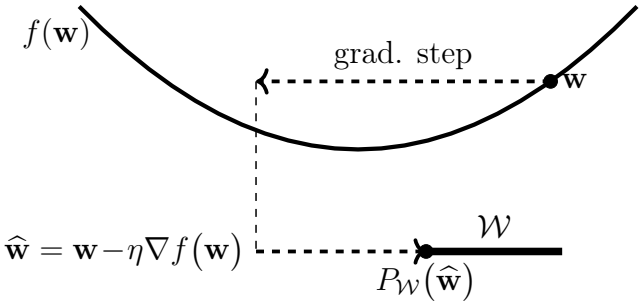


Figure 6: GD proyectado amplía un paso de gradiente básico con una proyección de regreso al conjunto de restricciones $\mathcal{W}$.

device Any physical system that can be used to store and process datos. In the context of ML, we typically mean a computer that is able to read

in punto de datoss from different sources and, in turn, to train an ML model using these punto de datoss.

diferenciable Una función real $f : \mathbb{R}^d \to \mathbb{R}$ es diferenciable si, en cualquier punto, puede aproximarse localmente mediante una función lineal. La aproximación lineal local en el punto $\mathbf{x}$ es determinada por el gradiente $\nabla f(\mathbf{x})$ [2].

dimensión de Vapnik–Chervonenkis (dimensión VC) La dimensión VC (Vapnik–Chervonenkis) de un hypothesis space infinito es una medida ampliamente utilizada para su tamaño. Nos referimos a la literatura (vea [37]) para una definición precisa de la dimensión VC, y para una discusión de sus propiedades básicas y su uso en ML.

dimensión efectiva La dimensión efectiva $d_{\text{eff}}(\mathcal{H})$ de un hypothesis space infinito $\mathcal{H}$ es una medida de su tamaño. A grandes rasgos,la dimensión efectiva es igual al número efectivo de model parameters independientes y ajustables. Estos parameters pueden ser los coeficientes utilizados en un mapa lineal o los weights y términos de sesgo de una ANN.

discrepancia Considere una aplicacion de FL con datos en red representada por un red de aprendizaje federado (red FL). Los métodos de FL utilizan una medida de discrepancia para comparar los mapas de hipótesis generados por los local models en los nodos i, i' conectados por una arista en el red de aprendizaje federado (red FL).

distribución normal multivariante La distribución normal multivariante $\mathcal{N}(\mathbf{m}, \mathbf{C})$ es una familia importante de probability distributions para

una RV continua $\mathbf{x} \in \mathbb{R}^d$ [5, 38, 39]. Esta familia está parametrizada por la media $\mathbf{m}$ y la covariance matrix $\mathbf{C}$ de $\mathbf{x}$. Si la covariance matrix es invertible, la probability distribution de $\mathbf{x}$ es

$$p(\mathbf{x}) \propto \exp\Big(-(1/2)\big(\mathbf{x}-\mathbf{m}\big)^T \mathbf{C}^{-1}\big(\mathbf{x}-\mathbf{m}\big)\Big).$$

edge weight Each edge $\{i, i'\}$ of an red de aprendizaje federado (red FL) is assigned a non-negative edge weight $A_{i,i'} \geq 0$. A zero edge weight $A_{i,i'} = 0$ indicates the absence of an edge between nodes $i, i' \in \mathcal{V}$.

eigenvalue We refer to a number $\lambda \in \mathbb{R}$ as an eigenvalue of a square matrix $\mathbf{A} \in \mathbb{R}^{d \times d}$ if there is a non-zero vector $\mathbf{x} \in \mathbb{R}^d \setminus \{\mathbf{0}\}$ such that $\mathbf{A}\mathbf{x} = \lambda \mathbf{x}$.

eigenvalue decomposition (EVD) The eigenvalue decomposition for a square matrix $\mathbf{A} \in \mathbb{R}^{d \times d}$ is a factorization of the form

$$\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}.$$

The columns of the matrix $\mathbf{V} = \big(\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(d)}\big)$ are the eigenvectors of the matrix $\mathbf{V}$. The diagonal matrix $\mathbf{\Lambda} = \mathrm{diag}\big\{\lambda_1, \ldots, \lambda_d\big\}$ contains the eigenvalues λ_j corresponding to the eigenvectors $\mathbf{v}^{(j)}$. Note that the above decomposition exists only if the matrix $\mathbf{A}$ is diagonalizable.

eigenvector An eigenvector of a matrix $\mathbf{A} \in \mathbb{R}^{d \times d}$ is a non-zero vector $\mathbf{x} \in \mathbb{R}^d \setminus \{\mathbf{0}\}$ such that $\mathbf{A}\mathbf{x} = \lambda \mathbf{x}$ with some eigenvalue λ.

embudo de privacidad El embudo de privacidad es un método para aprender features amigables con la privacidad de los punto de datoss [40].

entorno El entorno de un nodo $i \in \mathcal{V}$ es el subconjunto de nodos constituido por los vecinos de i.

error cuadrático medio de estimación (MSEE) Consideremos un método de ML que aprende model parameters $\widehat{\mathbf{w}}$ a partir de un conjunto de datos $\mathcal{D}$. Si interpretamos los punto de datoss en $\mathcal{D}$ como realizacións i.i.d. de una RV $\mathbf{z}$, definimos el error de estimación como $\Delta\mathbf{w} := \widehat{w} - \overline{\mathbf{w}}$. Aquí, $\overline{\mathbf{w}}$ representa los verdaderos model parameters de la probability distribution de $\mathbf{z}$. El error cuadrático medio de estimación se define como la expectation $\mathbb{E}\{\|\Delta\mathbf{w}\|^2\}$ del cuadrado de la norma euclidiana del error de estimación [22, 41].

error de entrenamiento El loss promedio de una hipótesis al predecir las etiquetas de los punto de datoss en un conjunto de entrenamiento. A veces también nos referimos al error de entrenamiento como el loss promedio mínimo que se logra mediante una solución de ERM.

error de estimación Consideremos punto de datoss, cada uno con un vector de características $\mathbf{x}$ y una etiqueta y. En algunas aplicaciones, podemos modelar la relación entre el vector de características y la etiqueta de un punto de datos como $y = \bar{h}(\mathbf{x}) + \varepsilon$. Aquí $\bar{h}$ representa la hipótesis verdadera subyacente y ε es un término de ruido que resume errores de modelado o etiquetado. El error de estimación incurrido por un método de ML que aprende una hipótesis $\widehat{h}$, por ejemplo usando ERM, se define como $\widehat{h}(\mathbf{x}) - \bar{h}(\mathbf{x})$, para algún vector de características dado. En un hypothesis space paramétrico, donde las hipótesisse determinan mediante model parameters $\mathbf{w}$, podemos definir el error de estimación

como $\Delta\mathbf{w} = \widehat{\mathbf{w}} - \overline{\mathbf{w}}$ [41, 42].

error de validación Consideremos una hipótesis $\hat{h}$ obtenida por algún método de ML, e.g., por ejemplo, utilizando ERM en un conjunto de entrenamiento. El loss promedio de $\hat{h}$ en un conjunto de validación, que es diferente del conjunto de entrenamiento, se denomina error de validación.

espacio de características El espacio de características de una aplicación o método de ML está constituido por todos los valores potenciales que el vector de características de un punto de datos puede asumir. Una elección común para el espacio de características es el Euclidean space $\mathbb{R}^d$, donde la dimensión d es el número de features individuales de un punto de datos.

espacio de etiquetas Consideremos una aplicación de ML que involucra punto de datoss caracterizados por features y etiquetas. El espacio de etiqueta etiquetas está constituido por todos los valores potenciales que una etiqueta de un punto de datos puede asumir. Los métodos de Regresión, que buscan predecir etiquetas numéricas etiquetas, a menudo utilizan el espacio de etiquetas etiqueta $\mathcal{Y} = \mathbb{R}$. Los métodos de clasificación binaria utilizan un espacio de etiquetas etiqueta que consiste de dos elementos diferentes, por ejemplo, $\mathcal{Y} = \{-1, 1\}$, $\mathcal{Y} = \{0, 1\}$, o $\mathcal{Y} = \{$"imagen de gato","sin imagen de gato"$\}$.

espectrograma Un espectrograma representa la distribución tiempo-frecuencia de la energía de una señal temporal $x(t)$. Intuitivamente, cuantifica la cantidad de energía de la señal presentedentro de un segmento de

tiempo específico $[t_1, t_2] \subseteq \mathbb{R}$ y en un intervalo de frecuencia $[f_1, f_2] \subseteq \mathbb{R}$. Formalmente, el espectrograma de una señal se define como el módulo al cuadrado de su transformada de Fourier de ventana corta (STFT, en inglés) [43]. La Figure 7 muestra una señal temporal junto con su espectrograma.

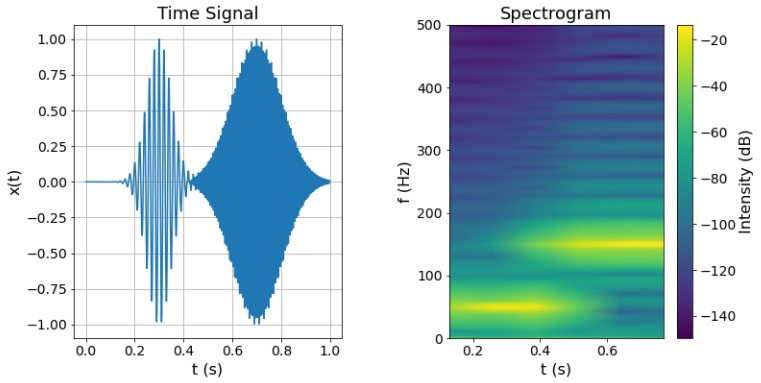


Figure 7: Izquierda: una señal temporal compuesta por dos pulsos gaussianos modulados. Derecha: representación de intensidad de su espectrograma.

La representación de intensidad del espectrograma puede considerarse como una imagen de la señal. Una estrategia sencilla para la clasificación de señales de audio consiste en introducir esta imagen en una red profunda desarrollada originalmente para tareas de clasificación de imágenes y detección de objetos [44]. Conviene señalar que, además del espectrograma, existen otras representaciones alternativas para describir la distribución tiempo-frecuencia de la energía de una señal [45, 46].

etiqueta Una es un hecho de nivel superior o una cantidad de interés asociada a un punto de datos. Por ejemplo, si el punto de datos es una imagen, la

etiqueta podría indicar si la imagen contiene un gato o no. Los sinónimos de etiqueta, comúnmente utilizados en dominios específicos, incluyen "variable de respuesta", "variable de salida" y "objetivo" [47], [48], [49].

Euclidean space The Euclidean space $\mathbb{R}^d$ of dimension $d \in \mathbb{N}$ consists of vectors $\mathbf{x} = \big(x_1, \ldots, x_d\big)$, with d real-valued entries $x_1, \ldots, x_d \in \mathbb{R}$. Such an Euclidean space is equipped with a geometric structure defined by the inner product $\mathbf{x}^T\mathbf{x}' = \sum_{j=1}^{d} x_j x'_j$ between any two vectors $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$ [2].

expectation Consider a numeric vector de características $\mathbf{x} \in \mathbb{R}^d$ which we interpret as the realización of an RV with a probability distribution $p(\mathbf{x})$. The expectation of $\mathbf{x}$ is defined as the integral $\mathbb{E}\{\mathbf{x}\} := \int \mathbf{x} p(\mathbf{x})$ [2,50,51]. Note that the expectation is only defined if this integral exists, i.e., if the RV is integrable.

expectation-maximization (EM) Consider a modelo probabilístico $p(\mathbf{z}; \mathbf{w})$ for the punto de datoss $\mathcal{D}$ generated in some ML application. The maximum likelihood estimator for the model parameters $\mathbf{w}$ is obtained by maximizing $p(\mathcal{D}; \mathbf{w})$. However, the resulting optimization problem might be computationally challenging. Expectation-maximization approximates the maximum likelihood estimator by introducing a latent RV $\mathbf{z}$ such that maximizing $p(\mathcal{D}, \mathbf{z}; \mathbf{w})$ would be easier [42,52,53]. Since we do not observe $\mathbf{z}$, we need to estimate it from the observed conjunto de datos $\mathcal{D}$ using a conditional expectation. The resulting estimate $\widehat{\mathbf{z}}$ is then used to compute a new estimate $\widehat{\mathbf{w}}$ by solving $\max_{\mathbf{w}} p(\mathcal{D}, \widehat{\mathbf{z}}; \mathbf{w})$. The crux is that the conditional expectation $\widehat{\mathbf{z}}$ depends on the model parameters $\widehat{\mathbf{w}}$, which we have updated based on $\widehat{\mathbf{z}}$. Thus, we have to

re-calculate $\widehat{\mathbf{z}}$, which, in turn, results in a new choice $\widehat{\mathbf{w}}$ for the model parameters. In practice, we repeat the computation of the conditional expectation (i.e., the E-step) and the update of the model parameters (i.e., the M-step) until some stopping criterion is met.

experto El ML tiene como objetivo aprender una hipótesis h que prediga con precisión la etiqueta de un punto de datos basado en sus features. Medimos el error de predicción utilizando una loss function. Idealmente, buscamos una hipótesis que incurra en la loss mínima para cualquier punto de datos. Podemos hacer este objetivo más preciso mediante el independent and identically distributed assumption (i.i.d. assumption) y utilizando el Bayes risk como referencia referencia (baseline) para la loss promedio de una hipótesis. Una manera alternativa de obtener unareferencia (baseline) es utilizar la hipótesis h' aprendida por un método de ML existente. A esta hipótesis h' la denominamos experto [16]. Los métodos de minimización de Arrepentimiento (regret) aprenden una hipótesis que incurre en una loss comparable a la del mejor experto [15, 16].

explainable empirical risk minimization (EERM) Explainable ERM is an instance of SRM that adds a regularization term to the average loss in the función objetivo of ERM. The regularization term is chosen to favor hipótesis maps that are intrinsically explainable for a specific user. This user is characterized by their predicciónss provided for the punto de datoss in a conjunto de entrenamiento [54].

explainable machine learning (explainable ML) Explainable ML meth-

ods aim at complementing each predicción with an explicación of how the predicción has been obtained. The construction of an explicit explicación might not be necessary if the ML method uses a sufficiently simple (or interpretable) model [55].

explicabilidad Definimos la (subjetiva) explicabilidad de un método de ML como el nivel de simulabilidad [56] de las predicciónes entregadas por un sistema de ML a un usuario humano. Se pueden construir medidas cuantitativas para la explicabilidad (subjetiva) de un model entrenado comparando sus predicciónes con las predicciónes proporcionadas por un usuario en un conjunto de prueba [54, 56]. Alternativamente, podemos usar modelo probabilísticos para los datos y medir la explicabilidad de un model de ML entrenado mediante la entropía condicional (diferencial) de sus predicciónes, dadas las predicciónes del usuario [57, 58].

Explicaciones Locales Interpretables e Independientes del Modelo (LIME) Consideremos un model entrenado (o una hipótesis aprendida) $\widehat{h} \in \mathcal{H}$, que asigna la vector de características de un punto de datos a la predicción $\widehat{y} = \widehat{h}$. Las explicaciones Locales Interpretables e Independientes del Modelo (LIME) son una tecnica para explicar el comportamiento de $\widehat{h}$, localmente, alrededor de un punto de datos con vector de características $\mathbf{x}^{(0)}$ [59]. La explicación se da en forma de una aproximación local $g \in \mathcal{H}'$ de $\widehat{h}$ (véa Fig.). Esta aproximación puede obtenerse mediante una instancia de ERM con un conjunto de entrenamiento diseñado cuidadosamente. En particular, el conjunto de entrenamiento consiste en punto de datoss con vector de características $\mathbf{x}$ cercana a

$\mathbf{x}^{(0)}$ y la (pseudo-)etiqueta $\widehat{h}(\mathbf{x})$. Nótese que podemos utilizar un model $\mathcal{H}'$ diferente para la aproximación que el model original $\mathcal{H}$. Por ejemplo, podemos usar un decision tree para aproximar (localmente) una red profunda. Otra elección ampliamente utilizada para $\mathcal{H}'$ es el modelo lineal.

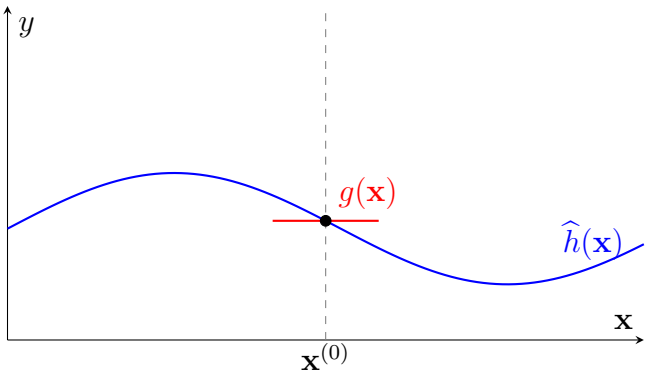


Figure 8: Para explicar un model $\widehat{h} \in \mathcal{H}$ entrenado, alrededor de un vector de características $\mathbf{x}^{(0)}$, podemos usar una aproximación local $g \in \mathcal{H}'$.

explicación Un manera para hacer que los métodos de ML sean transparentes consiste en proporcionar una explicación junto con la predicción generada por el método ML. Las explicaciones pueden adoptar muchas formas diferentes. Pueden ser un texto natural o una medida cuantitativa que indique la importancia de características individuales de un punto de datos [60]. También podemos usar formas visuales, como los mapas de intensidad usados en tareas de clasificación de imágenes [61].

feature A feature of a punto de datos is one of its properties that can be measured or computed easily without the need for human supervision.

For example, if a punto de datos is a digital image (e.g., stored as a `.jpeg` file), then we could use the red-green-blue intensities of its pixels as features. Domain-specific synonyms for the term feature are "covariate," "explanatory variable," "independent variable," "input (variable)," "predictor (variable)," or "regressor" [47], [48], [49].

feature map Feature map refers to a map that transforms the original features of a punto de datos into new features. The so-obtained new features might be preferable over the original features for several reasons. For example, the arrangement of punto de datoss might become simpler (or more linear) in the new espacio de características, allowing the use of modelo lineals in the new features. This idea is a main driver for the development of kernel methods [62]. Moreover, the hidden layers of a red profunda can be interpreted as a trainable feature map followed by a modelo lineal in the form of the output layer. Another reason for learning a feature map could be that learning a small number of new features helps to avoid sobreajuste and ensures interpretabilidad [59]. The special case of a feature map delivering two numeric features is particularly useful for datos visualization. Indeed, we can depict punto de datoss in a scatterplot by using two features as the coordinates of a punto de datos.

feature matrix Consider a conjunto de datos $\mathcal{D}$ with m punto de datoss with vector de característicass $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)} \in \mathbb{R}^d$. It is convenient to collect the individual vector de característicass into a feature matrix $\mathbf{X} := \big(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}\big)^T$ of size $m \times d$.

federated learning (FL) FL is an umbrella term for ML methods that train models in a collaborative fashion using decentralized datos and computation.

FedProx FedProx se refiere a un algorithm iterativo de FL que alterna entre entrenar local models por separado y combinar los model parameters locales actualizados. A diferencia de FedAvg, que utiliza stochastic gradient descent (SGD) para entrenar los local models, FedProx usa un operador proximal para el entrenamiento [63].

filtración de privacidad Consideremos una aplicacion de ML que procesa un conjunto de datos $\mathcal{D}$ y produce una salida, como las predicciónes obtenidas para nuevos punto de datoss. Se produce una filtración de privacidad cuando la salida contiene información privada sobre un feature de un punto de datos (que podría representar a una persona) en $\mathcal{D}$. Basado en la modelo probabilístico para la generacion de los datos, podemos medir la filtración de privacidad usando la MI entre la salida y la feature sensible. Otra medida cuantitativa de la filtración de privacidad es la privacidad diferencial (DP). Las relaciones entre las diferentes medidas de filtración de privacidad han sido estudiadas en la literatura (véa [64]).

Finnish Meteorological Institute (FMI) The FMI is a government agency responsible for gathering and reporting weather datos in Finland.

fuertemente convexa Una función real diferenciable de valor continuo $f(\mathbf{x})$es fuertemente convex con coeficiente σ si cumple: $f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x}) + (\sigma/2)\left\|\mathbf{y}-\mathbf{x}\right\|_2^2$ [65], [66, Sec. B.1.1].

función cuadrática Una función $f : \mathbb{R}^d \to \mathbb{R}$ de la forma

$$f(\mathbf{w}) = \mathbf{w}^T \mathbf{Q} \mathbf{w} + \mathbf{q}^T \mathbf{w} + a,$$

donde $\mathbf{Q} \in \mathbb{R}^{d \times d}$ es una matriz, $\mathbf{q} \in \mathbb{R}^d$ es un vector y $a \in \mathbb{R}$ es un escalar.

función de activación Cada neurona artificial dentro de una ANN se le asigna una función de activación $\sigma(\cdot)$ que transforma una combinación ponderada de sus entradas $x_1, \ldots, x_d$ en un único valor de salida: $a = \sigma\big(w_1 x_1 + \ldots + w_d x_d\big)$. Cada neurona está parametrizada por los weights $w_1, \ldots, w_d$.

función objetivo Una función objetivo es un mapa que asigna a cada valor de una variable de optimización, como los model parameters $\mathbf{w}$ de una hipótesis $h^{(\mathbf{w})}$, un valor objetivo $f(\mathbf{w})$. El valor objetivo $f(\mathbf{w})$ podria ser el riesgo o el riesgo empírico de una hipótesis $h^{(\mathbf{w})}$.

Gaussian mixture model (GMM) A GMM is a particular type of modelo probabilístico for a numeric vector $\mathbf{x}$ (e.g., the features of a punto de datos). Within a GMM, the vector $\mathbf{x}$ is drawn from a randomly selected distribución normal multivariante $p^{(c)} = \mathcal{N}\left(\boldsymbol{\mu}^{(c)}, \mathbf{C}^{(c)}\right)$ with $c = I$. The index $I \in \{1, \ldots, k\}$ is an RV with probabilities $p(I = c) = p_c$. Note that a GMM is parametrized by the probability p_c, the media vector $\boldsymbol{\mu}^{(c)}$, and the covariance matrix $\boldsymbol{\Sigma}^{(\mathbf{c})}$ for each $c = 1, \ldots, k$. GMMs are widely used for clustering, density estimation, and as a generative model.

generalization Many current ML (and AI) systems are based on ERM: At their core, they train a model (i.e., learn a hipótesis $\hat{h} \in \mathcal{H}$) by minimizing the average loss (or riesgo empírico) on some punto de datoss $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}$, which serve as a conjunto de entrenamiento $\mathcal{D}^{(\text{train})}$. Generalization refers to an ML method's ability to perform well outside the conjunto de entrenamiento. Any mathematical theory of generalization needs some mathematical concept for the "outside the conjunto de entrenamiento." For example, statistical learning theory uses a modelo probabilístico such as the i.i.d. assumption for datos generation: the punto de datoss in the conjunto de entrenamiento are i.i.d. realizacións of some underlying probability distribution $p(\mathbf{z})$. A modelo probabilístico allows us to explore the outside of the conjunto de entrenamiento by drawing additional i.i.d. realizacións from $p(\mathbf{z})$. Moreover, using the i.i.d. assumption allows us to define the riesgo of a trained model $\hat{h} \in \mathcal{H}$ as the expected loss $\bar{L}\big(\hat{h}\big)$. What is more, we can use concentration bounds or convergence results for sequences of i.i.d. RVs to bound the deviation between the riesgo empírico $\widehat{L}\big(\hat{h}|\mathcal{D}^{(\text{train})}\big)$ of a trained model and its riesgo [37]. It is possible to study generalization also without using modelo probabilísticos. For example, we could use (deterministic) perturbations of the punto de datoss in the conjunto de entrenamiento to study its outside. In general, we would like the trained model to be robust, i.e., its prediccióńs should not change too much for small perturbations of a punto de datos. Consider a trained model for detecting an object in a smartphone snapshot. The detection result should not change if we mask a small number of randomly chosen pixels in the

image [67].

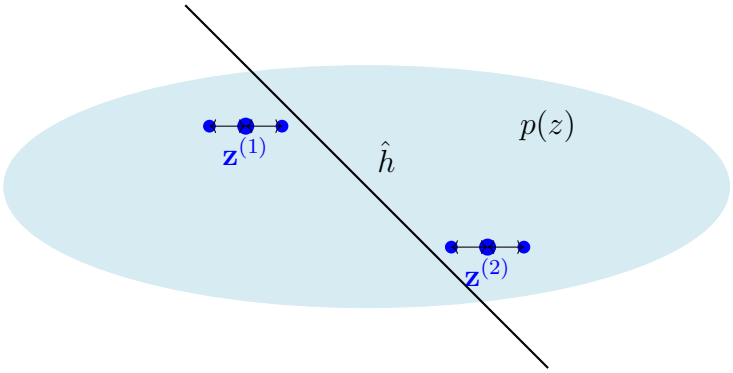


Figure 9: Two punto de datoss $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}$ that are used as a conjunto de entrenamiento to learn a hipótesis $\hat{h}$ via ERM. We can evaluate $\hat{h}$ outside $\mathcal{D}^{(\text{train})}$ either by an i.i.d. assumption with some underlying probability distribution $p(\mathbf{z})$ or by perturbing the punto de datoss.

generalized total variation (GTV) GTV is a measure of the variation of trained local models $h^{(i)}$ (or their model parameters $\mathbf{w}^{(i)}$) assigned to the nodes $i = 1, \ldots, n$ of an undirected weighted graph $\mathcal{G}$ with edges $\mathcal{E}$. Given a measure $d^{(h,h')}$ for the discrepancia between hipótesis maps h, h', the GTV is

$$\sum_{\{i,i'\} \in \mathcal{E}} A_{i,i'} d^{(h^{(i)}, h^{(i')})}.$$

Here, $A_{i,i'} > 0$ denotes the weight of the undirected edge $\{i, i'\} \in \mathcal{E}$.

gradient descent (GD) Gradiente descent is an iterative method for finding the mínimo of a diferenciable function $f(\mathbf{w})$ of a vector-valued argument $\mathbf{w} \in \mathbb{R}^d$. Consider a current guess or approximation $\mathbf{w}^{(k)}$ for the mínimo of the function $f(\mathbf{w})$. We would like to find a new (better) vector $\mathbf{w}^{(k+1)}$ that has a smaller objective value $f(\mathbf{w}^{(k+1)}) < f\big(\mathbf{w}^{(k)}\big)$ than the current guess $\mathbf{w}^{(k)}$. We can achieve this typically by using a paso de gradiente

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \eta \nabla f(\mathbf{w}^{(k)}) \tag{2}$$

with a sufficiently small step size $\eta > 0$. Figure 10 illustrates the effect of a single gradiente descent step (2).

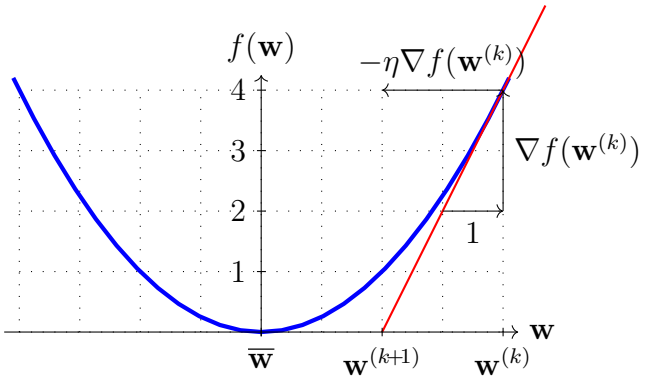


Figure 10: A single paso de gradiente (2) towards the minimizer $\overline{\mathbf{w}}$ of $f(\mathbf{w})$.

gradient-based methods Gradiente-based methods are iterative techniques for finding the mínimo (or máximo) of a diferenciable función objetivo of the model parameters. These methods construct a sequence of approximations to an optimal choice for model parameters that results in a mínimo (or máximo) value of the función objetivo. As their name indicates, gradiente-based methods use the gradientes of the función objetivo evaluated during previous iterations to construct new, (hopefully) improved model parameters. One important example of a gradiente-based method is GD.

gradiente Para una función de valor real $f : \mathbb{R}^d \to \mathbb{R} : \mathbf{w} \mapsto f(\mathbf{w})$, un vector $\mathbf{g}$ tal que $\lim_{\mathbf{w}\to\mathbf{w}'} \frac{f(\mathbf{w})-\left(f(\mathbf{w}')+\mathbf{g}^T(\mathbf{w}-\mathbf{w}')\right)}{\|\mathbf{w}-\mathbf{w}'\|} = 0$ se denomina gradiente de f en $\mathbf{w}'$. Si existe tal vector, se denota como $\nabla f(\mathbf{w}')$ o $\nabla f(\mathbf{w})\big|_{\mathbf{w}'}$ [2].

grado de nodo El grado $d^{(i)}$ de un nodo $i \in \mathcal{V}$ en un graph no dirigido, es el número de sus vecinos, es decir, $d^{(i)} := \left|\mathcal{N}^{(i)}\right|$.

grado de pertenencia El grado de pertenencia es un número que indica en qué medida un punto de datos pertenece a un cluster [6, Ch. 8]. Este grado puede interpretarse como una asignación blanda (*soft*) al cluster.Los métodos de Soft clustering pueden codificar el grado de pertenencia mediante un número real en el intervalo $[0, 1]$. El Hard clustering se obtiene como caso extremo, cuando el grado de pertenencia solo toma los valores 0 o 1.

grafo Un grafo $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ es un par compuesto por un conjunto de nodos $\mathcal{V}$ y un conjunto de aristas $\mathcal{E}$. En su forma màs general, un grafo se específica por una función que asigna a cada arista $e \in \mathcal{E}$ un par de nodos [68]. Un grupo importante de grafos son los grafos no dirigidos. Un grafo simple no dirigido es obtenida identificando cada arista $e \in \mathcal{E}$ con dos nodos diferentes $\{i, i'\}$. Los grafos etiquetados asignan un peso númerico especifico weights A_e a cada arista $e \in \mathcal{E}$.

grafo conexo Un graph no dirigido $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ es conexo si todo subconjunto no vacío $\mathcal{V}' \subset \mathcal{V}$ tiene al menos una arista que lo conecta con $\mathcal{V} \setminus \mathcal{V}'$.

hard clustering Hard clustering refers to the task of partitioning a given set of punto de datoss into (a few) non-overlapping clusters. The most widely used hard clustering method is k-means.

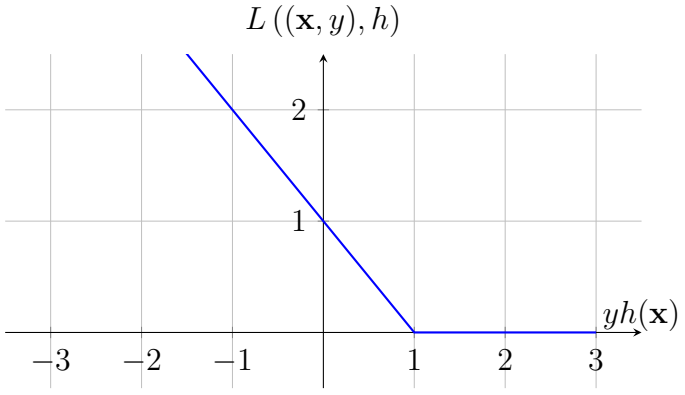
high-dimensional regime The high-dimensional regime of ERM is characterized by the dimensión efectiva of the model being larger than the sample size, i.e., the number of (labeled) punto de datoss in the conjunto de entrenamiento. For example, linear regression methods operate in

the high-dimensional regime whenever the number d of features used to characterize punto de datoss exceeds the number of punto de datoss in the conjunto de entrenamiento. Another example of ML methods that operate in the high-dimensional regime is large ANNs, which have far more tunable weights (and bias terms) than the total number of punto de datoss in the conjunto de entrenamiento. High-dimensional statistics is a recent main thread of probabilidad theory that studies the behavior of ML methods in the high-dimensional regime [69, 70].

Hilbert space A Hilbert space is a linear vector space equipped with an inner product between pairs of vectors. One important example of a Hilbert space is the Euclidean space $\mathbb{R}^d$, for some dimension d, which consists of Euclidean vectors $\mathbf{u} = \big(u_1, \ldots, u_d\big)^T$ along with the inner product $\mathbf{u}^T\mathbf{v}$.

hinge loss Consider a punto de datos characterized by a vector de características $\mathbf{x} \in \mathbb{R}^d$ and a binary etiqueta $y \in \{-1, 1\}$. The hinge loss incurred by a real-valued hipótesis map $h(\mathbf{x})$ is defined as

$$L\left((\mathbf{x}, y), h\right) := \max\{0, 1 - yh(\mathbf{x})\}. \tag{3}$$



A regularized variant of the hinge loss is used by the support vector machine (SVM) [71].

hipótesis Una hipótesis se refiere a un mapa (o función) $h : \mathcal{X} \to \mathcal{Y}$ que va del espacio de características $\mathcal{X}$ al espacio de etiquetas $\mathcal{Y}$. Dado un punto de datos con features $\mathbf{x}$, utilizamos un mapa de hipótesis h para estimar (o aproximar) la etiqueta y mediante la predicción $\hat{y} = h(\mathbf{x})$. El ML se centra en aprender (o encontrar) un mapa de hipótesis h tal que $y \approx h(\mathbf{x})$ para cualquier punto de datos (con features $\mathbf{x}$ y etiqueta y).

histograma Un histograma considera un conjunto de datos $\mathcal{D}$ que consiste en m punto de datoss $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}$, cada uno de los cuales pertenece a una celda $[-U, U] \times \ldots \times [-U, U] \subseteq \mathbb{R}^d$ con longitud de lado U. Dividimos esta celda uniformemente en celdas elementales más pequeñas con longitud de lado Δ. El histograma de $\mathcal{D}$ asigna a cada celda elemental la fracción correspondiente de punto de datoss en $\mathcal{D}$ que caen dentro de esa celda.

Huber loss The Huber loss unifies the pérdida de error cuadrático and the absolute error loss.

Huber regression Huber regresión refers to ERM-based methods that use the Huber loss as a measure of the predicción error. Two important special cases of Huber regresión are least absolute deviation regression and linear regression. Tuning the threshold parameter of the Huber loss allows the user to trade the robustness of the absolute error loss against the computational benefits of the smooth pérdida de error cuadrático.

hypothesis space Every practical ML method uses a hipótesis space (or model) $\mathcal{H}$. The hipótesis space of an ML method is a subset of all possible maps from the espacio de características to the espacio de etiquetas. The design choice of the hipótesis space should take into account available computational resources and aspectos estadísticos. If the computational infrastructure allows for efficient matrix operations, and there is an (approximately) linear relation between a set of features and a etiqueta, a useful choice for the hipótesis space might be the modelo lineal.

independent and identically distributed (i.i.d.) It can be useful to interpret punto de datoss $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}$ as realizacións of i.i.d. RVs with a common probability distribution. If these RVs are continuous-valued, their joint probability density function (pdf) is $p\big(\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}\big) = \prod_{r=1}^{m} p\big(\mathbf{z}^{(r)}\big)$, with $p(\mathbf{z})$ being the common marginal pdf of the underlying RVs.

independent and identically distributed assumption (i.i.d. assumption) The i.i.d. assumption interprets punto de datoss of a conjunto de datos as the realizacións of i.i.d. RVs.

inteligencia artificial confiable (IA confiable) Además de los aspectos computacionales y los aspectos estadísticos, un tercer aspecto principal en el diseño de métodos de ML es su confiabilidad [72]. La Unión Europea ha propuesto siete requisitos clave (KRs) para una AI confiable (que típicamente se basa en métodos de ML) [73]:

1) KR1 - Agencia y supervisión humana;

2) KR2 - Robustez técnica y seguridad;

3) KR3 - Privacidad y gobernanza de los datos;

4) KR4 - Transparencia;

5) KR5 - Diversidad, no discriminación y equidad;

6) KR6 - Bienestar social y ambiental;

7) KR7 - Responsabilidad.

interpretabilidad Un método de ML es interpretable por un usuario específico si puede anticipar adecuadamente las predicciónes entregadas por el método. La noción de interpretabilidad puede precisarse utilizando medidas cuantitativas de la incertidumbre sobre las predicciónes [57].

kernel Consider punto de datoss characterized by a vector de características $\mathbf{x} \in \mathcal{X}$ with a generic espacio de características $\mathcal{X}$. A (real-valued) kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ assigns each pair of vector de característicass $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ a real number $K\big(\mathbf{x}, \mathbf{x}'\big)$. The value $K\big(\mathbf{x}, \mathbf{x}'\big)$ is often interpreted as a measure for the similarity between $\mathbf{x}$ and $\mathbf{x}'$. Kernel methods use a kernel to transform the vector de características $\mathbf{x}$ to a new vector de características $\mathbf{z} = K\big(\mathbf{x}, \cdot\big)$. This new vector de características belongs to a linear espacio de características $\mathcal{X}'$ which is (in general) different from the original espacio de características $\mathcal{X}$. The espacio de características $\mathcal{X}'$ has a specific mathematical structure, i.e., it is a reproducing kernel Hilbert space [62, 71].

kernel method A kernel method is an ML method that uses a kernel K to map the original (raw) vector de características $\mathbf{x}$ of a punto de datos to a new (transformed) vector de características $\mathbf{z} = K\big(\mathbf{x}, \cdot\big)$ [62, 71]. The motivation for transforming the vector de característicass is that, by using a suitable kernel, the punto de datoss have a "more pleasant" geometry in the transformed espacio de características. For example, in a binary clasificación problem, using transformed vector de característicass $\mathbf{z}$ might allow us to use modelo lineals, even if the punto de datoss are not linearly separable in the original espacio de características (see Figure 11).

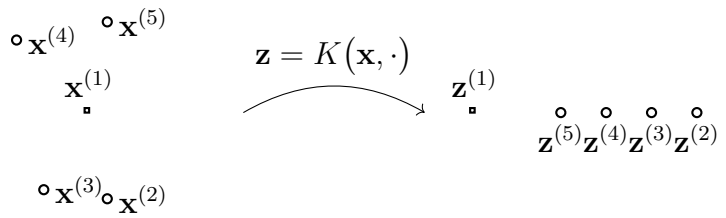


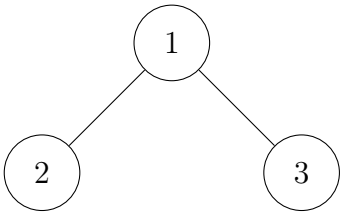
Figure 11: Five punto de datoss characterized by vector de característicass $\mathbf{x}^{(r)}$ and etiquetas $y^{(r)} \in \{\circ, \square\}$, for $r = 1, \ldots, 5$. With these vector de característicass, there is no way to separate the two classes by a straight line (representing the decision boundary of a clasificador lineal). In contrast, the transformed vector de característicass $\mathbf{z}^{(r)} = K\big(\mathbf{x}^{(r)}, \cdot\big)$ allow us to separate the punto de datoss using a clasificador lineal.

Kullback-Leibler divergence (KL divergence) The KL divergence is a quantitative measure of how much one probability distribution is different from another probability distribution [74].

Laplacian matrix The structure of a graph $\mathcal{G}$, with nodes $i = 1, \ldots, n$, can be analyzed using the properties of special matrices that are associated with $\mathcal{G}$. One such matrix is the graph Laplacian matrix $\mathbf{L}^{(\mathcal{G})} \in \mathbb{R}^{n \times n}$, which is defined for an undirected and weighted graph [8, 75]. It is defined element-wise as (see Figure 12)

$$L_{i,i'}^{(\mathcal{G})} := \begin{cases} -A_{i,i'} & \text{for } i \neq i', \{i, i'\} \in \mathcal{E}, \\ \sum_{i'' \neq i} A_{i,i''} & \text{for } i = i', \\ 0 & \text{else.} \end{cases} \tag{4}$$

Here, $A_{i,i'}$ denotes the edge weight of an edge $\{i, i'\} \in \mathcal{E}$.



$$\mathbf{L}^{(\mathcal{G})} = \begin{pmatrix} 2 & -1 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix}$$

Figure 12: Left: Some undirected graph $\mathcal{G}$ with three nodes $i = 1, 2, 3$. Right: The Laplacian matrix $\mathbf{L}^{(\mathcal{G})} \in \mathbb{R}^{3 \times 3}$ of $\mathcal{G}$.

large language model (LLM) Large language models is an umbrella term for ML methods that process and generate human-like text. These methods typically use red profundas with billions (or even trillions) of parameters. A widely used choice for the network architecture is referred to as Transformers [76]. The training of large language models is often based on the task of predicting a few words that are intentionally removed from a large text corpus. Thus, we can construct

punto de dato etiquetados simply by selecting some words of a text as etiquetas and the remaining words as features of punto de datoss. This construction requires very little human supervision and allows for generating sufficiently large conjunto de entrenamientos for large language models.

law of large numbers The law of large numbers refers to the convergence of the average of an increasing (large) number of i.i.d. RVs to the media of their common probability distribution. Different instances of the law of large numbers are obtained by using different notions of convergence [77].

learning rate Consider an iterative ML method for finding or learning a useful hipótesis $h \in \mathcal{H}$. Such an iterative method repeats similar computational (update) steps that adjust or modify the current hipótesis to obtain an improved hipótesis. One well-known example of such an iterative learning method is GD and its variants, SGD and descenso por gradiente proyectado (GD proyectado). A key parameter of an iterative method is the learning rate. The learning rate controls the extent to which the current hipótesis can be modified during a single iteration. A well-known example of such a parameter is the step size used in GD [6, Ch. 5].

least absolute deviation regression Least absolute deviation regression is an instance of ERM using the absolute error loss. It is a special case of Huber regression.

least absolute shrinkage and selection operator (Lasso) The Lasso is

an instance of SRM. It learns the weights $\mathbf{w}$ of a linear map $h(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ based on a conjunto de entrenamiento. Lasso is obtained from linear regression by adding the scaled ℓ_1-norma $\alpha \|\mathbf{w}\|_1$ to the average pérdida de error cuadrático incurred on the conjunto de entrenamiento.

linear regression Linear regresión aims to learn a linear hipótesis map to predict a numeric etiqueta based on the numeric features of a punto de datos. The quality of a linear hipótesis map is measured using the average pérdida de error cuadrático incurred on a set of punto de dato etiquetados, which we refer to as the conjunto de entrenamiento.

local dataset The concept of a local conjunto de datos is in between the concept of a punto de datos and a conjunto de datos. A local conjunto de datos consists of several individual punto de datoss, which are characterized by features and etiquetas. In contrast to a single conjunto de datos used in basic ML methods, a local conjunto de datos is also related to other local conjunto de datoss via different notions of similarity. These similarities might arise from modelo probabilísticos or communication infrastructure and are encoded in the edges of an red de aprendizaje federado (red FL).

local model Consider a collection of local datasets that are assigned to the nodes of an red de aprendizaje federado (red FL). A local model $\mathcal{H}^{(i)}$ is a hypothesis space assigned to a node $i \in \mathcal{V}$. Different nodes might be assigned different hypothesis spaces, i.e., in general $\mathcal{H}^{(i)} \neq \mathcal{H}^{(i')}$ for different nodes $i, i' \in \mathcal{V}$.

logistic loss Consider a punto de datos characterized by the features $\mathbf{x}$ and

a binary etiqueta $y \in \{-1, 1\}$. We use a real-valued hipótesis h to predict the etiqueta y from the features $\mathbf{x}$. The logistic loss incurred by this predicción is defined as

$$L\left((\mathbf{x}, y), h\right) := \log(1 + \exp(-yh(\mathbf{x}))). \tag{5}$$

Carefully note that the expression (5) for the logistic loss applies only for the espacio de etiquetas $\mathcal{Y} = \{-1, 1\}$ and when using the thresholding rule (1).

logistic regression Logistic regresión learns a linear hipótesis map (or clasificador) $h(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ to predict a binary etiqueta y based on the numeric vector de características $\mathbf{x}$ of a punto de datos. The quality of a linear hipótesis map is measured by the average logistic loss on some punto de dato etiquetados (i.e., the conjunto de entrenamiento).

loss ML methods use a loss function $L\left(\mathbf{z}, h\right)$ to measure the error incurred by applying a specific hipótesis to a specific punto de datos. With a slight abuse of notation, we use the term loss for both the loss function L itself and the specific value $L\left(\mathbf{z}, h\right)$, for a punto de datos $\mathbf{z}$ and hipótesis h.

loss function A loss function is a map

$$L : \mathcal{X} \times \mathcal{Y} \times \mathcal{H} \to \mathbb{R}_+ : \big(\big(\mathbf{x}, y\big), h\big) \mapsto L\left((\mathbf{x}, y), h\right).$$

It assigns a non-negative real number (i.e., the loss) $L\left((\mathbf{x}, y), h\right)$ to a pair that consists of a punto de datos, with features $\mathbf{x}$ and etiqueta y, and a hipótesis $h \in \mathcal{H}$. The value $L\left((\mathbf{x}, y), h\right)$ quantifies the discrepancy between the true etiqueta y and the predicción $h(\mathbf{x})$. Lower (closer

to zero) values $L\left((\mathbf{x}, y), h\right)$ indicate a smaller discrepancy between predicción $h(\mathbf{x})$ and etiqueta y. Figure 13 depicts a loss function for a given punto de datos, with features $\mathbf{x}$ and etiqueta y, as a function of the hipótesis $h \in \mathcal{H}$.

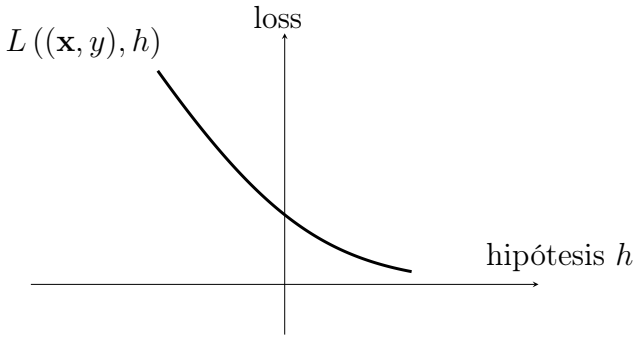


Figure 13: Some loss function $L\left((\mathbf{x}, y), h\right)$ for a fixed punto de datos, with vector de características $\mathbf{x}$ and etiqueta y, and a varying hipótesis h. ML methods try to find (or learn) a hipótesis that incurs minimal loss.

lote En el contexto de SGD, un lote se refiere a un subconjunto elegido al azar del conjunto total de conjunto de entrenamiento. Utilizamos los punto de datoss de este subconjunto para estimar el gradiente del error de entrenamiento y, a su vez, actualizar los model parameters.

maximum likelihood Consider punto de datoss $\mathcal{D} = \left\{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}\right\}$ that are interpreted as the realizacións of i.i.d. RVs with a common probability distribution $p(\mathbf{z}; \mathbf{w})$ which depends on the model parameters $\mathbf{w} \in \mathcal{W} \subseteq \mathbb{R}^n$. Máximo likelihood methods learn model parameters $\mathbf{w}$ by maximizing the probability (density) $p(\mathcal{D}; \mathbf{w}) = \prod_{r=1}^{m} p(\mathbf{z}^{(r)}; \mathbf{w})$ of

the observed conjunto de datos. Thus, the máximo likelihood estimator is a solution to the optimization problem $\max_{\mathbf{w} \in \mathcal{W}} p(\mathcal{D}; \mathbf{w})$.

media La expectation $\mathbb{E}\{\mathbf{x}\}$ de una RV numérica $\mathbf{x}$.

minimización de variación total generalizada (GTVMin) La minimización de variación total generalizada (GTVMin) es una instancia de regularized empirical risk minimization (RERM) que utiliza la GTV de los model parameters locales como un regularizador [78].

minimización empírica del riesgo (ERM) La minimización del Riesgo empírico es el problema de optimización que consiste en encontrar una hipótesis (dentro de un model) con la mínimo loss promedio (o riesgo empírico) en un conjunto de datos dado $\mathcal{D}$ (es decir, el conjunto de entrenamiento). Muchos métodos de ML se obtienen a partir de la riesgo empírico mediante elecciones específicas de diseño para el conjunto de datos, el model, y la loss [6, Ch. 3].

model In the context of ML methods, the term model typically refers to the hypothesis space employed by an ML method [6, 37].

model parameters Model parameters are quantities that are used to select a specific hipótesis map from a model. We can think of a list of model parameters as a unique identifier for a hipótesis map, similar to how a social security number identifies a person in Finland.

model selection En ML, la selección de modelo se refiere al proceso de elegir entre diferentes models candidatos. En su forma más básica, la selección de modelo consiste en: 1) entrenar cada model candidato; 2)

calcular el error de validación para cada model entrenado; y 3) elegir el model con el menor error de validación [6, Ch. 6].

modelo en red Un modelo en red sobre un red de aprendizaje federado (red FL) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ asigna un local model (es decir, un hypothesis space) a cada nodo $i \in \mathcal{V}$ del red de aprendizaje federado (red FL) $\mathcal{G}$.

modelo estocástico de bloques (SBM) El modelo estocástico de bloques (SBM) es un model generativo probabilístico para un graph no dirigido $\mathcal{G} = \big(\mathcal{V}, \mathcal{E}\big)$ con conjunto de nodos $\mathcal{V}$ [79]. En su forma básica, asigna aleatoriamente cada nodo $i \in \mathcal{V}$ a una cluster $c_i \in \{1, \ldots, k\}$. Cada par de nodos distintos se conecta con probabilidad $p_{i,i'}$ que depende únicamente de sus etiquetas c_i y $c_{i'}$. La presencia de aristas entre pares de nodos es estadísticamente independiente.

modelo lineal Consideremos punto de datoss, cada uno caracterizado por una vector de características numérica $\mathbf{x} \in \mathbb{R}^d$. Un model lineal es un hypothesis space que consiste en todos los mapeos lineales,

$$\mathcal{H}^{(d)} := \left\{ h(\mathbf{x}) = \mathbf{w}^T \mathbf{x} : \mathbf{w} \in \mathbb{R}^d \right\}. \tag{6}$$

Nótese que (6) define toda una familia de hypothesis spaces, parametrizada por el número d de features que se combinan linealmente para formar la predicción $h(\mathbf{x})$. La elección de diseño de d se guía por aspectos computacionales (por ejemplo, reducir d implica menor computación), por aspectos estadísticos (por ejemplo, aumentar d podría reducir el error de predicción), y por la interpretabilidad. Un model lineal que utiliza pocas features cuidadosamente elegidas suele considerarse más interpretable [55, 59].

modelo probabilístico Un model probabilístico interpreta los punto de datoss como realizaciónes de RVs con una probability distribution conjunta. Esta probability distribution conjunta típicamente incluye parameters que deben seleccionarse manualmente or o aprenderse usando métodos de inferencia estadística como la estimación por maximum likelihood [22].

multi-armed bandit A multi-armed bandit (MAB) problem models a repeated decision-making scenario in which, at each time step k, a learner must choose one out of several possible actions, often referred to as arms, from a finite set $\mathcal{A}$. Each arm $a \in \mathcal{A}$ yields a stochastic reward $r^{(a)}$ drawn from an unknown probability distribution with media $\mu^{(a)}$. The learner's goal is to maximize the cumulative reward over time by strategically balancing exploration (gathering information about uncertain arms) and exploitation (selecting arms known to perform well). This balance is quantified by the notion of arrepentimiento (regret), which measures the performance gap between the learner's strategy and the optimal strategy that always selects the best arm. MAB problems form a foundational model in online learning, reinforcement learning, and sequential experimental design [80].

mutual information (MI) The MI $I(\mathbf{x}; y)$ between two RVs $\mathbf{x}$, y defined on the same probability space is given by [74]

$$I(\mathbf{x}; y) := \mathbb{E}\left\{\log \frac{p(\mathbf{x}, y)}{p(\mathbf{x})p(y)}\right\}.$$

It is a measure of how well we can estimate y based solely on $\mathbf{x}$. A large value of $I(\mathbf{x}; y)$ indicates that y can be well predicted solely from

$\mathbf{x}$. This predicción could be obtained by a hipótesis learned by an ERM-based ML method.

máximo El máximo de un conjunto $\mathcal{A} \subseteq \mathbb{R}$ de números reales es el elemento más grande en ese conjunto, si tal elemento existe. Un conjunto $\mathcal{A}$ tiene un máximo si está acotado superiormente y alcanza su supremo (o mínimo de las cotas superiores) [2, Sec. 1.4].

mínimo Dado un conjunto de números reales, el mínimo es el menor de esos números.

networked exponential families (nExpFam) A collection of exponential families, each of them assigned to a node of an red de aprendizaje federado (red FL). The model parameters are coupled via the network structure by requiring them to have a small GTV [81].

non-smooth We refer to a function as non-smooth if it is not smooth [65].

norma Una norma es una función que asigna a cada elemento (vectorial) de un espacio vectorial lineal un número real no negativo. Esta función debe ser homogénea, definida positiva y debe cumplir la desigualdad triangular [82].

número de condición El número de condición $\kappa(\mathbf{Q}) \geq 1$ de una matriz definida positiva $\mathbf{Q} \in \mathbb{R}^{d \times d}$ es el cociente α/β entre el mayor α y el menor β eigenvalue de $\mathbf{Q}$. El número de condición es útil para el análisis de métodos de ML. La complejidad computacional de los gradient-based methods para linear regression depende críticamente del número de

condición de la matriz $\mathbf{Q} = \mathbf{X}\mathbf{X}^T$, donde $\mathbf{X}$ es la feature matrix del conjunto de entrenamiento. Es por eso que desde una perspectiva computacional, preferimos features de los punto de datoss que hagan que $\mathbf{Q}$ tenga un número de condición cercano a 1.

online gradient descent (online GD) Consider an ML method that learns model parameters $\mathbf{w}$ from some parameter space $\mathcal{W} \subseteq \mathbb{R}^d$. The learning process uses punto de datoss $\mathbf{z}^{(t)}$ that arrive at consecutive time-instants $t = 1, 2, \ldots$. Let us interpret the punto de datoss $\mathbf{z}^{(t)}$ as i.i.d. copies of an RV $\mathbf{z}$. The riesgo $\mathbb{E}\{L\left(\mathbf{z}, \mathbf{w}\right)\}$ of a hipótesis $h^{(\mathbf{w})}$ can then (under mild conditions) be obtained as the limit $\lim_{T\to\infty}(1/T)\sum_{t=1}^{T} L\left(\mathbf{z}^{(t)}, \mathbf{w}\right)$. We might use this limit as the función objetivo for learning the model parameters $\mathbf{w}$. Unfortunately, this limit can only be evaluated if we wait infinitely long in order to collect all punto de datoss. Some ML applications require methods that learn online: as soon as a new punto de datos $\mathbf{z}^{(t)}$ arrives at time t, we update the current model parameters $\mathbf{w}^{(t)}$. Note that the new punto de datos $\mathbf{z}^{(t)}$ contributes the component $L\left(\mathbf{z}^{(t)}, \mathbf{w}\right)$ to the riesgo. As its name suggests, online GD updates $\mathbf{w}^{(t)}$ via a (projected) paso de gradiente

$$\mathbf{w}^{(t+1)} := P_{\mathcal{W}}\big(\mathbf{w}^{(t)} - \eta_t \nabla_{\mathbf{w}} L\left(\mathbf{z}^{(t)}, \mathbf{w}\right)\big). \tag{7}$$

Note that (7) is a paso de gradiente for the current component $L\left(\mathbf{z}^{(t)}, \cdot\right)$ of the riesgo. The update (7) ignores all the previous components $L\left(\mathbf{z}^{(t')}, \cdot\right)$, for $t' < t$. It might therefore happen that, compared to $\mathbf{w}^{(t)}$, the updated model parameters $\mathbf{w}^{(t+1)}$ increase the retrospective average loss $\sum_{t'=1}^{t-1} L\left(\mathbf{z}^{(t')}, \cdot\right)$. However, for a suitably chosen learning rate η_t,

online GD can be shown to be optimal in practically relevant settings. By optimal, we mean that the model parameters $\mathbf{w}^{(T+1)}$ delivered by online GD after observing T punto de datoss $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(T)}$ are at least as good as those delivered by any other learning method [15, 83].

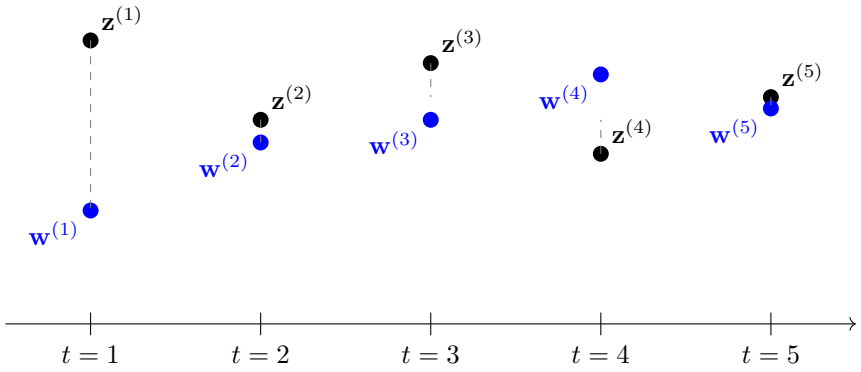


Figure 14: An instance of online GD that updates the model parameters $\mathbf{w}^{(t)}$ using the punto de datos $\mathbf{z}^{(t)} = x^{(t)}$ arriving at time t. This instance uses the pérdida de error cuadrático $L\left(\mathbf{z}^{(t)}, w\right) = (x^{(t)} - w)^2$.

operador proximal Dada una funcionconvex $f(\mathbf{w}')$, definimos su operador proximal como [84, 85]

$$\mathbf{prox}_{f(\cdot),\rho}(\mathbf{w}) := \underset{\mathbf{w}' \in \mathbb{R}^d}{\operatorname{argmin}} \left[f(\mathbf{w}') + (\rho/2) \left\| \mathbf{w} - \mathbf{w}' \right\|_2^2 \right] \text{ with } \rho > 0.$$

Como se ilustra en la Figura 15, evaluar el operador proximal equivale a minimizar una variante penalizada de $f(\mathbf{w}')$. El término de penalización es la distancia euclidiana cuadrada escalada hacia un vector dado $\mathbf{w}$ (que es la entrada del operador proximal). El operador proximal puede interpretarse como una generalization del paso de gradiente, definido

para una función smooth y convex $f(\mathbf{w}')$. De hecho, realizar un paso de gradiente con step size η en el vector actual $\mathbf{w}$ es lo mismo que aplicar el operador proximalde la función $\tilde{f}(\mathbf{w}') = \big(\nabla f(\mathbf{w})\big)^T(\mathbf{w}' - \mathbf{w})$ y usar $\rho = 1/\eta$.

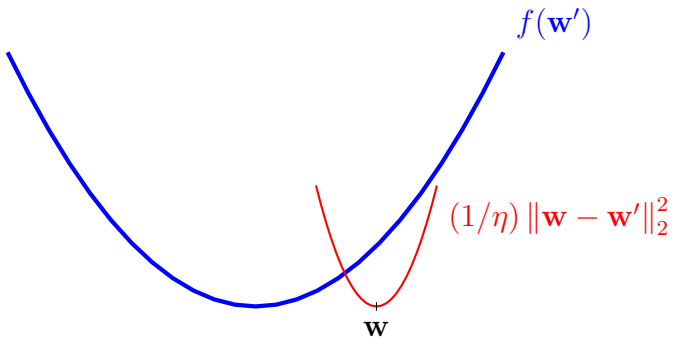


Figure 15: Un paso de gradiente generalizado actualiza un vector $\mathbf{w}$ minimizando una versión penalizada de la función $f(\cdot)$. El término de penalización es la distancia euclidiana cuadrada escalada entre la variable de optimización $\mathbf{w}'$ y el vector dado $\mathbf{w}$.

optimismo ante la incertidumbre Los metodos de ML aprenden model parameters $\mathbf{w}$ de acuerdo con algún criterio de desempeño $\bar{f}(\mathbf{w})$. Sin embargo, normalmente no pueden acceder directamente a $\bar{f}(\mathbf{w})$ pero dependen de una estimación (o aproximación) de $f(\mathbf{w})$. Por ejemplo, los métodos basados en ERM usan la loss promedio en un conjunto de datos (por ejemplo, el conjunto de entrenamiento) como estimación del riesgo de una hipótesis. Usando un modelo probabilístico, se puede construir un intervalo de confianza. $\big[l^{(\mathbf{w})}, u^{(\mathbf{w})}\big]$ para cada elección $\mathbf{w}$ de los model parameters. Una construcción simple es $l^{(\mathbf{w})} := f(\mathbf{w}) - \sigma/2$,

$u^{(\mathbf{w})} := f(\mathbf{w}) + \sigma/2$, donde σ representa una medida de la desviación entre $f(\mathbf{w})$ y $\bar{f}(\mathbf{w})$. También se pueden usar otras construcciones del intervalo, mientras se aseguren que $\bar{f}(\mathbf{w}) \in \left[l^{(\mathbf{w})}, u^{(\mathbf{w})}\right]$ con un probabilidad suficientemente alta. Siendo optimistas, elegimos los model parameters según el valor más favorable - pero realista - del criterio de desempeño $\tilde{f}(\mathbf{w}) := l^{(\mathbf{w})}$. Dos ejemplos de métodos de ML que usan una construcción optimista de una función objetivo son métodos de SRM [37, Ch. 11] y upper confidence bound (UCB) para decisiones secuenciales [80, Sec. 2.2].

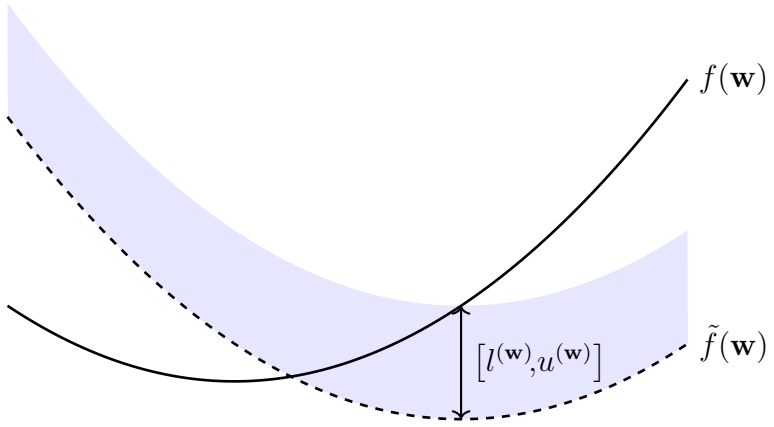


Figure 16: Los métodos de ML aprenden model parameters $\mathbf{w}$ usando una estimación de $f(\mathbf{w})$ como aproximación del criterio de desempeño $\bar{f}(\mathbf{w})$. Usando un modelo probabilístico, se pueden construir intervalos de confianza $\left[l^{(\mathbf{w})}, u^{(\mathbf{w})}\right]$ que contienen $\bar{f}(\mathbf{w})$ con alta probabilidad. La mejor medida plausible del desempeño para una elección especifica $\mathbf{w}$ es $\tilde{f}(\mathbf{w}) := l^{(\mathbf{w})}$.

outlier Many ML methods are motivated by the i.i.d. assumption, which

interprets punto de datoss as realizacións of i.i.d. RVs with a common probability distribution. The i.i.d. assumption is useful for applications where the statistical properties of the datos generation process are stationary (or time-invariant) [86]. However, in some applications the datos consists of a majority of regular punto de datoss that conform with an i.i.d. assumption as well as a small number of punto de datoss that have fundamentally different statistical properties compared to the regular punto de datoss. We refer to a punto de datos that substantially deviates from the statistical properties of most punto de datoss as an outlier. Different methods for outlier detection use different measures for this deviation. Stastistical learning theory studies fundamental limits on the ability to mitigate outliers reliably [87, 88].

parameter space The parameter space $\mathcal{W}$ of an ML model $\mathcal{H}$ is the set of all feasible choices for the model parameters (see Figure 17). Many important ML methods use a model that is parametrized by vectors of the Euclidean space $\mathbb{R}^d$. Two widely used examples of parametrized models are modelo lineals and red profundas. The parameter space is then often a subset $\mathcal{W} \subseteq \mathbb{R}^d$, e.g., all vectors $\mathbf{w} \in \mathbb{R}^d$ with a norma smaller than one.

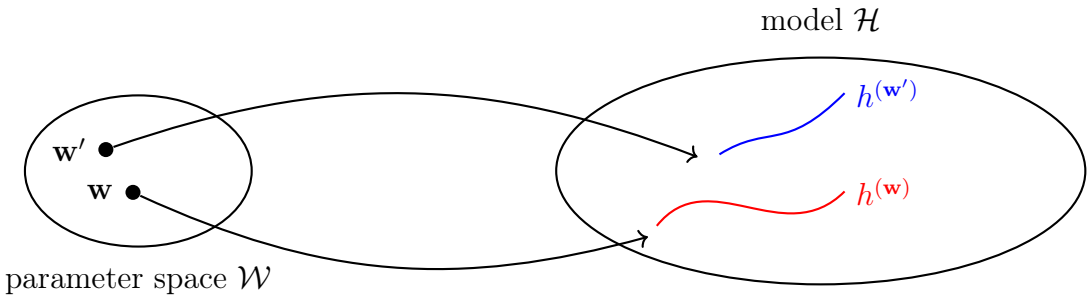


Figure 17: The parameter space $\mathcal{W}$ of an ML model $\mathcal{H}$ consists of all feasible choices for the model parameters. Each choice $\mathbf{w}$ for the model parameters selects a hipótesis map $h^{(\mathbf{w})} \in \mathcal{H}$.

parameters The parameters of an ML model are tunable (i.e., learnable or adjustable) quantities that allow us to choose between different hipótesis maps. For example, the modelo lineal $\mathcal{H} := \{h^{(\mathbf{w})} : h^{(\mathbf{w})}(x) = w_1 x + w_2\}$ consists of all hipótesis maps $h^{(\mathbf{w})}(x) = w_1 x + w_2$ with a particular choice for the parameters $\mathbf{w} = \big(w_1, w_2\big)^T \in \mathbb{R}^2$. Another example of parameters is the weights assigned to the connections between neurons of an ANN.

paso de gradiente Dada una función diferenciable diferenciable de valores reales $f(\cdot) : \mathbb{R}^d \to \mathbb{R}$ y un vector $\mathbf{w} \in \mathbb{R}^d$, el paso de gradiente actualiza $\mathbf{w}$ sumándole el negativo escalado del gradiente $\nabla f(\mathbf{w})$ para obtener el nuevo vector (véase la Figura 18)

$$\widehat{\mathbf{w}} := \mathbf{w} - \eta \nabla f(\mathbf{w}). \tag{8}$$

Matemáticamente, el paso de gradiente es un operador (típicamente no lineal) $\mathcal{T}^{(f,\eta)}$ que está parametrizado por la función f y el step size η.

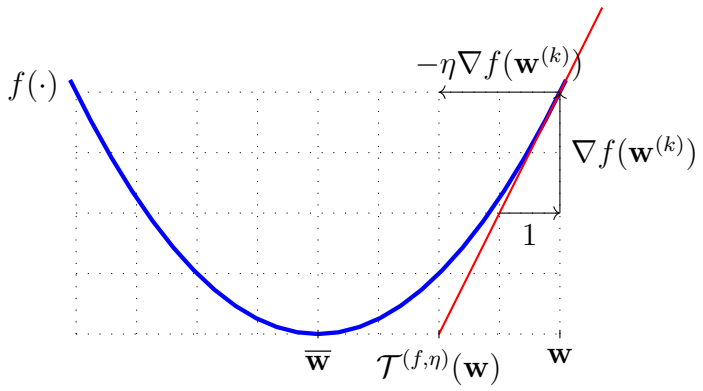


Figure 18: El paso básico de gradiente (8) mapea un vector $\mathbf{w}$ al vector actualizado $\mathbf{w}'$. Define un operador $\mathcal{T}^{(f,\eta)}(\cdot) : \mathbb{R}^d \to \mathbb{R}^d : \mathbf{w} \mapsto \widehat{\mathbf{w}}$.

Nótese que el paso de gradiente (8) optimiza localmente - en una entorno cuyo tamaño está determinado por el step size η - una aproximación lineal de la función $f(\cdot)$. Un generalization natural de (8) es optimizar localmente la función misma - en lugar de su aproximación lineal - de tal manera que:

$$\widehat{\mathbf{w}} = \underset{\mathbf{w}' \in \mathbb{R}^d}{\operatorname{argmin}} f(\mathbf{w}') + (1/\eta) \left\| \mathbf{w} - \mathbf{w}' \right\|_2^2 . \tag{9}$$

Intencionalmente usamos el mismo símbolo η para el parámetro en (9) que en el step size de (8). Mientras mayor sea el valor de η en (9), más progreso hará la actualización en la reducción del valor de la función $f(\widehat{\mathbf{w}})$. Nótese que, al igual que el paso de gradiente (8), la actualización (9) también define un operador (típicamente no lineal) parametrizado por la función $f(\cdot)$ y el parámetro η. Para una función convex $f(\cdot)$, este operador es conocido como el operador proximal de $f(\cdot)$ [84].

polynomial regression Polynomial regresión aims at learning a polynomial hipótesis map to predict a numeric etiqueta based on the numeric features of a punto de datos. For punto de datoss characterized by a single numeric feature, polynomial regresión uses the hypothesis space $\mathcal{H}_d^{(\text{poly})} := \{h(x) = \sum_{j=0}^{d-1} x^j w_j\}$. The quality of a polynomial hipótesis map is measured using the average pérdida de error cuadrático incurred on a set of punto de dato etiquetados (which we refer to as the conjunto de entrenamiento).

positive semi-definite (psd) A (real-valued) symmetric matrix $\mathbf{Q} = \mathbf{Q}^T \in \mathbb{R}^{d \times d}$ is referred to as psd if $\mathbf{x}^T \mathbf{Q} \mathbf{x} \geq 0$ for every vector $\mathbf{x} \in \mathbb{R}^d$. The property of being psd can be extended from matrices to (real-valued) symmetric kernel maps $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ (with $K(\mathbf{x}, \mathbf{x}') = K(\mathbf{x}', \mathbf{x})$) as follows: For any finite set of vector de características $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}$, the resulting matrix $\mathbf{Q} \in \mathbb{R}^{m \times m}$ with entries $Q_{r,r'} = K\big(\mathbf{x}^{(r)}, \mathbf{x}^{(r')}\big)$ is psd [62].

precisión (accuracy) Consideremos punto de datoss caracterizados porfeatures $\mathbf{x} \in \mathcal{X}$ y una etiqueta categórica y que toma valores de un conjunto finito espacio de etiquetas $\mathcal{Y}$. La precisión de una hipótesis $h : \mathcal{X} \to \mathcal{Y}$, cuando se aplica a los punto de datoss en un conjunto de datos $\mathcal{D} = \big\{\big(\mathbf{x}^{(1)}, y^{(1)}\big), \ldots, \big(\mathbf{x}^{(m)}, y^{(m)}\big)\big\}$, se define como $1 - (1/m) \sum_{r=1}^{m} L^{(0/1)}\left(\big(\mathbf{x}^{(r)}, y^{(r)}\big), h\right)$ usando la 0/1 loss $L^{(0/1)}(\cdot, \cdot)$.

predicción Una predicción es una estimación o aproximación de una cantidad de interés. El ML se centra en aprender o encontrar un mapa de hipótesis h que recibe las features $\mathbf{x}$ de un punto de datos and y produce una

predicción $\widehat{y} := h(\mathbf{x})$ para su etiqueta y.

predictor Un predictor es un mapa de hipótesis con valores reales. Dado unpunto de datos con features $\mathbf{x}$, el valor $h(\mathbf{x}) \in \mathbb{R}$ se utiliza como una predicción para la verdadera etiqueta numérica $y \in \mathbb{R}$ del punto de datos.

principal component analysis (PCA) PCA determines a linear feature map such that the new features allow us to reconstruct the original features with the mínimo reconstruction error [6].

privacidad diferencial (DP) Consideremos un método de ML $\mathcal{A}$ que recibe como entrada un conjunto de datos (por ejemplo, el conjunto de entrenamiento usado para ERM) y entrega una salida $\mathcal{A}(\mathcal{D})$. La salida puede ser los model parameters aprendidos o las predicciónes para ciertos punto de datoss. DP es una medida precisa de la filtración de privacidad ocasionada al revelar dicha salida. Aproximadamente, un método de ML es diferencialmente privado si la probability distribution de la salida $\mathcal{A}(\mathcal{D})$ no cambia significativamente cuando se modifica el atributo sensible de un solo punto de datos del conjunto de entrenamiento. Nótese que la DP se basa en un modelo probabilístico para un método de ML, es decir, interpretamos su salida $\mathcal{A}(\mathcal{D})$ como la realización de un RV. La aleatoriedad en la salida puede asegurarse añadiendo intencionalmente la realización de una RV auxiliar (ruido) a la salida del método de ML.

probabilidad Asignamos un valor de probabilidad, típicamente elegido en el intervalo $[0, 1]$, a cada evento que pueda ocurrir en un experimento aleatorio [5, 50, 51, 89].

probabilistic principal component analysis (PPCA) Probabilistic PCA extends basic PCA by using a modelo probabilístico for punto de datoss. The modelo probabilístico of probabilistic PCA reduces the task of dimensionality reduction to an estimation problem that can be solved using EM methods.

probability density function (pdf) The probabilidad density function $p(x)$ of a real-valued RV $x \in \mathbb{R}$ is a particular representation of its probability distribution. If the probabilidad density function exists, it can be used to compute the probabilidad that x takes on a value from a (measurable) set $\mathcal{B} \subseteq \mathbb{R}$ via $p(x \in \mathcal{B}) = \int_{\mathcal{B}} p(x')dx'$ [5, Ch. 3]. The probabilidad density function of a vector-valued RV $\mathbf{x} \in \mathbb{R}^d$ (if it exists) allows us to compute the probabilidad of $\mathbf{x}$ belonging to a (measurable) region $\mathcal{R}$ via $p(\mathbf{x} \in \mathcal{R}) = \int_{\mathcal{R}} p(\mathbf{x}')dx'_1 \ldots dx'_d$ [5, Ch. 3].

probability distribution To analyze ML methods, it can be useful to interpret punto de datoss as i.i.d. realizacións of an RV. The typical properties of such punto de datoss are then governed by the probabilidad distribution of this RV. The probabilidad distribution of a binary RV $y \in \{0, 1\}$ is fully specified by the probabilities $p(y = 0)$ and $p(y = 1) = 1 - p(y = 0)$. The probabilidad distribution of a real-valued RV $x \in \mathbb{R}$ might be specified by a pdf $p(x)$ such that $p(x \in [a, b]) \approx p(a)|b - a|$. In the most general case, a probabilidad distribution is defined by a probabilidad measure [38, 51].

probability space A probabilidad space is a mathematical model of a physical process (a random experiment) with an uncertain outcome. Formally,

a probabilidad space $\mathcal{P}$ is a triplet $(\Omega, \mathcal{F}, P)$ where

- Ω is a sample space containing all possible elementary outcomes of a random experiment;
- $\mathcal{F}$ is a sigma-algebra, a collection of subsets of Ω (called events) that satisfies certain closure properties under set operations;
- P is a probabilidad measure, a function that assigns a probabilidad $P(\mathcal{A}) \in [0, 1]$ to each event $\mathcal{A} \in \mathcal{F}$. The function must satisfy $P(\Omega) = 1$ and $P\left(\bigcup_{i=1}^{\infty} \mathcal{A}_i\right) = \sum_{i=1}^{\infty} P(\mathcal{A}_i)$ for any countable sequence of pairwise disjoint events $\mathcal{A}_1, \mathcal{A}_2, \ldots$ in $\mathcal{F}$.

Probabilidad spaces provide the foundation for defining RVs and to reason about uncertainty in ML applications [38, 51, 90].

promedio federado (FedAvg) El promedio federado (FedAvg)se refiere a un algorithm iterativo de FL que alterna entre entrenar local models por separado y combinar los model parameters locales actualizados. El entrenamiento de los local models se implementa a través de varios pasos de SGD [91].

protección de la privacidad Consideremos un método de ML $\mathcal{A}$ que recibe como entrada una conjunto de datos $\mathcal{D}$ y entega una salida $\mathcal{A}(\mathcal{D})$. La salida puede ser los model parameters aprendidos $\widehat{\mathbf{w}}$ o una predicción $\hat{h}(\mathbf{x})$ obtenida para un punto de datos específico con features $\mathbf{x}$. Muchas aplicaciones importantes de ML involucran punto de datoss que representan a personas. Cada punto de datos se caracteriza por features $\mathbf{x}$, posiblemente una etiqueta y, y un atributo sensible s (por ejemplo,

un diagnóstico medico). Mas o menos, la protección de la privacidad significa que debería ser imposible inferir, de la salida $\mathcal{A}(\mathcal{D})$, cualquier atributo sensibles de los punto de datoss en $\mathcal{D}$. Matemáticamente, la protección de privacidad requiere que el mapeo $\mathcal{A}(\mathcal{D})$ no sea invertible. En general, el solo hacer que $\mathcal{A}(\mathcal{D})$ no sea invertible no es suficiente. Necesitamos que sea suficientemente no invertible.

proximable Una funcion convex para la cual el operador proximal puede calcularse de manera eficiente se denomina a veces proximable o simple [92].

proyección Consideremos un subconjunto $\mathcal{W} \subseteq \mathbb{R}^d$ del Euclidean space de dimensión d. Definimos la proyección $P_{\mathcal{W}}\big(\mathbf{w}\big)$ de un vector $\mathbf{w} \in \mathbb{R}^d$ sobre $\mathcal{W}$ como

$$P_{\mathcal{W}}\big(\mathbf{w}\big) = \operatorname*{argmin}_{\mathbf{w}' \in \mathcal{W}} \left\|\mathbf{w} - \mathbf{w}'\right\|_2 . \tag{10}$$

En otras palabras, $P_{\mathcal{W}}\big(\mathbf{w}\big)$ es el vector en $\mathcal{W}$ más cercano a $\mathbf{w}$. La proyección está bien definida solo para aquellos subconjuntos $\mathcal{W}$ para los cuales existe el mínimo anterior [29].

punto de dato etiquetado Un punto de datos cuya etiqueta es conocida o ha sido determinada por algún medio, lo que podría requerir trabajo humano.

punto de datos Un punto de datos es cualquier objeto que transmite información [74]. Los puntos de datos pueden ser estudiantes, señales de radio, árboles, bosques, imágenes, RVs, números reales o proteínas. Caracterizamos los puntos de datos utilizando dos tipos de propiedades.

Un tipo de propiedad se denomina feature. Las Features son propiedades de un punto de datos que se pueden medir o calcular de manera automatizada. Un tipo diferente de propiedad se denomina etiqueta. La etiqueta de un punto de datos representa algún hecho de mayor nivel (o cantidad de interés). A diferencia de las features, determinar la etiqueta de un punto de datos suele requerir expertos humanos (expertos en dominio). En términos generales, el ML tiene como objetivo predecir la etiqueta of a datos de un punto de datos únicamente a partir de sus features.

pérdida de error cuadrático La loss de error cuadrático mide el error de predicción de una hipótesis h al predecir una etiqueta numérica $y \in \mathbb{R}$ a partir de las features $\mathbf{x}$ de un punto de datos. Se define como

$$L\left((\mathbf{x}, y), h\right) := \big(y - \underbrace{h(\mathbf{x})}_{=\hat{y}}\big)^2.$$

random forest A random forest is a set (or ensemble) of different decision trees. Each of these decision trees is obtained by fitting a perturbed copy of the original conjunto de datos.

realización Consideremos una RV x que asigna cada elemento (es decir, resultado o evento elemental) $\omega \in \mathcal{P}$ de un probability space $\mathcal{P}$ a un elemento a de un espacio medible $\mathcal{N}$ [2, 50, 51]. Una realización de x es cualquier elemento $a' \in \mathcal{N}$ tal que existe un elemento $\omega' \in \mathcal{P}$ con $x(\omega') = a'$.

red de aprendizaje federado (red FL) Una red federada es un graph no dirigido y ponderado, cuyos nodos representan generadores de datos que buscan entrenar un model local (o personalizado). Cada nodo de una red federada representa un device capaz de recopilar un local dataset y, a su vez, entrenar un local model. Los métodos de FL aprenden una hipótesis local $h^{(i)}$ para cada nodo $i \in \mathcal{V}$, de manera que incurra en una loss baja sobre su local dataset. ,first=red de aprendizaje federado (red FL),text=red FL

red profunda Una red profunda es una ANN con un número (relativamente) grande de capas ocultas. El aprendizaje profundo (deep learning) es un término general para los métodos de ML que utilizan una red profunda como model [93].

reducción de dimensionalidad Los métodos de reducción de dimensionalidad mapean (normalmente muchos) features originales a un conjunto (relativamente pequeño) de nuevos features. Estos métodos pueden utilizarse para visualizar punto de datoss aprendiendo dos features que sirvan como coordenadas de una representación en un scatterplot.

referencia (baseline) Consideremos un método de ML que produce una hipótesis aprendida (o un model entrenado) $\hat{h} \in \mathcal{H}$. Evaluamos la calidad del model entrenado mediante el cálculo de la loss promedio en un conjunto de prueba.Pero, ¿cómo saber si ese rendimiento es lo suficientemente bueno? ¿Cómo saber si el model entrenado se acerca al óptimo y si tiene sentido o no invertir más recursos (como recopilación de datos o potencia computacional) para mejorarlo? Para ello, es útil

contar con un valor de referencia (o *baseline*) con el cual comparar el rendimiento del modelo model entrenado. Este valor puede provenir del rendimiento humano, como la tasa de error de dermatólogos que diagnostican cáncer mediante inspección visual de la piel [94]. Otra fuente de referencia puede ser un método de ML ya existente que, por alguna razón, no sea adecuado para la aplicación (por ejemplo, por ser computacionalmente costoso), pero cuya tasa de error en el conjunto de prueba puede servir como baseline. Un enfoque más fundamentado para construir una baseline es utilizar un modelo probabilístico. En muchos casos, dado un modelo probabilístico modelo probabilístico $p(\mathbf{x}, y)$, podemos determinar con precisión el mínimo riesgo alcanzable entre todas las hipótesis (incluso aquellas que no pertenecen al hypothesis space $\mathcal{H}$) [22]. Este mínimo alcanzable se conoce como Bayes risk y corresponde al riesgo del Bayes estimator para la etiqueta y de un punto de datos, dados sus features $\mathbf{x}$. Dado un loss function específico, el Bayes estimator (si existe) está completamente determinado por la probability distribution $p(\mathbf{x}, y)$ [22, Cap. 4]. Calcular el Bayes estimator y el Bayes risk presenta dos desafíos principales:

1) La probability distribution $p(\mathbf{x}, y)$ desconocida y debe estimarse.
2) Incluso si se conoce $p(\mathbf{x}, y)$ calcular el Bayes risk puede ser computacionalmente muy costoso [95].

Un modelo probabilístico ampliamente utilizado es la distribución normal multivariante $(\mathbf{x}, y) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ para punto de datoss caracterizados por features y etiquetas numéricos. En este caso, bajo la pérdida de

error cuadrático, el Bayes estimator corresponde a la media posterior $\mu_{y|\mathbf{x}}$ de la etiqueta y, dado features $\mathbf{x}$ [22, 38]. El Bayes risk asociado es la varianza posterior $\sigma^2_{y|\mathbf{x}}$ (see Figure 19).

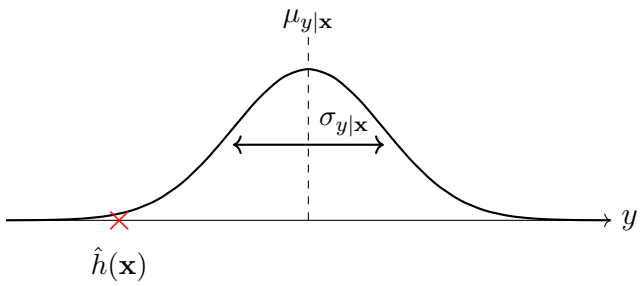


Figure 19: Si los features y la etiqueta de un punto de datos siguen una distribución normal multivariante, we podemos alcanzar el mínimo riesgo (bajo pérdida de error cuadrático) usando el Bayes estimator $\mu_{y|\mathbf{x}}$ para predecir el etiqueta y de un punto de datos con features $\mathbf{x}$. El mínimo riesgo es dada por la varianza posterior $\sigma^2_{y|\mathbf{x}}$. Podemos usar esta cantidad como baseline para evaluar la loss promedio del model $\hat{h}$ entrenado.

reglamento general de protección de datos (RGPD) El RGPD fue promulgado por la Union Europea (EU), y entró en efecto el 25 de Mayo de 2018 [32]. Protege la privacidad y los derechos sobre los datos de los individuos dentro de la EU. El RGPD tiene implicaciones significativas sobre cómo se recopilan, almacenan y utilizan los datos en aplicaciones de ML. Las disposiciones clave incluyen:

- Data minimization principle: los sistemas de ML deben utilizar únicamente la cantidad necesaria de datos personal para su propósito.

- Transparencia y explicabilidad: los sistemas de ML deben permitir a sus usuarios comprender cómo se toman las decisiones que los afectan.
- Derechos del titular de los datos Datos:los usuarios deben tener la posibilidad de acceder, rectificar y eliminar sus datos, así como oponerse a decisiones automatizadas y perfiles.
- Responsabilidad: las organizaciones deben garantizar una seguridad robusta de los datos y demostrar cumplimiento mediante documentación y auditorías periódicas.

regresión Los problemas de regresión se centran en la predicción de una etiqueta numérica basada únicamente en las features de un punto de datos [6, Ch. 2].

regularizador Un regularizador asigna a cada hipótesis h de un hypothesis space $\mathcal{H}$ una medida cuantitativa $\mathcal{R}\big\{h\big\}$ que indica cuánto podría diferir su error de predicción en un conjunto de entrenamiento de sus errores de predicción en punto de datoss fuera del conjunto de entrenamiento. Ridge regression utiliza el regularizador $\mathcal{R}\big\{h\big\} := \|\mathbf{w}\|_2^2$ para mapas de hipótesis lineales $h^{(\mathbf{w})}(\mathbf{x}) := \mathbf{w}^T\mathbf{x}$ [6, Ch. 3]. Lasso utiliza el regularizador $\mathcal{R}\big\{h\big\} := \|\mathbf{w}\|_1$ para mapas de hipótesis lineales $h^{(\mathbf{w})}(\mathbf{x}) := \mathbf{w}^T\mathbf{x}$ [6, Ch. 3].

regularization A key challenge of modern ML applications is that they often use large models, which have an dimensión efectiva in the order of billions. Training a high-dimensional model using basic ERM-based methods is prone to sobreajuste: the learned hipótesis performs well

on the conjunto de entrenamiento but poorly outside the conjunto de entrenamiento. Regularization refers to modifications of a given instance of ERM in order to avoid sobreajuste, i.e., to ensure that the learned hipótesis performs not much worse outside the conjunto de entrenamiento. There are three routes for implementing regularization:

1) Model pruning: We prune the original model $\mathcal{H}$ to obtain a smaller model $\mathcal{H}'$. For a parametric model, the pruning can be implemented via constraints on the model parameters (such as $w_1 \in [0.4, 0.6]$ for the weight of feature x_1 in linear regression).

2) Loss penalization: We modify the función objetivo of ERM by adding a penalty term to the error de entrenamiento. The penalty term estimates how much larger the expected loss (or riesgo) is compared to the average loss on the conjunto de entrenamiento.

3) Data augmentation: We can enlarge the conjunto de entrenamiento $\mathcal{D}$ by adding perturbed copies of the original punto de datoss in $\mathcal{D}$. One example for such a perturbation is to add the realización of an RV to the vector de características of a punto de datos.

Figure 20 illustrates the above three routes to regularization. These routes are closely related and sometimes fully equivalent: data augmentation using VA gaussianas to perturb the vector de característicass in the conjunto de entrenamiento of linear regression has the same effect as adding the penalty $\lambda \|\mathbf{w}\|_2^2$ to the error de entrenamiento (which is nothing but ridge regression). The decision on which route to use for regularization can be based on the available computational infras-

tructure. For example, it might be much easier to implement data augmentation than model pruning.

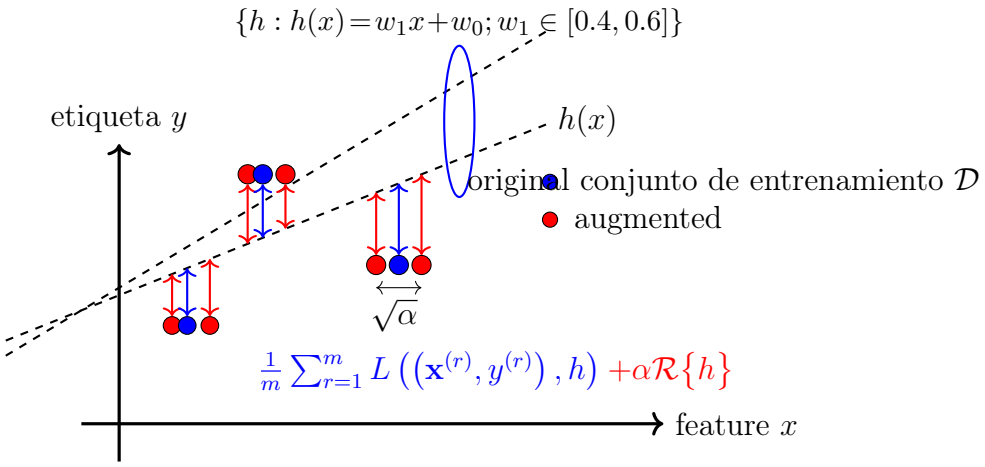


Figure 20: Three approaches to regularization: 1) data augmentation; 2) loss penalization; and 3) model pruning (via constraints on model parameters).

regularized empirical risk minimization (RERM) Basic ERM learns a hipótesis (or trains a model) $h \in \mathcal{H}$ based solely on the riesgo empírico $\widehat{L}\big(h|\mathcal{D}\big)$ incurred on a conjunto de entrenamiento $\mathcal{D}$. To make ERM less prone to sobreajuste, we can implement regularization by including a (scaled) regularizador $\mathcal{R}\big\{h\big\}$ in the learning objective. This leads to regularized empirical risk minimization (RERM),

$$\hat{h} \in \operatorname*{argmin}_{h \in \mathcal{H}} \widehat{L}\big(h|\mathcal{D}\big) + \alpha \mathcal{R}\big\{h\big\}. \tag{11}$$

The parameter $\alpha \geq 0$ controls the regularization strength. For $\alpha = 0$, we recover standard ERM without regularization. As α increases, the learned hipótesis is increasingly biased toward small values of $\mathcal{R}\big\{h\big\}$. The component $\alpha\mathcal{R}\big\{h\big\}$ in the función objetivo of (11) can be intuitively understood as a surrogate for the increased average loss that may occur when predicting etiquetas for punto de datoss outside the conjunto de entrenamiento. This intuition can be made precise in various ways. For example, consider a modelo lineal trained using pérdida de error cuadrático and the regularizador $\mathcal{R}\big\{h\big\} = \|\mathbf{w}\|_2^2$. In this setting, $\alpha\mathcal{R}\big\{h\big\}$ corresponds to the expected increase in loss caused by adding VA gaussianas to the vector de característicass in the conjunto de entrenamiento [6, Ch. 3]. A principled construction for the regularizador $\mathcal{R}\big\{h\big\}$ arises from approximate upper bounds on the generalization error. The resulting RERM instance is known as SRM [96, Sec. 7.2].

regularized loss minimization (RLM) See RERM.

Rényi divergence The Rényi divergence measures the (dis)similarity between two probability distributions [97].

reward A reward refers to some observed (or measured) quantity that allows us to estimate the loss incurred by the predicción (or decision) of a hipótesis $h(\mathbf{x})$. For example, in an ML application to self-driving vehicles, $h(\mathbf{x})$ could represent the current steering direction of a vehicle. We could construct a reward from the measurements of a collision sensor that indicate if the vehicle is moving towards an obstacle. We define a low reward for the steering direction $h(\mathbf{x})$ if the vehicle moves dangerously towards an obstacle.

ridge regression Ridge regresión learns the weights $\mathbf{w}$ of a linear hipótesis map $h^{(\mathbf{w})}(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$. The quality of a particular choice for the model parameters $\mathbf{w}$ is measured by the sum of two components. The first component is the average pérdida de error cuadrático incurred by $h^{(\mathbf{w})}$ on a set of punto de dato etiquetados (i.e., the conjunto de entrenamiento). The second component is the scaled squared Euclidean norma $\alpha\|\mathbf{w}\|_2^2$ with a regularization parameter $\alpha > 0$. Adding $\alpha\|\mathbf{w}\|_2^2$ to the average pérdida de error cuadrático is equivalent to replacing each original punto de datoss by the realización of (infinitely many) i.i.d. RVs centered around these punto de datoss (see regularization).

riesgo Consideremos una hipótesis h que se utiliza para predecir la etiqueta y de un punto de datos a partir de sus features $\mathbf{x}$. Evaluamos la calidad de una predicción específica usando una loss function $L\left((\mathbf{x}, y), h\right)$. Si interpretamos los punto de datoss como realizaciónes de RVs i.i.d., entonces $L\left((\mathbf{x}, y), h\right)$ también se convierte en unarealización de una RV. El i.i.d. assumption nos permite definir el riesgo de una hipótesis como

la expectation de la loss $\mathbb{E}\big\{ L\left((\mathbf{x}, y), h\right) \big\}$. El riesgo de h depende tanto de la loss function elegida como de la probability distribution de los punto de datoss.

riesgo empírico El riesgo empírico $\widehat{L}\big(h|\mathcal{D}\big)$ de una hipótesis sobre un conjunto de datos $\mathcal{D}$ es la loss promedio incurrida por h al aplicarse a los punto de datoss en el $\mathcal{D}$.

sample A finite sequence (or list) of punto de datoss $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}$ that is obtained or interpreted as the realización of m i.i.d. RVs with a common probability distribution $p(\mathbf{z})$. The length m of the sequence is referred to as the sample size.

sample covariance matrix The sample covariance matrix $\widehat{\mathbf{\Sigma}} \in \mathbb{R}^{d \times d}$ for a given set of vector de características $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)} \in \mathbb{R}^d$ is defined as

$$\widehat{\mathbf{\Sigma}} = (1/m) \sum_{r=1}^{m} (\mathbf{x}^{(r)} - \widehat{\mathbf{m}})(\mathbf{x}^{(r)} - \widehat{\mathbf{m}})^T.$$

Here, we use the sample mean $\widehat{\mathbf{m}}$.

sample mean The sample media $\mathbf{m} \in \mathbb{R}^d$ for a given conjunto de datos, with vector de características $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)} \in \mathbb{R}^d$, is defined as

$$\mathbf{m} = (1/m) \sum_{r=1}^{m} \mathbf{x}^{(r)}.$$

sample size The number of individual punto de datoss contained in a conjunto de datos.

scatterplot A visualization technique that depicts punto de datoss by markers in a two-dimensional plane. Figure 21 depicts an example of a scatterplot.

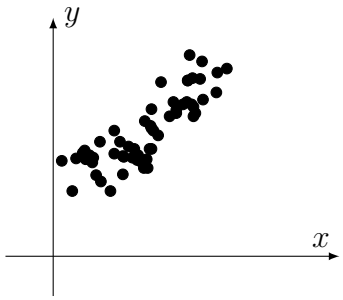


Figure 21: A scatterplot of some punto de datoss representing daily weather conditions in Finland. Each punto de datos is characterized by its mínimo daytime temperature x as the feature and its máximo daytime temperature y as the etiqueta. The temperatures have been measured at the FMI weather station Helsinki Kaisaniemi during 1.9.2024 - 28.10.2024.

sesgo Consideremos un método de ML que utiliza un hypothesis space $\mathcal{H}$ parametrizado. Este aprende los model parameters $\mathbf{w} \in \mathbb{R}^d$ utilizando el conjunto de datos

$$\mathcal{D} = \left\{ \left(\mathbf{x}^{(r)}, y^{(r)}\right) \right\}_{r=1}^{m}.$$

Para analizar las propiedades del método de ML, típicamente interpretamos los punto de datoss como realizaciónes de i.i.d. RVs,

$$y^{(r)} = h^{(\overline{\mathbf{w}})}\left(\mathbf{x}^{(r)}\right) + \boldsymbol{\varepsilon}^{(r)}, r = 1, \ldots, m.$$

Entonces podemos interpetar el método de ML como un estimador $\widehat{\mathbf{w}}$ calculado a partir de $\mathcal{D}$ (por ejemplo, resolviendo ERM). El sesgo (cuadrado) del estimador $\widehat{\mathbf{w}}$ se define como $B^2 := \left\|\mathbb{E}\{\widehat{\mathbf{w}}\} - \overline{\mathbf{w}}\right\|_2^2$.

similarity graph Some ML applications generate punto de datoss that are related by a domain-specific notion of similarity. These similarities can be represented conveniently using a similarity graph $\mathcal{G} = \big(\mathcal{V} := \{1, \ldots, m\}, \mathcal{E}\big)$. The node $r \in \mathcal{V}$ represents the r-th punto de datos. Two nodes are connected by an undirected edge if the corresponding punto de datoss are similar.

singular value decomposition (SVD) The SVD for a matrix $\mathbf{A} \in \mathbb{R}^{m \times d}$ is a factorization of the form

$$\mathbf{A} = \mathbf{V} \mathbf{\Lambda} \mathbf{U}^T,$$

with orthonormal matrices $\mathbf{V} \in \mathbb{R}^{m \times m}$ and $\mathbf{U} \in \mathbb{R}^{d \times d}$ [3]. The matrix $\mathbf{\Lambda} \in \mathbb{R}^{m \times d}$ is only non-zero along the main diagonal, whose entries $\Lambda_{j,j}$ are non-negative and referred to as singular values.

smooth A real-valued function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is smooth if it is diferenciable and its gradiente $\nabla f(\mathbf{w})$ is continuous at all $\mathbf{w} \in \mathbb{R}^d$ [65, 98]. A smooth function f is referred to as β-smooth if the gradiente $\nabla f(\mathbf{w})$ is Lipschitz continuous with Lipschitz constant β, i.e.,

$$\|\nabla f(\mathbf{w}) - \nabla f(\mathbf{w}')\| \leq \beta \|\mathbf{w} - \mathbf{w}'\|, \text{ for any } \mathbf{w}, \mathbf{w}' \in \mathbb{R}^d.$$

The constant β quantifies the amount of smoothness of the function f: the smaller the β, the smoother f is. Optimization problems with a smooth función objetivo can be solved effectively by gradient-based methods. Indeed, gradient-based methods approximate the función objetivo locally around a current choice $\mathbf{w}$ using its gradiente. This approximation works well if the gradiente does not change too rapidly.

We can make this informal claim precise by studying the effect of a single paso de gradiente with step size $\eta = 1/\beta$ (see Figure 22).

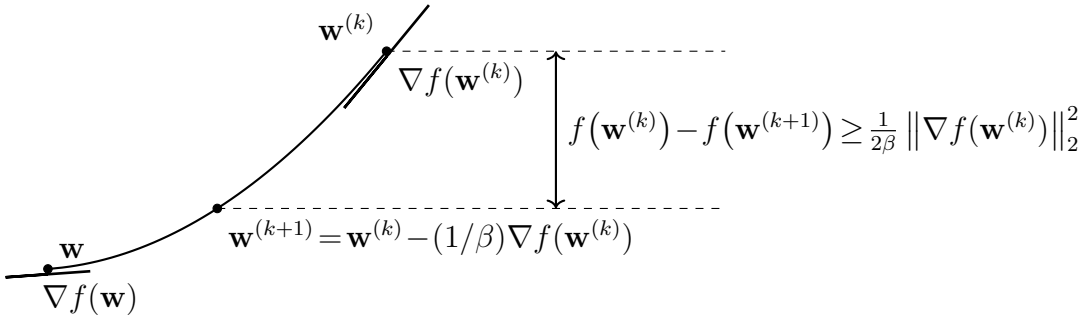


Figure 22: Consider an función objetivo $f(\mathbf{w})$ that is β-smooth. Taking a paso de gradiente, with step size $\eta = 1/\beta$, decreases the objective by at least $\frac{1}{2\beta} \left\|\nabla f(\mathbf{w}^{(k)})\right\|_2^2$ [65, 66, 98]. Note that the step size $\eta = 1/\beta$ becomes larger for smaller β. Thus, for smoother función objetivos (i.e., those with smaller β), we can take larger steps.

sobreajuste Consideremos un método de ML que utiliza ERM para aprender una hipótesis con el mínimo riesgo empírico en un conjunto de entrenamiento dado. Dicho método presenta sobreajuste del conjunto de entrenamiento si aprende una hipótesis con un riesgo empírico pequeño sobre el conjunto de entrenamiento pero una loss significativamente mayor fuera de él conjunto de entrenamiento.

soft clustering Soft clustering refers to the task of partitioning a given set of punto de datoss into (a few) overlapping clusters. Each punto de datos is assigned to several different clusters with varying degrees of belonging. Soft clustering methods determine the grado de pertenencia

(or soft cluster assignment) for each punto de datos and each cluster. A principled approach to soft clustering is by interpreting punto de datoss as i.i.d. realizacións of a GMM. We then obtain a natural choice for the grado de pertenencia as the conditional probabilidad of a punto de datos belonging to a specific mixture component.

stability Stability is a desirable property of a ML method $\mathcal{A}$ that maps a conjunto de datos $\mathcal{D}$ (e.g., a conjunto de entrenamiento) to an output $\mathcal{A}(\mathcal{D})$, such as learned model parameters or the predicción for a specific punto de datos. Intuitively, $\mathcal{A}$ is stable if small changes in the input conjunto de datos $\mathcal{D}$ lead to small changes in the output $\mathcal{A}(\mathcal{D})$. Several formal notions of stability exist that enable bounds on the generalization error or riesgo of the method; see [37, Ch. 13]. To build intuition, consider the three datasets depicted in Fig. 23, each of which is equally likely under the same datos-generating probability distribution. Since the optimal model parameters are determined by this underlying probability distribution, an accurate ML method $\mathcal{A}$ should return the same (or very similar) output $\mathcal{A}(\mathcal{D})$ for all three conjunto de datos. In other words, any useful $\mathcal{A}$ must be robust to variability in sample realizacións from the same probability distribution, i.e., it must be stable.

step size See learning rate.

stochastic gradient descent (SGD) Stochastic GD is obtained from GD by replacing the gradiente of the función objetivo with a stochastic approximation. A main application of stochastic GD is to train a parametrized model via ERM on a conjunto de entrenamiento $\mathcal{D}$ that is

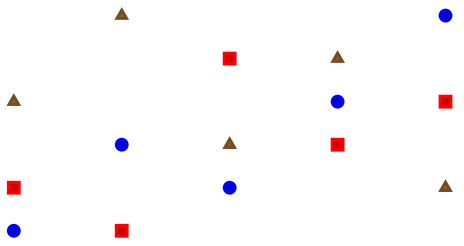


Figure 23: Three conjunto de datoss $\mathcal{D}^{(*)}$, $\mathcal{D}^{(\square)}$, and $\mathcal{D}^{(\triangle)}$, each sampled independently from the same datos-generating probability distribution. A stable ML method should return similar outputs when trained on any of these conjunto de datoss.

either very large or not readily available (e.g., when punto de datoss are stored in a database distributed all over the planet). To evaluate the gradiente of the riesgo empírico (as a function of the model parameters $\mathbf{w}$), we need to compute a sum $\sum_{r=1}^{m} \nabla_{\mathbf{w}} L\left(\mathbf{z}^{(r)}, \mathbf{w}\right)$ over all punto de datoss in the conjunto de entrenamiento. We obtain a stochastic approximation to the gradiente by replacing the sum $\sum_{r=1}^{m} \nabla_{\mathbf{w}} L\left(\mathbf{z}^{(r)}, \mathbf{w}\right)$ with a sum $\sum_{r \in \mathcal{B}} \nabla_{\mathbf{w}} L\left(\mathbf{z}^{(r)}, \mathbf{w}\right)$ over a randomly chosen subset $\mathcal{B} \subseteq \{1, \ldots, m\}$ (see Figure 24). We often refer to these randomly chosen punto de datoss as a lote. The lote size $|\mathcal{B}|$ is an important parameter of stochastic GD. Stochastic GD with $|\mathcal{B}| > 1$ is referred to as mini-lote stochastic GD [99].

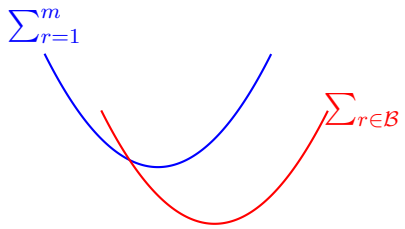


Figure 24: Stochastic GD for ERM approximates the gradiente $\sum_{r=1}^{m} \nabla_{\mathbf{w}} L\left(\mathbf{z}^{(r)}, \mathbf{w}\right)$ by replacing the sum over all punto de datoss in the conjunto de entrenamiento (indexed by $r = 1, \ldots, m$) with a sum over a randomly chosen subset $\mathcal{B} \subseteq \{1, \ldots, m\}$.

stopping criterion Many ML methods use iterative algorithms that construct a sequence of model parameters (such as the weights of a linear map or the weights of an ANN). These parameters (hopefully) converge to an optimal choice for the model parameters. In practice, given finite computational resources, we need to stop iterating after a finite number of repetitions. A stopping criterion is any well-defined condition required for stopping the iteration.

structural risk minimization (SRM) Structural risk minimization (SRM) is an instance of RERM, which which the model $\mathcal{H}$ can be expressed as a countable union of sub-models: $\mathcal{H} = \bigcup_{n=1}^{\infty} \mathcal{H}^{(n)}$. Each sub-model $\mathcal{H}^{(n)}$ permits the derivation of an approximate upper bound on the generalization error incurred when applying ERM to train $\mathcal{H}^{(n)}$. These individual bounds—one for each sub-model—are then combined to form a regularizador used in the RERM objective. These approximate upper bounds (one for each $\mathcal{H}^{(n)}$) are then combined to construct a

regularizador for RERM [37, Sec. 7.2].

subajuste Consideremos un método de ML que utiliza ERM para aprender una hipótesis con el mínimo riesgo empírico en un conjunto de entrenamiento dado. Dicho método presenta subajuste del conjunto de entrenamiento si no es capaz de aprender una hipótesis con un riesgo empírico suficientemente pequeño sobre el conjunto de entrenamiento. Si un método sufre de subajuste, típicamente tampoco podrá aprender una hipótesis con un riesgo pequeño.

subgradient descent Subgradiente descent is a generalization of GD that does not require differentiability of the function to be minimized. This generalization is obtained by replacing the concept of a gradiente with that of a subgradiente. Similar to gradientes, also subgradientes allow us to construct local approximations of an función objetivo. The función objetivo might be the riesgo empírico $\widehat{L}\big(h^{(\mathbf{w})}\big|\mathcal{D}\big)$ viewed as a function of the model parameters $\mathbf{w}$ that select a hipótesis $h^{(\mathbf{w})} \in \mathcal{H}$.

subgradiente Para una función de valor real $f : \mathbb{R}^d \to \mathbb{R} : \mathbf{w} \mapsto f(\mathbf{w})$, un vector $\mathbf{a}$ tal que $f(\mathbf{w}) \geq f(\mathbf{w}') + \big(\mathbf{w} - \mathbf{w}'\big)^T \mathbf{a}$ se se denomina subgradiente de f en $\mathbf{w}'$ [100, 101].

support vector machine (SVM) The SVM is a binary clasificación method that learns a linear hipótesis map. Thus, like linear regression and logistic regression, it is also an instance of ERM for the modelo lineal. However, the SVM uses a different loss function from the one used in those methods. As illustrated in Figure 25, it aims to maximally separate punto de datoss from the two different classes in the espacio

de características (i.e., máximo margin principle). Maximizing this separation is equivalent to minimizing a regularized variant of the hinge loss (3) [52, 71, 102].

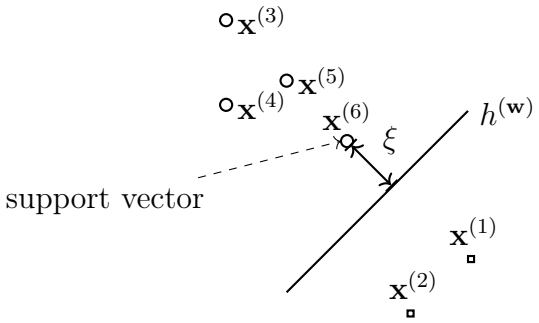


Figure 25: The SVM learns a hipótesis (or clasificador) $h^{(\mathbf{w})}$ with minimal average soft-margin hinge loss. Minimizing this loss is equivalent to maximizing the margin ξ between the decision boundary of $h^{(\mathbf{w})}$ and each class of the conjunto de entrenamiento.

The above basic variant of SVM is only useful if the punto de datoss from different categories can be (approximately) linearly separated. For an ML application where the categories are not derived from a kernel.

supremo (o mínimo de las cotas superiores) El supremo de un conjunto de números reales es el número más pequeño que es mayor o igual que todos los elementos del conjunto. Formalmente, un número real a es el supremo de un conjunto $\mathcal{A} \subseteq \mathbb{R}$ si: 1) a es una cota superior de $\mathcal{A}$; y 2) ningún número menor que a es una cota superior de $\mathcal{A}$. Todo conjunto no vacío de números reales que esté acotado superiormente tiene un supremo,aun si no contiene su supremo como un elemento [2, Sec. 1.4].

tarea de aprendizaje Consideremos un conjunto de datos $\mathcal{D}$ constituido por varios punto de datoss, cada uno caracterizado por features $\mathbf{x}$. Por ejemplo, el conjunto de datos $\mathcal{D}$ podría estar constituido por imágenes de una base de datos particular. A veces puede ser útil representar un conjunto de datos $\mathcal{D}$, junto con la elección de features, por un probability distribution $p(\mathbf{x})$. Una tarea de aprendizaje asociada a $\mathcal{D}$ consiste en una elección específica de la etiqueta de un punto de datos y el correspondiente espacio de etiquetas. Dada una elección de la loss function y el model, una tarea de aprendizaje da lugar a una instancia de ERM. Así, también podríamos definir una tarea de aprendizaje mediante una instancia de ERM, es decir, mediante una función objetivo. Nótese que, para el mismo conjunto de datos, obtenemos diferentes tareas de aprendizaje utilizando distintas elecciones de features y etiqueta de un punto de datos. Estas tareas de aprendizaje están relacionadas, ya que se basan en el mismo conjunto de datos, y resolverlas conjuntamente (usando métodos de aprendizaje multitarea) es preferible a resolverlas de forma independiente [103], [104], [105].

total variation See GTV.

transparencia La transparencia es un requisito fundamental para una trustworthy AI confiable [106]. En ML, suele utilizarse como sinónimo de explicabilidad [57, 107] pero en el contexto más amplio de sistemas de AI, incluye también información sobre limitaciones, confiabilidad y uso previsto. En sistemas de diagnóstico médico, se requiere informar el nivel de confianza de una predicción. En aplicaciones financieras

como la puntuación crediticia, las decisiones automatizadas basadas en AI deben ir acompañadas de explicaciones sobre los factores que influyeron en ellas, como el nivel de ingresos o el historial crediticio. These explanations Estas explicaciones permiten que las personas (por ejemplo, un solicitante de crédito) comprendan y, si es necesario, impugnen decisiones automatizadas.. Algunos métodos de ML ofrecen transparencia de manera intrínseca. Por ejemplo, la logistic regression permite interpretar la fiabilidad de una clasificación mediante el valor absoluto $|h(\mathbf{x})|$. Las decision trees, también son consideradas transparentes porque generan reglas comprensibles para los humanos. [55]. La transparencia también requiere que se informe explícitamente cuando una persona está interactuando con un sistema AI. Por ejemplo, los chatbots impulsados por AI deben indicar claramente que no son humanos. For example, AI-powered chatbots should notify users that they are interacting with an automated system rather than a human. Furthermore, transparency encompasses comprehensive documentation detailing the purpose and design choices underlying the AI system. Además, la transparencia incluye documentación exhaustiva que detalle el propósito, las decisiones de diseño y los casos de uso previstos del sistema. Ejemplos de esto son las hojas de datos de models [28] y las tarjetas de sistemas de AI [108], que ayudan a los desarrolladores y usuarios a entender las limitaciones y aplicaciones adecuadas de un sistema AI [109].

uncertainty Uncertainty refers to the degree of confidence—or lack thereof—associated

with a quantity such as a model prediction, parameter estimate, or observed data point. In ML, uncertainty arises from various sources, including noisy data, limited training samples, or ambiguity in model assumptions. Probability theory offers a principled framework for representing and quantifying such uncertainty.

unidad lineal rectificada (ReLU) La unidad lineal rectificada (ReLU) es una elección popular para la función de activación de una neurona dentro de una ANN. Se define como $\sigma(z) = \max\{0, z\}$, donde z es la entrada ponderada de la neurona artificial.

upper confidence bound (UCB) Consider a ML application that requires selecting, at each time step k, an action a_k from a finite set of alternatives $\mathcal{A}$. The utility of selecting action a_k is quantified by a numeric reward signal $r^{(a_k)}$. A widely used modelo probabilístico for this type of sequential decision-making problem is the stochastic multi-armed bandit setting [80]. In this model, the reward $r^{(a)}$ is viewed as the realización of a RV with unknown media $\mu^{(a)}$. Ideally, we would always choose the action with the largest expected reward $\mu^{(a)}$, but these means are unknown and must be estimated from observed datos. Simply choosing the action with the largest estimate $\widehat{\mu}^{(a)}$ can lead to suboptimal outcomes due to estimation uncertainty. The UCB strategy addresses this by selecting actions not only based on their estimated means but also by incorporating a term that reflects the uncertainty in these estimates—favouring actions with high potential reward and high uncertainty. Theoretical guarantees for the performance of UCB strategies,including logarithmic

regret bounds, are established in [80].

validación La validación se refiere a la práctica de evaluar el loss incurrido por una hipótesis $\hat{h}$ que ha sido aprendida mediante algún método de ML, por ejemplo, resolviendo ERM en un conjunto de entrenamiento $\mathcal{D}$. La validación implica evaluar el desempeño de la hipótesis en un conjunto de punto de datoss que no están contenidos en el conjunto de entrenamiento $\mathcal{D}$.

variable aleatoria (RV) Una RV es una función que mapea desde un probability space $\mathcal{P}$ a un espacio de valores [38, 51]. El probability space consiste en eventos elementales y está equipado con una medida de probabilidad que asigna probabilidades a subconjuntos de $\mathcal{P}$. Existen diferentes tipos de variables aleatorias (RV), que incluyen:

- RVs binarias, que asignan eventos elementales a un conjunto de dos valores distintos, como $\{-1, 1\}$ o $\{\text{cat}, \text{no cat}\}$;
- RVs de valor real, que toman valores en los números reales $\mathbb{R}$;
- RVs de valor vectorial, que mapean eventos elementales al Euclidean space $\mathbb{R}^d$.

La teoría de Probabilidad utiliza el concepto de espacios medibles para definir rigurosamente y estudiar las propiedades de (grandes) colecciones de RVs [51].

variable aleatoria gaussiana (VA gaussiana) Una RV gaussiana están-

dar es una RV real x con pdf [5, 38, 77]

$$p(x) = \frac{1}{\sqrt{2\pi}} \exp^{-x^2/2}.$$

Dada una RV gaussiana estándar x, podemos construir una RV gaussiana general x' con media μ y varianza σ^2 mediante $x' := \sigma(x + \mu)$. La probability distribution de una RV gaussiana se conoce como distribución normal, denotada $\mathcal{N}(\mu, \sigma)$.

Un vector aleatorio gaussiano $\mathbf{x} \in \mathbb{R}^d$ con covariance matrix $\mathbf{C}$ y media $\boldsymbol{\mu}$ puede construirse como $\mathbf{x} := \mathbf{A}\big(\mathbf{z} + \boldsymbol{\mu}\big)$. Aquí, $\mathbf{A}$ es cualquier matriz que satisface $\mathbf{A}\mathbf{A}^T = \mathbf{C}$ y $\mathbf{z} := \big(z_1, \ldots, z_d\big)^T$ es un vector cuyos elementos son i.i.d. gaussianas estándar RVs $z_1, \ldots, z_d$. Los procesos aleatorios gaussianos generalizan los vectores aleatorios gaussianos aplicando transformaciones lineales a a secuencias infinitas de RVs gaussianas estándar [110].

Las RVs gaussianas se utilizan ampliamente como modelo probabilísticos en el análisis estadístico de métodos de ML. Su importancia se debe, en parte, al teorema del límite central, que establece que el promedio de un número creciente de RVs independientes (aunque no sean gaussianas) converge a una RV gaussiana [90].

varianza La varianza de una RV real x se define como la expectation $\mathbb{E}\big\{\big(x - \mathbb{E}\{x\}\big)^2\big\}$ de la diferencia cuadrada entre x y su expectation $\mathbb{E}\{x\}$. Extendemos esta definición a RVs vectoriales$\mathbf{x}$ como $\mathbb{E}\big\{\big\|\mathbf{x} - \mathbb{E}\{\mathbf{x}\}\big\|_2^2\big\}$.

vecino más cercano (NN) Los métodos de vecino más cerano (NN) aprenden una hipótesis $h : \mathcal{X} \to \mathcal{Y}$ cuyo valor $h(\mathbf{x})$ se determina únicamente

por los vecinos más cercanos dentro de un conjunto de datos. Distintos métodos usan diferentes medidas para determinar los vecinos más cercanos. Si los punto de datoss se caracterizan por vector de característicass numéricos, podemos usar la distancia euclidiana como medida. the metric.

vecinos Los vecinos de un nodo $i \in \mathcal{V}$ dentro de un red de aprendizaje federado (red FL) son los nodos $i' \in \mathcal{V} \setminus \{i\}$ conectados con i por una arista.

vector de características El vector de características se refiere a un vector $\mathbf{x} = \big(x_1, \ldots, x_d\big)^T$ cuyos elementos son características individuales $x_1, \ldots, x_d$. Muchos métodos de ML utilizan vectores de características que pertenecen a algún Euclidean space de dimensión finita $\mathbb{R}^d$. Sin embargo, para algunos métodos de ML, puede ser más conveniente trabajar con vectores de características que pertenezcan a un espacio vectorial de dimensión infinita (por ejemplo, ver kernel method).

weights Consider a parametrized hypothesis space $\mathcal{H}$. We use the term weights for numeric model parameters that are used to scale features or their transformations in order to compute $h^{(\mathbf{w})} \in \mathcal{H}$. A modelo lineal uses weights $\mathbf{w} = \big(w_1, \ldots, w_d\big)^T$ to compute the linear combination $h^{(\mathbf{w})}(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$. Weights are also used in ANNs to form linear combinations of features or the outputs of neurons in hidden layers.

zero-gradient condition Consider the unconstrained optimization problem $\min_{\mathbf{w} \in \mathbb{R}^d} f(\mathbf{w})$ with a smooth and convex función objetivo $f(\mathbf{w})$. A

necessary and sufficient condition for a vector $\widehat{\mathbf{w}} \in \mathbb{R}^d$ to solve this problem is that the gradiente $\nabla f\left(\widehat{\mathbf{w}}\right)$ is the zero vector,

$$\nabla f\left(\widehat{\mathbf{w}}\right) = \mathbf{0} \Leftrightarrow f\left(\widehat{\mathbf{w}}\right) = \min_{\mathbf{w} \in \mathbb{R}^d} f(\mathbf{w}).$$

0/1 loss La loss 0/1 $L^{(0/1)}\left((\mathbf{x}, y), h\right)$ mide la calidad de un clasificador $h(\mathbf{x})$ que genera una predicción $\hat{y}$ (por ejemplo, mediante un umbral como en (1)) para la etiqueta y de un punto de datos con features $\mathbf{x}$. Es igual a 0 si la predicción es correcta, es decir, $L^{(0/1)}\left((\mathbf{x}, y), h\right) = 0$ cuando $\hat{y} = y$. Es igual a 1 si la predicción es incorrecta, es decir, $L^{(0/1)}\left((\mathbf{x}, y), h\right) = 1$ cuando $\hat{y} \neq y$.

Index

References

[1] W. Rudin, *Real and Complex Analysis*, 3rd ed. New York, NY, USA: McGraw-Hill, 1987.

[2] W. Rudin, *Principles of Mathematical Analysis*, 3rd ed. New York, NY, USA: McGraw-Hill, 1976.

[3] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 4th ed. Baltimore, MD, USA: The Johns Hopkins Univ. Press, 2013.

[4] G. H. Golub and C. F. Van Loan, "An analysis of the total least squares problem," *SIAM J. Numer. Anal.*, vol. 17, no. 6, pp. 883–893, Dec. 1980, doi: 10.1137/0717073.

[5] D. P. Bertsekas and J. N. Tsitsiklis, *Introduction to Probability*, 2nd ed. Belmont, MA, USA: Athena Scientific, 2008.

[6] A. Jung, *Machine Learning: The Basics*. Singapore, Singapore: Springer Nature, 2022.

[7] Y. SarcheshmehPour, Y. Tian, L. Zhang, and A. Jung, "Flow-based clustering and spectral clustering: A comparison," in *2021 55th Asilomar Conf. Signals, Syst., Comput.*, M. B. Matthews, Ed. pp. 1292–1296, doi: 10.1109/IEEECONF53345.2021.9723162.

[8] U. von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, Dec. 2007, doi: 10.1007/s11222-007-9033-z.

[9] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. Cambridge, MA, USA: MIT Press, 2022. [Online].

Available: http://ebookcentral.proquest.com/lib/aalto-ebooks/detail.action?docID=6925615

[10] M. Sipser, *Introduction to the Theory of Computation*, 3rd ed. Andover, U.K.: Cengage Learning, 2013.

[11] R. Motwani and P. Raghavan, *Randomized Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 1995.

[12] R. G. Gallager, *Stochastic Processes: Theory for Applications*. New York, NY, USA: Cambridge Univ. Press, 2013.

[13] G. Tel, *Introduction to Distributed Algorithms*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2000.

[14] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Belmont, MA, USA: Athena Scientific, 2015.

[15] E. Hazan, "Introduction to online convex optimization," *Found. Trends Optim.*, vol. 2, no. 3–4, pp. 157–325, Aug. 2016, doi: 10.1561/2400000013.

[16] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games*. New York, NY, USA: Cambridge Univ. Press, 2006.

[17] L. Richardson and M. Amundsen, *RESTful Web APIs*. Sebastopol, CA, USA: O'Reilly Media, 2013.

[18] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical Learning with Sparsity: The Lasso and Generalizations*. Boca Raton, FL, USA: CRC Press, 2015.

[19] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu, "Horizontal federated learning," in *Federated Learning*. Cham, Switzerland: Springer Nature, 2020, ch. 4, pp. 49–67.

[20] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu, "Vertical federated learning," in *Federated Learning*. Cham, Switzerland: Springer Nature, 2020, ch. 5, pp. 69–81.

[21] O. Chapelle, B. Schölkopf, and A. Zien, Eds. *Semi-Supervised Learning*. Cambridge, MA, USA: MIT Press, 2006.

[22] E. L. Lehmann and G. Casella, *Theory of Point Estimation*, 2nd ed. New York, NY, USA: Springer-Verlag, 1998.

[23] A. Silberschatz, H. F. Korth, and S. Sudarshan, *Database System Concepts*, 7th ed. New York, NY, USA: McGraw-Hill Education, 2019. [Online]. Available: https://db-book.com/

[24] S. Abiteboul, R. Hull, and V. Vianu, *Foundations of Databases*. Reading, MA, USA: Addison-Wesley Publishing Company, 1995.

[25] S. Hoberman, *Data Modeling Made Simple: A Practical Guide for Business and IT Professionals*, 2nd ed. Basking Ridge, NJ, USA: Technics Publications, 2009.

[26] R. Ramakrishnan and J. Gehrke, *Database Management Systems*, 3rd ed. New York, NY, USA: McGraw-Hill, 2002.

[27] E. F. Codd, "A relational model of data for large shared data

banks," *Commun. ACM*, vol. 13, no. 6, pp. 377–387, Jun. 1970, doi: 10.1145/362384.362685.

[28] T. Gebru et al., "Datasheets for datasets," *Commun. ACM*, vol. 64, no. 12, pp. 86–92, Nov. 2021, doi: 10.1145/3458723.

[29] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[30] D. Sun, K.-C. Toh, and Y. Yuan, "Convex clustering: Model, theoretical guarantee and efficient algorithm," *J. Mach. Learn. Res.*, vol. 22, no. 9, pp. 1–32, Jan. 2021. [Online]. Available: http://jmlr.org/papers/v22/18-694.html

[31] K. Pelckmans, J. De Brabanter, J. A. K. Suykens, and B. De Moor, "Convex clustering shrinkage," presented at the PASCAL Workshop Statist. Optim. Clustering Workshop, 2005.

[32] European Union, "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation) (Text with EEA relevance)," L 119/1, May 4, 2016. [Online]. Available: https://eur-lex.europa.eu/eli/reg/2016/679/oj

[33] European Union, "Regulation (EU) 2018/1725 of the European Parliament and of the Council of 23 October 2018 on the protection of natural persons with regard to the processing of personal data by the Union institutions, bodies, offices and agencies and on the free movement

of such data, and repealing Regulation (EC) No 45/2001 and Decision No 1247/2002/EC (Text with EEA relevance)," L 295/39, Nov. 21, 2018. [Online]. Available: https://eur-lex.europa.eu/eli/reg/2018/1725/oj

[34] X. Liu, H. Li, G. Xu, Z. Chen, X. Huang, and R. Lu, "Privacy-enhanced federated learning against poisoning adversaries," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 4574–4588, 2021, doi: 10.1109/TIFS.2021.3108434.

[35] J. Zhang, B. Chen, X. Cheng, H. T. T. Binh, and S. Yu, "PoisonGAN: Generative poisoning attacks against federated learning in edge computing systems," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3310–3322, Mar. 2021, doi: 10.1109/JIOT.2020.3023126.

[36] K. Abayomi, A. Gelman, and M. Levy, "Diagnostics for multivariate imputations," *J. Roy. Statist. Soc.: Ser. C (Appl. Statist.)*, vol. 57, no. 3, pp. 273–291, Jun. 2008, doi: 10.1111/j.1467-9876.2007.00613.x.

[37] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. New York, NY, USA: Cambridge Univ. Press, 2014.

[38] R. M. Gray, *Probability, Random Processes, and Ergodic Properties*, 2nd ed. New York, NY, USA: Springer Science+Business Media, 2009.

[39] A. Lapidoth, *A Foundation in Digital Communication*. Cambridge, U.K.: Cambridge Univ. Press, 2009.

[40] A. Makhdoumi, S. Salamatian, N. Fawaz, and M. Médard, "From the

information bottleneck to the privacy funnel," in *2014 IEEE Inf. Theory Workshop*, pp. 501–505, doi: 10.1109/ITW.2014.6970882.

[41] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1993.

[42] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY, USA: Springer Science+Business Media, 2001.

[43] L. Cohen, *Time-Frequency Analysis*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1995.

[44] J. Li, L. Han, X. Li, J. Zhu, B. Yuan, and Z. Gou, "An evaluation of deep neural network models for music classification using spectrograms," *Multimedia Tools Appl.*, vol. 81, no. 4, pp. 4621–4647, Feb. 2022, doi: 10.1007/s11042-020-10465-9.

[45] B. Boashash, Ed. *Time Frequency Signal Analysis and Processing: A Comprehensive Reference*. Oxford, U.K.: Elsevier, 2003.

[46] S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way*, 3rd ed. Burlington, MA, USA: Academic, 2009.

[47] D. N. Gujarati and D. C. Porter, *Basic Econometrics*, 5th ed. New York, NY, USA: McGraw-Hill/Irwin, 2009.

[48] Y. Dodge, Ed. *The Oxford Dictionary of Statistical Terms*. New York, NY, USA: Oxford Univ. Press, 2003.

[49] B. S. Everitt, *The Cambridge Dictionary of Statistics*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2002.

[50] P. R. Halmos, *Measure Theory*. New York, NY, USA: Springer-Verlag, 1974.

[51] P. Billingsley, *Probability and Measure*, 3rd ed. New York, NY, USA: Wiley, 1995.

[52] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer Science+Business Media, 2006.

[53] M. J. Wainwright and M. I. Jordan, "Graphical models, exponential families, and variational inference," *Found. Trends Mach. Learn.*, vol. 1, no. 1–2, pp. 1–305, Nov. 2008, doi: 10.1561/2200000001.

[54] L. Zhang, G. Karakasidis, A. Odnoblyudova, L. Dogruel, Y. Tian, and A. Jung, "Explainable empirical risk minimization," *Neural Comput. Appl.*, vol. 36, no. 8, pp. 3983–3996, Mar. 2024, doi: 10.1007/s00521-023-09269-3.

[55] C. Rudin, "Stop explaining black box machine learning models for high-stakes decisions and use interpretable models instead," *Nature Mach. Intell.*, vol. 1, no. 5, pp. 206–215, May 2019, doi: 10.1038/s42256-019-0048-x.

[56] J. Colin, T. Fel, R. Cadène, and T. Serre, "What I cannot predict, I do not understand: A human-centered evaluation framework for explainability methods," in *Adv. Neural Inf. Process.*

Syst., S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds. vol. 35, 2022, pp. 2832–2845. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/hash/13113e938f2957891c0c5e8df811dd01-Abstract-Conference.html

[57] A. Jung and P. H. J. Nardelli, "An information-theoretic approach to personalized explainable machine learning," *IEEE Signal Process. Lett.*, vol. 27, pp. 825–829, 2020, doi: 10.1109/LSP.2020.2993176.

[58] J. Chen, L. Song, M. J. Wainwright, and M. I. Jordan, "Learning to explain: An information-theoretic perspective on model interpretation," in *Proc. 35th Int. Conf. Mach. Learn.*, J. Dy and A. Krause, Eds. vol. 80, 2018, pp. 883–892. [Online]. Available: https://proceedings.mlr.press/v80/chen18j.html

[59] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should i trust you?: Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 1135–1144, doi: 10.1145/2939672.2939778.

[60] C. Molnar, *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*, 3rd ed., 2025. [Online]. Available: https://christophm.github.io/interpretable-ml-book/

[61] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *2017 IEEE Int. Conf. Comput. Vis.*, pp. 618–626, doi: 10.1109/ICCV.2017.74.

[62] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2002.

[63] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Proc. Mach. Learn. Syst.*, I. Dhillon, D. Papailiopoulos, and V. Sze, Eds. vol. 2, 2020. [Online]. Available: https://proceedings.mlsys.org/paper_files/paper/2020/hash/1f5fe83998a09396ebe6477d9475ba0c-Abstract.html

[64] A. Ünsal and M. Önen, "Information-theoretic approaches to differential privacy," *ACM Comput. Surv.*, vol. 56, no. 3, Oct. 2023, Art. no. 76, doi: 10.1145/3604904.

[65] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*. Boston, MA, USA: Kluwer Academic Publishers, 2004.

[66] D. P. Bertsekas, *Convex Optimization Algorithms*. Belmont, MA, USA: Athena Scientific, 2015.

[67] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 828–841, Oct. 2019, doi: 10.1109/TEVC.2019.2890858.

[68] R. T. Rockafellar, *Network Flows and Monotropic Optimization*. Belmont, MA, USA: Athena Scientific, 1998.

[69] M. J. Wainwright, *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge, U.K.: Cambridge Univ. Press, 2019.

[70] P. Bühlmann and S. van de Geer, *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Berlin, Germany: Springer-Verlag, 2011.

[71] C. H. Lampert, "Kernel methods in computer vision," *Found. Trends Comput. Graph. Vis.*, vol. 4, no. 3, pp. 193–285, Sep. 2009, doi: 10.1561/0600000027.

[72] D. Pfau and A. Jung, "Engineering trustworthy AI: A developer guide for empirical risk minimization," Nov. 2024. [Online]. Available: https://arxiv.org/abs/2410.19361

[73] High-Level Expert Group on Artificial Intelligence, "The assessment list for trustworthy artificial intelligence (ALTAI): For self assessment," European Commission, Jul. 17, 2020. [Online]. Available: https://digital-strategy.ec.europa.eu/en/library/assessment-list-trustworthy-artificial-intelligence-altai-self-assessment

[74] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. Hoboken, NJ, USA: Wiley, 2006.

[75] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Adv. Neural Inf. Process. Syst.*, T. Dietterich, S. Becker, and Z. Ghahramani, Eds. vol. 14, 2001, pp. 849–856. [Online]. Available: https://papers.nips.cc/paper_files/paper/2001/hash/801272ee79cfde7fa5960571fee36b9b-Abstract.html

[76] A. Vaswani et al., "Attention is all you need," in *Adv. Neural Inf. Process. Syst.*, I. Guyon, U. von Luxburg, S. Bengio, H. Wallach,

R. Fergus, S. Vishwanathan, and R. Garnett, Eds. vol. 30, 2017, pp. 5998–6008. [Online]. Available: https://papers.nips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[77] A. Papoulis and S. U. Pillai, *Probability, Random Variables, and Stochastic Processes*, 4th ed. New York, NY, USA: McGraw-Hill Higher Education, 2002.

[78] Y. SarcheshmehPour, Y. Tian, L. Zhang, and A. Jung, "Clustered federated learning via generalized total variation minimization," *IEEE Trans. Signal Process.*, vol. 71, pp. 4240–4256, 2023, doi: 10.1109/TSP.2023.3322848.

[79] E. Abbe, "Community detection and stochastic block models: Recent developments," *J. Mach. Learn. Res.*, vol. 18, no. 177, pp. 1–86, Apr. 2018. [Online]. Available: http://jmlr.org/papers/v18/16-480.html

[80] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Found. Trends Mach. Learn.*, vol. 5, no. 1, pp. 1–122, Dec. 2012, doi: 10.1561/2200000024.

[81] A. Jung, "Networked exponential families for big data over networks," *IEEE Access*, vol. 8, pp. 202 897–202 909, Nov. 2020, doi: 10.1109/ACCESS.2020.3033817.

[82] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed. New York, NY, USA: Cambridge Univ. Press, 2013.

[83] A. Rakhlin, O. Shamir, and K. Sridharan, "Making gradient descent optimal for strongly convex stochastic optimization," in *Proc. 29th Int.*

Conf. Mach. Learn., J. Langford and J. Pineau, Eds. 2012, pp. 449–456. [Online]. Available: https://icml.cc/Conferences/2012/papers/261.pdf

[84] N. Parikh and S. Boyd, "Proximal algorithms," *Found. Trends Optim.*, vol. 1, no. 3, pp. 127–239, Jan. 2014, doi: 10.1561/2400000003.

[85] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, 2nd ed. New York, NY, USA: Springer Science+Business Media, 2017.

[86] P. J. Brockwell and R. A. Davis, *Time Series: Theory and Methods*, 2nd ed. New York, NY, USA: Springer-Verlag, 1991.

[87] M. Kearns and M. Li, "Learning in the presence of malicious errors," *SIAM J. Comput.*, vol. 22, no. 4, pp. 807–837, Aug. 1993, doi: 10.1137/0222052.

[88] G. Lugosi and S. Mendelson, "Robust multivariate mean estimation: The optimality of trimmed mean," *Ann. Statist.*, vol. 49, no. 1, pp. 393–410, Feb. 2021, doi: 10.1214/20-AOS1961.

[89] O. Kallenberg, *Foundations of Modern Probability*. New York, NY, USA: Springer-Verlag, 1997.

[90] S. Ross, *A First Course in Probability*, 9th ed. Boston, MA, USA: Pearson Education, 2014.

[91] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, A.

Singh and J. Zhu, Eds. vol. 54, 2017, pp. 1273–1282. [Online]. Available: https://proceedings.mlr.press/v54/mcmahan17a.html

[92] L. Condat, "A primal–dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms," *J. Optim. Theory Appl.*, vol. 158, no. 2, pp. 460–479, Aug. 2013, doi: 10.1007/s10957-012-0245-9.

[93] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[94] M. P. Salinas et al., "A systematic review and meta-analysis of artificial intelligence versus clinicians for skin cancer diagnosis," *npj Digit. Med.*, vol. 7, no. 1, May 2024, Art. no. 125, doi: 10.1038/s41746-024-01103-x.

[95] G. F. Cooper, "The computational complexity of probabilistic inference using bayesian belief networks," *Artif. Intell.*, vol. 42, no. 2–3, pp. 393–405, Mar. 1990, doi: 10.1016/0004-3702(90)90060-D.

[96] S. Shalev-Shwartz and A. Tewari, "Stochastic methods for ℓ_1 regularized loss minimization," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, L. Bottou and M. Littman, Eds. Jun. 2009, pp. 929–936.

[97] I. Csiszar, "Generalized cutoff rates and Renyi's information measures," *IEEE Trans. Inf. Theory*, vol. 41, no. 1, pp. 26–34, Jan. 1995, doi: 10.1109/18.370121.

[98] S. Bubeck, "Convex optimization: Algorithms and complexity," *Found. Trends Mach. Learn.*, vol. 8, no. 3–4, pp. 231–357, Nov. 2015, 10.1561/2200000050.

[99] L. Bottou, "On-line learning and stochastic approximations," in *On-Line Learning in Neural Networks*, D. Saad, Ed. New York, NY, USA: Cambridge Univ. Press, 1999, ch. 2, pp. 9–42.

[100] D. P. Bertsekas, A. Nedic, and A. E. Ozdaglar, *Convex Analysis and Optimization*. Belmont, MA, USA: Athena Scientific, 2003.

[101] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA, USA: Athena Scientific, 1999.

[102] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. New York, NY, USA: Cambridge Univ. Press, 2000.

[103] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, pp. 41–75, Jul. 1997, doi: 10.1023/A:1007379606734.

[104] A. Jung, G. Hannak, and N. Goertz, "Graphical lasso based model selection for time series," *IEEE Signal Process. Lett.*, vol. 22, no. 10, pp. 1781–1785, Oct. 2015, doi: 10.1109/LSP.2015.2425434.

[105] A. Jung, "Learning the conditional independence structure of stationary time series: A multitask learning approach," *IEEE Trans. Signal Process.*, vol. 63, no. 21, Nov. 2015, doi: 10.1109/TSP.2015.2460219.

[106] High-Level Expert Group on Artificial Intelligence, "Ethics guidelines for trustworthy AI," European Commission, Apr. 8, 2019. [Online]. Available: https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai

[107] C. Gallese, "The AI act proposal: A new right to technical interpretability?," *SSRN Electron. J.*, Feb. 2023. [Online]. Available: https://ssrn.com/abstract=4398206

[108] M. Mitchell et al., "Model cards for model reporting," in *Proc. Conf. Fairness, Accountability, Transparency*, 2019, pp. 220–229, doi: 10.1145/3287560.3287596.

[109] K. Shahriari and M. Shahriari, "IEEE standard review — Ethically aligned design: A vision for prioritizing human wellbeing with artificial intelligence and autonomous systems," in *2017 IEEE Canada Int. Humanitarian Technol. Conf.*, pp. 197–201, doi: 10.1109/IHTC.2017.8058187.

[110] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. Cambridge, MA, USA: MIT Press, 2006.