El Diccionario de Aprendizaje Automático de A”alto

Alexander Jung and Konstantina Olioumtsevits

Traducido al español por Tommi Flores

May 23, 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 atributos $\mathbf{x}$ 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 esperanza de una función $f(\mathbf{z})$ de una variable aleatoria (RV) $\mathbf{z}$ cuya distribución de probabilidad es $p(\mathbf{z})$. Si la distribución de probabilidad es clara por el contexto, simplemente escribimos $\mathbb{E}\{f(\mathbf{z})\}$.
$p(\mathbf{x}, y)$	Una distribución de probabilidad conjunta de una RV cuyas realizaciónes son punto de datoss con atributos $\mathbf{x}$ y etiqueta y.
$p(\mathbf{x}\|y)$	Una distribución de probabilidad condicional de una RV $\mathbf{x}$ dado el valor de otra RV y [5, Sec. 3.5].
$p(\mathbf{x}; \mathbf{w})$	Una distribución de probabilidad parametrizada de una RV $\mathbf{x}$. La distribución de probabilidad 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 matriz de covarianza $\mathbb{E}\Big\{\big(\mathbf{x}-\mathbb{E}\{\mathbf{x}\}\big)\big(\mathbf{x}-\mathbb{E}\{\mathbf{x}\}\big)^T\Big\}$.
$\mathcal{N}(\mu, \sigma^2)$	La distribución de probabilidad de una variable aleatoria gaussiana (VA gaussiana) $x \in \mathbb{R}$ con media (o esperanza) $\mu = \mathbb{E}\{x\}$ 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 esperanza) $\boldsymbol{\mu} = \mathbb{E}\{\mathbf{x}\}$ y matriz de covarianza $\mathbf{C} = \mathbb{E}\big\{\big(\mathbf{x}-\boldsymbol{\mu}\big)\big(\mathbf{x}-\boldsymbol{\mu}\big)^T\big\}$.

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 atributos que caracterizan un punto de datos.
x_j	La j-ésima característica (feature) de un punto de datos. La primera atributo se denota como x_1, la segunda como x_2, y así sucesivamente.
$\mathbf{x}$	El vector de atributos (feature vector) $\mathbf{x} = \big(x_1, \ldots, x_d\big)^T$ de un punto de datos cuyas entradas son los atributos individuales de un punto de datos.
$\mathcal{X}$	El espacio de atributos $\mathcal{X}$ es el conjunto de todos los valores posibles que las atributos $\mathbf{x}$ 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 atributos individuales de un punto de datos. Necesitamos dos símbolos diferentes para distinguir entre atributos (features) crudas y atributos aprendidos [6, Ch. 9].
$\mathbf{x}^{(r)}$	El vector de atributos de el r-ésimo punto de datos dentro de un conjunto de datos.
$x_j^{(r)}$	La j-ésima atributo 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 atributos 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 tasa de aprendizaje (o tamaño de paso) utilizada por los métodos de métodos de gradiente.
$h(\cdot)$	Un mapa de hipótesis que toma como entrada las atributos $\mathbf{x}$ 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 espacio de hipótesis o modelo utilizado por un método de ML. El espacio de hipótesis 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 espacio de hipótesis $\mathcal{H}$.
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 (parámetros 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 pérdida 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 atributos $\mathbf{x}$ del punto de datos.
E_v	El error de validación de una hipótesis h, que es su pérdida promedio incurrida en un conjunto de validación.
$\widehat{L}\big(h\|\mathcal{D}\big)$	El riesgo empírico o pérdida 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 pérdida 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 regularización que controla la cantidad de regularización.
$\lambda_j\big(\mathbf{Q}\big)$	El j-ésimo valor propio (eigenvalue) (ordenado en forma ascendente o descendente) de una matriz semi-definida positiva (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 red neuronal artificial (RNA).
$\mathcal{R}_{\hat{y}}$	Una región de decisión dentro de un espacio de atributos.
$\mathbf{w}$	Un vector de parámetros $\mathbf{w} = \big(w_1, \ldots, w_d\big)^T$ de un modelo, por ejemplo, los pesos de un modelo lineal o en una ANN.
$h^{(\mathbf{w})}(\cdot)$	Un mapa de hipótesis que involucra parámetros del modelo ajustables $w_1, \ldots, w_d$ apilados en el vector $\mathbf{w} = \big(w_1, \ldots, w_d\big)^T$.
$\phi(\cdot)$	Un mapa de atributos $\phi : \mathcal{X} \to \mathcal{X}' : \mathbf{x} \mapsto \mathbf{x}' := \phi\big(\mathbf{x}\big) \in \mathcal{X}'$.
$K\big(\cdot, \cdot\big)$	Dado un espacio de atributos $\mathcal{X}$, 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 grafo no dirigido cuyos nodos $i \in \mathcal{V}$ representan dispositivos dentro de un red de aprendizaje federado (red FL). Los bordes ponderados no dirigidos $\mathcal{E}$ representan conectividad entre dispositivos y similitudes estadísticas entre sus conjunto de datoss y tarea de aprendizajes.
$i \in \mathcal{V}$	Un nodo que representa un dispositivo dentro de un red de aprendizaje federado (red FL). El dispositivo puede acceder a un conjunto de datos local y entrenar un modelo local.
$\mathcal{G}^{(\mathcal{C})}$	El subgrafo inducido de $\mathcal{G}$ utilizando los nodos en $\mathcal{C} \subseteq \mathcal{V}$.
$\mathbf{L}^{(\mathcal{G})}$	La matriz laplaciana de un grafo $\mathcal{G}$.
$\mathbf{L}^{(\mathcal{C})}$	La matriz laplaciana del grafo inducido $\mathcal{G}^{(\mathcal{C})}$.
$\mathcal{N}^{(i)}$	La entorno de un nodo i en un grafo $\mathcal{G}$.
$d^{(i)}$	El grado ponderado $d^{(i)} := \sum_{i' \in \mathcal{N}^{(i)}} A_{i,i'}$ de un nodo i en un grafo $\mathcal{G}$.
$d_{\max}^{(\mathcal{G})}$	El máximo grado ponderado de nodo en un grafo $\mathcal{G}$.
$\mathcal{D}^{(i)}$	El conjunto de datos local $\mathcal{D}^{(i)}$ 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 tamaño de la muestra) contenidos en el conjunto de datos local $\mathcal{D}^{(i)}$ en el nodo $i \in \mathcal{V}$.

$\mathbf{x}^{(i,r)}$	Las atributos del r-ésimo punto de datos en el conjunto de datos local $\mathcal{D}^{(i)}$.
$y^{(i,r)}$	La etiqueta del r-ésimo punto de datos en el conjunto de datos local $\mathcal{D}^{(i)}$.
$\mathbf{w}^{(i)}$	Los parámetros del modelo locales del dispositivo i dentro de un red de aprendizaje federado (red FL).
$L_i\left(\mathbf{w}\right)$	La función de pérdida local utilizada por el dispositivo i para medir la utilidad de alguna elección $\mathbf{w}$ para los parámetros del modelo locales.
$L^{(\mathrm{d})}\left(\mathbf{x}, h\big(\mathbf{x}\big), h'\big(\mathbf{x}\big)\right)$	La pérdida incurrida por una hipótesis h' en un punto de datos con atributos $\mathbf{x}$ 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 parámetros del modelo locales $\mathbf{w}^{(i)} \in \mathbb{R}^d$.

Machine Learning Concepts

k-means El algoritmo k-medias es un método de agrupamiento rígido que asigna cada punto de datos de un conjunto de datos a precisamente uno de k clústers diferentes. El método alterna entre actualizar las asignaciones de clúster (asignando al clúster con el media mas cercano) y, dado estas asignaciones actualizadas recalcular los medias de los clúster [6, Ch. 8].

agrupamiento Los métodos de agrupamiento descomponen un conjunto dado de punto de datoss en pocos subconjuntos, que se denominan clústers. Cada clúster está compuesto por punto de datoss que son más similares entre sí que con respecto a los punto de datoss fuera del clúster. Los diferentes métodos de agrupamiento utilizan distintas medidas de similitud entre punto de datoss y diferentes formas de representación de los clústers. El método de agrupamiento k-means utiliza el vector de atributo promedio (media del clúster) de un clúster como su representante. Un método popular de agrupamiento suave basado en modelo de mezcla gaussiana (GMM) representa un clúster mediante una distribución normal multivariante.

agrupamiento basado en densidad para aplicaciones espaciales con ruido (DBSCAN) DBSCAN es un algoritmo de agrupamiento para punto de datoss caracterizados por vector de atributoss numéricos. Al igual que k-means y

agrupamiento suave mediante GMM, DBSCAN utiliza las distancias euclidianas entre los vector de atributoss para determinar los clústers. Sin embargo, a diferencia de k-means y GMM, DBSCAN emplea una noción diferente de similitud entre punto de datoss. DBSCAN considera que dos punto de datoss son similares si están conectados a través de una secuencia (camino) de punto de datoss intermedios cercanos. Por lo tanto, DBSCAN puede considerar que dos punto de datoss son similares (y, por lo tanto, pertenecen al mismo clúster) incluso si sus vector de atributoss tienen una gran distancia euclidiana.

agrupamiento basado en flujo El agrupamiento basado en flujo agrupa los nodos de un grafo no dirigido aplicando el algoritmo de k-means sobre vector de atributoss específicos para cada nodo. Estos vector de atributoss se construyen a partir de flujos de red entre nodos fuente y destino seleccionados cuidadosamente [7].

agrupamiento convexo Considere un conjunto de datos $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)} \in \mathbb{R}^d$. El agrupamiento Convexo aprende vectores $\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(m)}$ minimizando

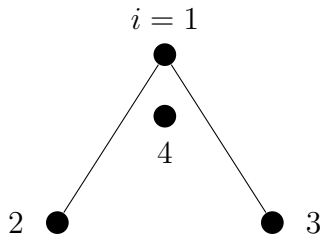
$$\sum_{r=1}^{m} \left\| \mathbf{x}^{(r)} - \mathbf{w}^{(r)} \right\|_2^2 + \alpha \sum_{i,i' \in \mathcal{V}} \left\| \mathbf{w}^{(i)} - \mathbf{w}^{(i')} \right\|_p .$$

Aquí, $\|\mathbf{u}\|_p := \left(\sum_{j=1}^{d} |u_j|^p \right)^{1/p}$ denota la p-norma (para $p \geq 1$). Resulta que muchos de los vectores óptimos $\widehat{\mathbf{w}}^{(1)}, \ldots, \widehat{\mathbf{w}}^{(m)}$ coinciden. Un clúster consiste entonces en aquellos punto de datoss $r \in \{1, \ldots, m\}$ con valores idénticos de $\widehat{\mathbf{w}}^{(r)}$ [8, 9].

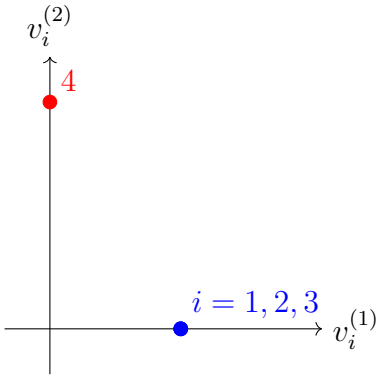
agrupamiento en grafos El agrupamiento en grafos tiene como objetivo

agrupar punto de datoss que están representados como nodos de un grafo $\mathcal{G}$. 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 peso de arista [7, 10].

agrupamiento espectral El agrupamiento 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 grafo $\mathcal{G}$. El agrupamiento espectral utiliza los vector propios de la matriz laplaciana $\mathbf{L}^{(\mathcal{G})}$ para construir vector de atributoss $\mathbf{x}^{(i)} \in \mathbb{R}^d$ para cada nodo (es decir, para cada punto de datos) $i = 1, \ldots, n$. Podemos utilizar estos vector de atributoss como entrada para métodos de agrupamiento en espacio euclidiano,como k-means o agrupamiento suave mediante GMM. Mas o menos, los vector de atributoss de los nodos que pertenecen a un subconjunto bien conectado (o clúster) de nodos en $\mathcal{G}$ estan ubicados cerca en el espacio euclidiano $\mathbb{R}^d$ (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 grafo no dirigido $\mathcal{G}$ con cuatro nodos $i = 1, 2, 3, 4$, donde cada nodo representa un punto de datos. Derecha: La matriz laplaciana $\mathbf{L}^{(\mathcal{G})} \in \mathbb{R}^{4\times 4}$ y su EVD. **Abajo.** Izquierda: Un diagrama de dispersión de los punto de datoss usando los vector de atributoss $\mathbf{x}^{(i)} = \left(v_i^{(1)}, v_i^{(2)}\right)^T$. Derecha: Dos vector propios $\mathbf{v}^{(1)}, \mathbf{v}^{(2)} \in \mathbb{R}^d$ correspondientes al valor propio $\lambda = 0$ de la matriz laplaciana $\mathbf{L}^{(\mathcal{G})}$.

agrupamiento rígido El agrupamiento rígido se refiere a la tarea de dividir un conjunto dado de punto de datoss en (pocos) clústers no superpuestos. El método de agrupamiento rígido más utilizado es k-means.

agrupamiento suave El agrupamiento suave se refiere a la tarea de dividir un conjunto dado de punto de datoss en (pocos) clústers superpuestos. Cada punto de datos se asigna a varios clústers diferentes con diversos grados de pertenencia. Los métodos de agrupamiento suave determinan el grado de pertenencia (o asignación suave a un clúster) para cada punto de datos y cada clúster. Un enfoque fundamentado para el agrupamiento suave consiste en interpretar los punto de datoss como independientes e idénticamente distribuidos (i.i.d.) realizacións de un GMM. De este modo, se obtiene una elección natural para el grado de pertenencia como la probabilidad condicional de que un punto de datos pertenezca a un componente específico de la mezcla.

algoritmo Un algoritmo es una especificación precisa y paso a paso de cómo producir una salida a partir de una entrada dada en un número finito de pasos computacionales [11]. Por ejemplo, un algoritmo para entrenar un modelo lineal describe explícitamente cómo transformar un conjunto de entrenamiento dado en parámetros del modelo a través de una secuencia de paso de gradientes. Esta caracterización informal puede formalizarse rigurosamente mediante diferentes modelos matemáticos [12]. Un modelo simple de un algoritmo es una colección de ejecuciones posibles. Cada ejecución es una secuencia:

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

que respeta las restricciones inherentes al ordenador que ejecuta el algoritmo. Los algoritmos pueden ser deterministas, donde cada entrada resulta únicamente en una sola ejecución, o aleatorios, donde las ejecuciones pueden variar probabilísticamente. Los algoritmos aleatorios pueden analizarse modelando las secuencias de ejecución como resultados de experimentos aleatorios, considerando el algoritmo como un proceso estocástico [5, 13, 14]. Crucialmente, un algoritmo abarca más que solo un mapeo de entrada a salida; también incluye los pasos computacionales intermedios $s_1, \ldots, s_T$.

algoritmo distribuido Un algoritmo distribuido distributed es un algoritmo 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 [15, 16]. A diferencia de un algoritmo clásico, que se ejecuta en un solo dispositivo, un algoritmo distribuido algoritmo se ejecuta de forma concurrente en múltiples dispositivos 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 dispositivo 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 aprendizaje federado (FL), donde una red de dispositivos colabora para entrenar un modelo personalizado por dispositivo..

algoritmo en línea Un algoritmo en línea es un algoritmo 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 [17, 18]. 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 descenso por gradiente en línea (online GD), que actualiza parámetros del modelo conforme llegan nuevos punto de datoss.

análisis de componentes principales (PCA) El PCA determina una mapa de atributos lineal tales que las nuevos atributos permiten reconstruir los atributos originales con el mínimo error de reconstrucción [6].

análisis de componentes principales probabilístico (PPCA) El análisis de componentes principales probabilístico (PPCA) extiende el análisis

de componentes principales (PCA) básico mediante el uso de un modelo probabilístico para los punto de datoss. El modelo probabilístico de PPCA reduce la tarea de reducción de dimensionalidad a un problema de estimación que se puede resolver utilizando métodos de EM.

application programming interface (API) Una API es un mecanismo formal para permitir que componentes de software interactúen de manera estructurada [19]. En el contexto de ML, las APIs se utilizan frecuentemente para hacer accesible un ML modelo entrenado a diferentes tipos de usuarios. Estos usuarios, que pueden ser otros ordenadores o humanos, pueden solicitar una predicción para la etiqueta de un punto de datos al proporcionar sus atributos. La estructura interna del ML modelo permanece oculta para el usuario. Por ejemplo, considera un ML modelo entrenado $\widehat{h}(x) := 2x + 1$. Una API permite a un usuario enviar el valor de atributo $x = 3$ y obtener la respuesta $\widehat{h}(3) = 7$ sin conocimiento de la estructura detallada del ML modelo o su entrenamiento. En la práctica, el ML modelo suele estar alojado en un ordenador (es decir, un servidor) conectado a internet. Otro ordenador (es decir, un cliente) envía las atributos de un punto de datos al servidor, que luego calcula $\widehat{h}(\mathbf{x})$ y devuelve el resultado al sistema externo. Las APIs ayudan a modularizar el desarrollo de aplicaciones de ML al desacoplar tareas específicas. Por ejemplo, un equipo puede concentrarse en desarrollar y entrenar el modelo, mientras que otro equipo se encarga de la interacción con el usuario y la integración del modelo en aplicaciones.

aprendizaje automático (ML) El aprendizaje automático tiene como ob-

jetivo predecir una etiqueta a partir de las atributos de un punto de datos. Los métodos de ML logran esto aprendiendo una hipótesis de un espacio de hipótesis (o modelo) mediante la minimización de una función de pérdida [6, 20]. Una formulación precisa de este principio es el minimización empírica del riesgo (ERM). Diferentes métodos de ML se obtienen de distintas elecciones de diseño para los punto de datoss (sus atributos y etiqueta), el modelo, y la función de pérdida [6, Cap. 3].

aprendizaje automático explicable (explainable ML) El aprendizaje automático explicable (explainable ML) consiste en métodos de ML que tienen como objetivo complementar cada predicción con una explicación explícita de cómo se ha obtenido dicha predicción.La construcción de una explicación explícita puede no ser necesaria si el método de ML utiliza un modelo lo suficientemente simple (o interpretable) [21].

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

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

que recibe como entrada los atributos crudos $\mathbf{x} \in \mathcal{X}$ de un punto de datos y entrega nuevos atributos $\mathbf{x}' \in \mathcal{X}'$ de un nuevo espacio de atributos $\mathcal{X}'$. Se obtienen diferentes métodos de aprendizaje de atributos a partir de diferentes elecciones de $\mathcal{X}, \mathcal{X}'$, de un espacio de hipótesis $\mathcal{H}$ de posibles mapeos $\mathbf{\Phi}$, y de una medida cuantitativa de la utilidad de un mapeo específico $\mathbf{\Phi} \in \mathcal{H}$. Por ejemplo, PCA utiliza $\mathcal{X} := \mathbb{R}^d$, $\mathcal{X}' := \mathbb{R}^{d'}$ con

$d' < d$, y un espacio de hipótesis

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

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 (FL) FL es un término general para los métodos de ML que entrenan modelos de manera colaborativa, utilizando datos y computación descentralizadas.

aprendizaje federado agrupado (CFL) El FL agrupado asume que los conjunto de datos locals están naturalmente organizados en clústers. Los conjunto de datos locals que pertenecen al mismo clúster comparten propiedades estadísticas similares. El FL agrupado agrega los conjunto de datos locals dentro del mismo clúster para formar un conjunto de entrenamiento para el entrenamiento de un modelo específico del clúster. Minimización de variación total generalizada (GTVMin) facilita esta agrupación de manera implícita al imponer una similitud aproximada de los parámetros del modelo a través de subconjuntos bien conectados del red de aprendizaje federado (red FL).

aprendizaje federado en red (NFL) El aprendizaje federado en red (NFL) se refiere a métodos que aprenden modelos personalizados de manera distribuida. Estos métodos aprenden a partir de conjunto de datos locals que están relacionados por una estructura de red intrínseca.

aprendizaje federado horizontal (horizontal FL) El aprendizaje federado horizontal utiliza conjunto de datos locals constituidos por diferentes punto de datoss, pero emplea las mismas atributos para caracterizarlos [22]. 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 atributos de diferentes regiones espaciotemporales. Cada región espaciotemporal representa un punto de datos individual, caracterizado por las mismas atributos (por ejemplo, temperatura diaria o presión atmosférica).

aprendizaje federado vertical (FL vertical) El aprendizaje federado vertical utiliza conjunto de datos locals formados por los mismos punto de datoss, pero caracterizados mediante diferentes atributos [23]. 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 pesos 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 [24]. 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 pérdida incurrida por h y la pérdida incurrida por h' [18]. La referencia (baseline) hipótesis de referencia h' también se denomina experto.

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 parámetros del modelo útiles. Un ejemplo importante de técnica de optimización iterativa es el descenso por gradiente (GD).

aspectos estadísticos Por aspectos estadísticos de un método de ML, nos referimos a las propiedades de la distribución de probabilidad de su

salida bajo un modelo probabilístico para los datos introducidos en el método.

ataque de denegación de servicio (DoS) Un ataque de denegación de servicio (DoS) tiene como objetivo (por ejemplo, mediante envenenamiento de datos) dirigir el entrenamiento de un modelo para que tenga un rendimiento deficiente con punto de datos típicos.

atributo Un atributo de un punto de datos es una de sus propiedades que se puede medir o calcular fácilmente sin la necesidad de supervisión humana. Por ejemplo, si un punto de datos es una imagen digital (por ejemplo, almacenada como un archivo `.jpeg`), entonces podríamos usar las intensidades rojo-verde-azul de sus píxeles como atributos. Sinónimos específicos del dominio para el término atributo incluyen "covariable", "variable explicativa", "variable independiente", "variable de entrada", "predictor (variable)" o "regresor" [25], [26], [27].

atributo sensible La ML busca aprender una hipótesis que prediga la etiqueta de un punto de datos a partir de sus atributos. 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.

aumentación de datos Los métodos de aumentación de datos añaden punto de datoss sintéticos a un conjunto existente de punto de datoss. Estos punto de datoss sintéticos se obtienen mediante perturbaciones (por

ejemplo, añadir ruido a mediciones físicas) o transformaciones (por ejemplo, rotaciones de imágenes) de los punto de datoss originales. Estas perturbaciones y transformaciones son tales que los punto de datoss sintéticos resultantes deben tener la misma etiqueta. Como ejemplo, una imagen de un gato rotada sigue siendo una imagen de un gato, aunque sus vector de atributoss (obtenidos al apilar las intensidades de color de los píxeles) sean muy diferentes (ver Figura 2). La aumentación de datos puede ser una forma eficiente de regularización.

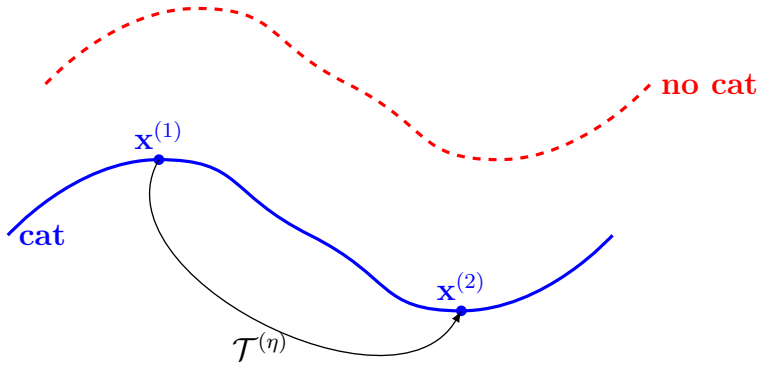


Figure 2: La aumentación de Datos aprovecha las simetrías intrínsecas de los punto de datoss en algún espacio de atributos $\mathcal{X}$. Podemos representar una simetría mediante un operador $\mathcal{T}^{(\eta)} : \mathcal{X} \to \mathcal{X}$, parametrizado por algún número $\eta \in \mathbb{R}$. Por ejemplo, $\mathcal{T}^{(\eta)}$ podría representar el efecto de rotar una imagen de un gato η grados. Un punto de datos con vector de atributos $\mathbf{x}^{(2)} = \mathcal{T}^{(\eta)}\left(\mathbf{x}^{(1)}\right)$ debete tener la misma etiqueta $y^{(2)} = y^{(1)}$ que un punto de datos con vector de atributos $\mathbf{x}^{(1)}$.

autoencoder Un autoencoder es un método de ML que aprende simultáneamente 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 pérdida calculada a partir del error de reconstrucción $\mathbf{x} - h^*\big(h\big(\mathbf{x}\big)\big)$.

bagging El bagging (o agregación por bootstrap) es una técnica genérica para mejorar (la robustez de) un método de ML dado. La idea es utilizar el bootstrap para generar copias perturbadas de un conjunto de datos dado y luego aprender una hipótesis separada para cada copia. Posteriormente, se predice la etiqueta de un punto de datos combinando o agregando las prediccións individuales de cada hipótesis separada. Para mapas de hipótesis que entregan valores numéricos de etiqueta, esta agregación podría implementarse calculando el promedio de las prediccións individuales.

bandido multi-brazo (MAB) Un problema de bandido multi-brazo (MAB) modela un escenario de toma de decisiones repetida en el que, en cada paso temporal k, un aprendiz debe elegir una de varias acciones posibles, a menudo denominadas brazos, de un conjunto finito $\mathcal{A}$. Cada brazo $a \in \mathcal{A}$ produce una recompensa estocástica $r^{(a)}$ extraída de una distribución de probabilidad desconocida con media $\mu^{(a)}$. El objetivo del aprendiz es maximizar la recompensa acumulada a lo largo del tiempo mediante un equilibrio estratégico entre exploración (recopilar información sobre brazos inciertos) y explotación (seleccionar brazos que se sabe que funcionan bien). Este equilibrio se cuantifica mediante la noción de arrepentimiento (regret), que mide la brecha de rendimiento entre la estrategia del aprendiz y la estrategia óptima que siempre selecciona el mejor brazo. Los problemas MAB forman un modelo fundamental

en aprendizaje en línea, aprendizaje por refuerzo y diseño experimental secuencial [28].

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 RVsi.i.d. con una distribución de probabilidad 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 distribución de probabilidad subyacente $p(\mathbf{z})$.

bosque aleatorio Un bosque aleatorio es un conjunto (o ensamblaje) de diferentes árbol de decisións. Cada uno de estos árbol de decisións se obtiene al ajustar una copia perturbada del conjunto de datos original.

caracterización min-max de Courant–Fischer–Weyl Considere una psd matriz $\mathbf{Q} \in \mathbb{R}^{d \times d}$ con EVD (o descomposición espectral),

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

Aquí, utilizamos los valor propios ordenados (de forma creciente)

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

La caracterización min-max de Courant–Fischer–Weyl [3, Th. 8.1.2] representa los valor propios de $\mathbf{Q}$ como las soluciones a ciertos problemas de optimización.

clasificación La clasificación es la tarea de determinar una etiqueta y con valor discreto para un punto de datos, basado únicamente en sus atributos $\mathbf{x}$. 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 región de decisiónes $\mathcal{R}_a$, para cada valor posible de etiqueta $a \in \mathcal{Y}$.

clasificador lineal Consideremos punto de datoss caracterizados por atributos 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 región de decisións que están separadas por hiperplanos en $\mathbb{R}^d$ [6, Ch. 2].

clúster Un clúster es un subconjunto de punto de datoss que son más similares entre sí que a los punto de datoss fuera del clúster. La medida cuantitativa de similitud entre los punto de datoss es una decisión de diseño. Si los punto de datoss se caracterizan por vectores de atributo euclidianos $\mathbf{x} \in \mathbb{R}^d$, podemos definir la similitud entre dos punto de datoss a través de la distancia euclidiana entre sus vectores de atributo.

condición de gradiente cero Considera el problema de optimización sin restricciones $\min_{\mathbf{w} \in \mathbb{R}^d} f(\mathbf{w})$ con una función objetivo suave y convexo $f(\mathbf{w})$. Una condición necesaria y suficiente para que un vector $\widehat{\mathbf{w}} \in \mathbb{R}^d$ resuelva este problema es que el gradiente $\nabla f\big(\widehat{\mathbf{w}}\big)$ sea el vector cero,

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

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 modelo (por ejemplo, a través de ERM) y para la validación de modelos. 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 3 muestra un conjunto de datos que consiste en vacas como punto de datoss.

Figure 3: “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 [29], [30], [31], [32]. Uno de los modelos de datos más adoptados es el modelo relacional, modelo, que organiza los datos en una tabla (o relación) [33], [29]. 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 atributos y etiquetas del punto de datos.

Por ejemplo, la Tabla 1 muestra una representación del conjunto de datos en la Figura 3. 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 atributos 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 [34].

conjunto de datos local El concepto de un conjunto de datos local se encuentra entre el concepto de un punto de datos y un conjunto de datos. Un conjunto de datos local consiste en varios punto de datoss individuales, que se caracterizan por atributos y etiquetas. A diferencia de un

único conjunto de datos utilizado en métodos ML básicos, un conjunto de datos local también está relacionado con otros conjunto de datoss locales mediante diferentes nociones de similitud. Estas similitudes pueden surgir de modelo probabilísticos o de la infraestructura de comunicación y están codificadas en las aristas de un red de aprendizaje federado (red FL).

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 pérdida 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 espacio de hipótesis 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 modelo (por ejemplo, mediante ERM) ni en un conjunto de validación para elegir entre diferentes modelos.

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 pérdida 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 unespacio de hipótesis diferente).

convexo Un subconjunto $\mathcal{C} \subseteq \mathbb{R}^d$ del espacio euclidiano $\mathbb{R}^d$ se denomina convexo si contiene el segmento de recta entre cualesquiera dos puntos $\mathbf{x}, \mathbf{y} \in \mathcal{C}$ en ese conjunto. Una función $f : \mathbb{R}^d \to \mathbb{R}$ es convexa si su epígrafo $\left\{ \left(\mathbf{w}^T, t \right)^T \in \mathbb{R}^{d+1} : t \geq f(\mathbf{w}) \right\}$ es un conjunto convexo [35]. Ilustramos un ejemplo de un conjunto convexo y una función convexa en la Figura 4.

Figure 4: Izquierda: Un conjunto convexo $\mathcal{C} \subseteq \mathbb{R}^d$. Derecha: Una función convexa $f : \mathbb{R}^d \to \mathbb{R}$.

cota superior de confianza (UCB) Considera una aplicación de ML que requiere seleccionar, en cada paso temporal k, una acción a_k de un conjunto finito de alternativas $\mathcal{A}$. La utilidad de seleccionar la acción a_k se cuantifica mediante una señal numérica de recompensa $r^{(a_k)}$. Un modelo probabilístico ampliamente utilizado para este tipo de problema de toma de decisiones secuencial es el entorno de bandidos multi-brazo estocásticos [28]. En este modelo, la recompensa $r^{(a)}$ se considera como la realización de una RV con media desconocida $\mu^{(a)}$. Idealmente, siempre elegiríamos la acción con la mayor recompensa esperada $\mu^{(a)}$, pero estas medias son desconocidas y deben estimarse a partir de datos observados. Simplemente elegir la acción con la mayor estimación $\widehat{\mu}^{(a)}$

puede conducir a resultados subóptimos debido a la incertidumbre en la estimación. La estrategia UCB aborda esto seleccionando acciones no solo basándose en sus medias estimadas, sino también incorporando un término que refleja la incertidumbre en estas estimaciones, favoreciendo acciones con alto potencial de recompensa y alta incertidumbre. Las garantías teóricas para el rendimiento de las estrategias UCB, incluyendo cotas de arrepentimiento logarítmicas, se establecen en [28].

criterio de parada Muchos métodos de ML utilizan algoritmos iterativos que construyen una secuencia de parámetros del modelo (como los pesos de un mapa lineal o los pesos de una ANN). Estos parámetros (idealmente) convergen a una elección óptima para los parámetros del modelo. En la práctica, dados recursos computacionales finitos, necesitamos detener la iteración después de un número finito de repeticiones. Un criterio de parada es cualquier condición bien definida requerida para detener la iteración.

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 [33].

datos en red Los datos en red consisten en conjunto de datos locals relacionados por alguna noción de similitud por pares. Podemos representar

los datos en red utilizando un grafo cuyos nodos contienen conjunto de datos locals y cuyas aristas codifican similitudes por pares. Un ejemplo de datos en red surge en las aplicaciones de FL donde los conjunto de datos locals son generados por dispositivos distribuidos espacialmente.

datos faltantes Considere un conjunto de datos constituido por punto de datoss recopilados a través de algún dispositivo físico. Debido a imperfecciones y fallas, algunos de los valores de atributo 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 atributo corrupta.

descenso de gradiente estocástico (SGD) El descenso de gradiente estocástico es una variante del GD en la que se reemplaza el gradiente de la función objetivo por una aproximación estocástica. Una aplicación principal del descenso por gradiente estocástico es entrenar un modelo parametrizado mediante ERM sobre un conjunto de entrenamiento $\mathcal{D}$ que es muy grande o no está fácilmente disponible (por ejemplo, cuando los punto de datoss están almacenados en bases de datos distribuidas por todo el mundo). Para evaluar el gradiente del riesgo empírico (como función de los parámetros del modelo $\mathbf{w}$), se requiere calcular una suma $\sum_{r=1}^{m} \nabla_{\mathbf{w}} L\left(\mathbf{z}^{(r)}, \mathbf{w}\right)$ sobre todos los punto de datoss del conjunto de entrenamiento. Una aproximación estocástica se obtiene al reemplazar esta suma $\sum_{r=1}^{m} \nabla_{\mathbf{w}} L\left(\mathbf{z}^{(r)}, \mathbf{w}\right)$ por una suma parcial $\sum_{r \in \mathcal{B}} \nabla_{\mathbf{w}} L\left(\mathbf{z}^{(r)}, \mathbf{w}\right)$ considerando un subconjunto aleatorio $\mathcal{B} \subseteq \{1, \ldots, m\}$ (see Figure 5).

A estos punto de datoss seleccionados aleatoriamente se les denomina a menudo lote. El tamaño del lote, denotado $|\mathcal{B}|$ es un parámetro importante del descenso por gradiente estocástico. GD. El descenso por gradiente estocástico con $|\mathcal{B}| > 1$ se conoce como mini-lote SGD [37].

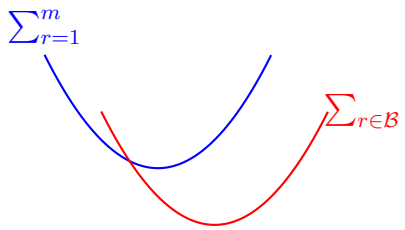


Figure 5: El descenso por gradiente estocástico (GD) para ERM aproxima el gradiente $\sum_{r=1}^{m} \nabla_{\mathbf{w}} L\left(\mathbf{z}^{(r)}, \mathbf{w}\right)$ reemplazando la suma sobre todos los punto de datoss del conjunto de entrenamiento (indexados por $r = 1, \ldots, m$) con una suma sobre un subconjunto aleatorio $\mathcal{B} \subseteq \{1, \ldots, m\}$.

descenso por gradiente (GD) El gradiente desceso por gradiente es un método iterativo para encontrar el mínimo de una función diferenciable $f(\mathbf{w})$ con un argumento vectorial $\mathbf{w} \in \mathbb{R}^d$. Considera una estimación o aproximación actual $\mathbf{w}^{(k)}$ para el mínimo de la función $f(\mathbf{w})$. Queremos encontrar un nuevo vector (mejor) $\mathbf{w}^{(k+1)}$ que tenga un valor objetivo menor $f(\mathbf{w}^{(k+1)}) < f\left(\mathbf{w}^{(k)}\right)$ que la estimación actual $\mathbf{w}^{(k)}$. Esto se puede lograr típicamente utilizando un paso de gradiente.

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

tamaño de paso suficientemente pequeño $\eta > 0$. La Figura 6 ilustra el efecto de un solo paso de gradiente (2).

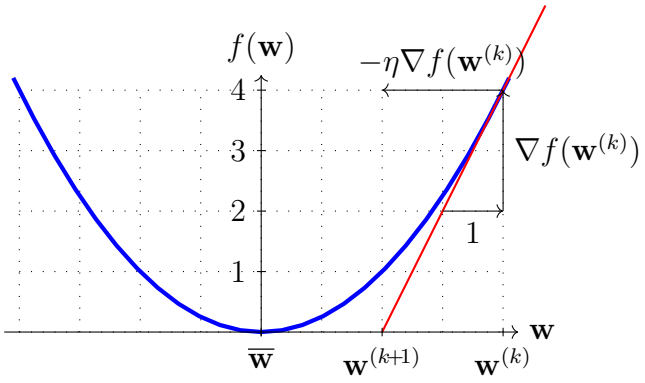


Figure 6: Un paso de gradiente singular (2) hacia el minimizador $\overline{\mathbf{w}}$ de $f(\mathbf{w})$.

descenso por gradiente en línea (online GD) Considere un método de ML que aprende los parámetros del modelo $\mathbf{w}$ a partir de un espacio de parámetros $\mathcal{W} \subseteq \mathbb{R}^d$. El proceso de aprendizaje utiliza punto de datoss $\mathbf{z}^{(t)}$ que llegan en instantes de tiempo consecutivos $t = 1, 2, \ldots$. Interpretamos los punto de datoss $\mathbf{z}^{(t)}$ como copias i.i.d. de una RV $\mathbf{z}$. El riesgo $\mathbb{E}\{L(\mathbf{z}, \mathbf{w})\}$ de una hipótesis $h^{(\mathbf{w})}$ puede entonces (bajo condiciones suaves) obtenerse como el límite $\lim_{T\to\infty}(1/T)\sum_{t=1}^{T} L\left(\mathbf{z}^{(t)}, \mathbf{w}\right)$. Podríamos usar este límite como la función objetivo para aprender los parámetros del modelo $\mathbf{w}$. Sin embargo, este límite solo puede evaluarse si esperamos un tiempo infinito para recolectar todos los punto de datoss. Algunas aplicaciones de ML requieren métodos que aprendan en línea: tan pronto como llega un nuevo punto de datos $\mathbf{z}^{(t)}$ en el tiempo t, actualizamos los parámetros del modelo actuales $\mathbf{w}^{(t)}$. Nótese que el nuevo punto de datos $\mathbf{z}^{(t)}$ contribuye con el componente $L\left(\mathbf{z}^{(t)}, \mathbf{w}\right)$ al riesgo. Como sugiere el nombre, el descenso por gradiente en línea

actualiza $\mathbf{w}^{(t)}$ mediante un (proyectado) paso de gradiente:

$$\mathbf{w}^{(t+1)} := P_{\mathcal{W}}\big(\mathbf{w}^{(t)} - \eta_t \nabla_{\mathbf{w}} L\left(\mathbf{z}^{(t)}, \mathbf{w}\right)\big). \tag{3}$$

Nótese que (3) es un paso de gradiente para el componente actual $L\left(\mathbf{z}^{(t)}, \cdot\right)$ del riesgo. La actualización (3) ignora todos los componentes anteriores $L\left(\mathbf{z}^{(t')}, \cdot\right)$ para $t' < t$. Por tanto, podría suceder que, comparado con $\mathbf{w}^{(t)}$, los parámetros del modelo actualizados $\mathbf{w}^{(t+1)}$ aumenten el pérdida promedio retrospectivo $\sum_{t'=1}^{t-1} L\left(\mathbf{z}^{(t')}, \cdot\right)$. Sin embargo, para un tasa de aprendizaje η_t elegido apropiadamente, se puede demostrar que el descenso por gradiente en línea es óptimo en escenarios de interés práctico. Por óptimo, entendemos que los parámetros del modelo $\mathbf{w}^{(T+1)}$ entregados por el descenso por gradiente en línea tras observar T punto de datoss $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(T)}$ son al menos tan buenos como los entregados por cualquier otro método de aprendizaje [17, 38].

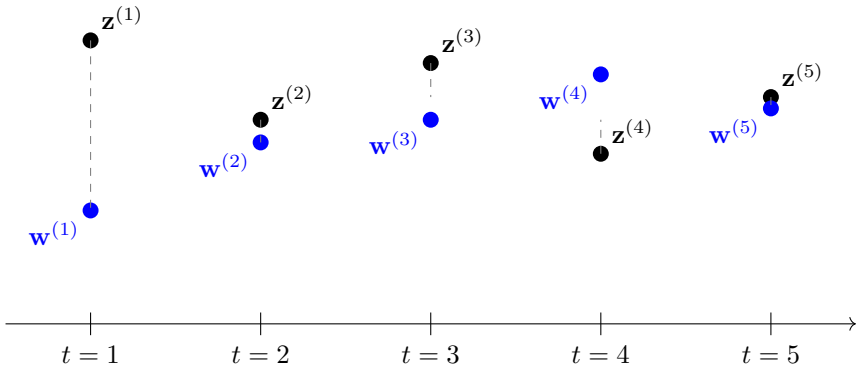


Figure 7: Una instancia de descenso por GD en línea que actualiza los parámetros del modelo $\mathbf{w}^{(t)}$ usando el punto de datos $\mathbf{z}^{(t)} = x^{(t)}$ que llega en el tiempo t. Esta instancia emplea la pérdida de error cuadrático $L\left(\mathbf{z}^{(t)}, w\right) = (x^{(t)} - w)^2$.

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

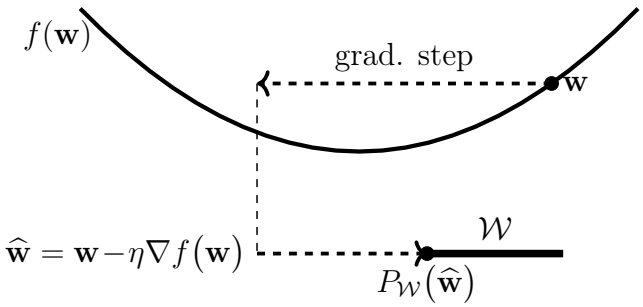


Figure 8: GD proyectado amplía un paso de gradiente básico con una proyección de regreso al conjunto de restricciones $\mathcal{W}$.

descenso por subgradiente El descenso por subgradiente es una generalización del GD que no requiere la diferenciabilidad de la función a minimizar. Esta generalización se obtiene al reemplazar el concepto de gradiente por el de subgradiente. Similar a los gradientes, los subgradientes permiten construir aproximaciones locales de una función objetivo. La función objetivo podría ser el riesgo empírico $\widehat{L}\big(h^{(\mathbf{w})} \big| \mathcal{D}\big)$ visto como una función de los parámetros del modelo $\mathbf{w}$ que seleccionan una hipótesis $h^{(\mathbf{w})} \in \mathcal{H}$.

descomposición en valores propios (EVD) La descomposición en valores propios para una matriz cuadrada $\mathbf{A} \in \mathbb{R}^{d \times d}$ es una factorización de la forma

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

Las columnas de la matriz $\mathbf{V} = \big(\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(d)}\big)$ son los vector propios de la matriz $\mathbf{A}$. La matriz diagonal $\mathbf{\Lambda} = \operatorname{diag}\big\{\lambda_1, \ldots, \lambda_d\big\}$ contiene los valor propios λ_j correspondientes a los vector propios $\mathbf{v}^{(j)}$. Es

importante notar que esta descomposición solo existe si la matriz $\mathbf{A}$ es diagonalizable.

descomposición en valores singulares (SVD) La descomposición en valores singulares (SVD) para una matriz $\mathbf{A} \in \mathbb{R}^{m \times d}$ es una factorización de la forma

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

con matrices ortonormales $\mathbf{V} \in \mathbb{R}^{m \times m}$ y $\mathbf{U} \in \mathbb{R}^{d \times d}$ [3]. La matriz $\mathbf{\Lambda} \in \mathbb{R}^{m \times d}$ es no nula solo en la diagonal principal, cuyos elementos $\Lambda_{j,j}$ son no negativos y se denominan valores singulares.

diagrama de dispersión Un método de visualización que representa punto de datos mediante marcadores en un plano bidimensional. La Figura 9 depicta un ejemplo de un diagrama de dispersión.

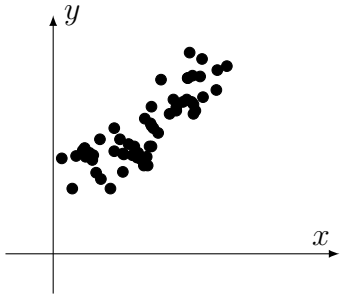


Figure 9: Un diagrama de dispersión de algunos punto de datos que representan las condiciones climáticas diarias en Finlandia. Cada punto de datos se caracteriza por su temperatura mínimo diaria x como atributo y su temperatura máximo diaria y como etiqueta. Las temperaturas se han medido en la estación meteorológica FMI de Helsinki Kaisaniemi durante el período 1.9.2024 - 28.10.2024.

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 espacio de hipótesis infinito es una medida ampliamente utilizada para su tamaño. Nos referimos a la literatura (vea [39]) 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 espacio de hipótesis infinito $\mathcal{H}$ es una medida de su tamaño. A grandes rasgos,la dimensión efectiva es igual al número efectivo de parámetros del modelo independientes y ajustables. Estos parámetros pueden ser los coeficientes utilizados en un mapa lineal o los pesos 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 modelo locals en los nodos i, i' conectados por una arista en el red de aprendizaje federado (red FL).

dispositivo Cualquier sistema físico que pueda usarse para almacenar y procesar datos. En el contexto de ML, generalmente nos referimos a un ordenador que puede leer punto de datoss de diferentes fuentes y, a su vez, entrenar un modelo de ML utilizando estos punto de datoss.

distribución de probabilidad Para analizar métodos de ML, puede ser útil interpretar los punto de datoss como i.i.d. realizacións de una RV. Las propiedades típicas de tales punto de datoss están gobernadas por la distribución de probabilidad de esta RV. La distribución de probabilidad de una RV binaria $y \in \{0,1\}$ se especifica completamente mediante las probabilidades $p(y=0)$ y $p(y=1)=1-p(y=0)$. La distribución de probabilidad de una RV con valores reales $x \in \mathbb{R}$ puede especificarse mediante una función de densidad de probabilidad (pdf) $p(x)$ tal que $p(x \in [a,b]) \approx p(a)|b-a|$. En el caso más general, una distribución de probabilidad se define mediante una medida de probabilidad [40, 41].

distribución normal multivariante La distribución normal multivariante $\mathcal{N}(\mathbf{m}, \mathbf{C})$ es una familia importante de distribución de probabilidads para una RV continua $\mathbf{x} \in \mathbb{R}^d$ [5, 40, 42]. Esta familia está parametrizada por la media $\mathbf{m}$ y la matriz de covarianza $\mathbf{C}$ de $\mathbf{x}$. Si la matriz de covarianza es invertible, la distribución de probabilidad de $\mathbf{x}$ es

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

divergencia de Kullback-Leibler (divergencia KL) La divergencia KL es una medida cuantitativa de cuánto se diferencia una distribución de probabilidad de otra distribución de probabilidad [43].

embudo de privacidad El embudo de privacidad es un método para aprender atributos amigables con la privacidad de los punto de datoss [44].

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

envenenamiento de datos El envenenamiento de datos se refiere a la manipulación intencional (o fabricación) de punto de datos para influir en el entrenamiento de un modelo de ML [45, 46]. La protección contra el envenenamiento de datos es especialmente importante en aplicaciones distribuidas de ML donde los conjunto de datoss están descentralizados.

error cuadrático medio de estimación (MSEE) Consideremos un método de ML que aprende parámetros del modelo $\widehat{\mathbf{w}}$ 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 parámetros del modelo de la distribución de probabilidad de $\mathbf{z}$. El error cuadrático medio de estimación se define como la esperanza $\mathbb{E}\{\|\Delta\mathbf{w}\|^2\}$ del cuadrado de la norma euclidiana del error de estimación [47, 48].

error de entrenamiento El pérdida 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 pérdida promedio mínimo que se logra mediante una solución de ERM.

error de estimación Consideremos punto de datoss, cada uno con un vector de atributos $\mathbf{x}$ y una etiqueta y. En algunas aplicaciones, podemos modelar la relación entre el vector de atributos 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 atributos dado. En un espacio de hipótesis paramétrico, donde las hipótesisse determinan mediante parámetros del modelo $\mathbf{w}$, podemos definir el error de estimación como $\Delta\mathbf{w} = \widehat{\mathbf{w}} - \overline{\mathbf{w}}$ [48, 49].

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 pérdida promedio de $\hat{h}$ en un conjunto de validación, que es diferente del conjunto de entrenamiento, se denomina error de validación.

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

espacio de etiquetas Consideremos una aplicación de ML que involucra punto de datoss caracterizados por atributos 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"$\}$.

espacio de Hilbert Un espacio de Hilbert es un espacio vectorial lineal

equipado con un producto interno entre pares de vectores. Un ejemplo importante de espacio de Hilbert es el espacio euclidiano $\mathbb{R}^d$, para alguna dimensión d, que consiste en vectores euclidianos $\mathbf{u} = \big(u_1, \ldots, u_d\big)^T$ junto con el producto interno $\mathbf{u}^T\mathbf{v}$.

espacio de hipótesis Cada método práctico deML utiliza un espacio de hipótesis (o modelo) $\mathcal{H}$. El espacio de hipótesis de un método de ML es un subconjunto de todos los posibles mapeos del espacio de atributos al espacio de etiquetas. La elección del diseño del espacio de hipótesis debe considerar los recursos computacionales disponibles y los aspectos estadísticos. Si la infraestructura computacional permite operaciones matriciales eficientes y existe una relación (aproximadamente) lineal entre un conjunto de atributos y una etiqueta, una elección útil para el espacio de hipótesis podria ser el modelo lineal.

espacio de parámetros El espacio de parámetros $\mathcal{W}$ de un modelo de ML $\mathcal{H}$ es el conjunto de todas las elecciones factibles para los parámetros del modelo (vea la Figura 10). Muchos métodos de ML importantes usan un modelo que está parametrizado por vectores del espacio euclidiano $\mathbb{R}^d$. Dos ejemplos ampliamente utilizados de modelos parametrizados son los modelo lineals y los red profundas. El espacio de parámetros es entonces a menudo un subconjunto $\mathcal{W} \subseteq \mathbb{R}^d$, por ejemplo, todos los vectores $\mathbf{w} \in \mathbb{R}^d$ con una norma menor a uno.

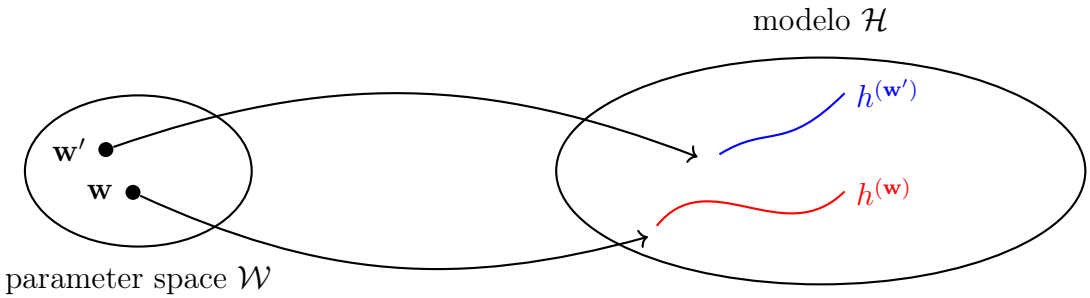


Figure 10: El espacio de parámetros $\mathcal{W}$ de un modelo deML $\mathcal{H}$ consiste en todas las elecciones factibles para los parámetros del modelo. Cada elección $\mathbf{w}$ para los parámetros del modelo seleccionan un mapa de hipótesis $h^{(\mathbf{w})} \in \mathcal{H}$.

espacio euclidiano El espacio euclidiano $\mathbb{R}^d$ de dimensión $d \in \mathbb{N}$ consiste en vectores $\mathbf{x} = \big(x_1, \ldots, x_d\big)$, con d entradas de valores reales $x_1, \ldots, x_d \in \mathbb{R}$. Dicho espacio euclidiano está equipado con una estructura geométrica definida por el producto interno $\mathbf{x}^T\mathbf{x}' = \sum_{j=1}^{d} x_j x'_j$ entre dos vectores cualesquiera $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$ [2].

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) [50]. La Figure 11 muestra una señal temporal junto con su espectrograma.

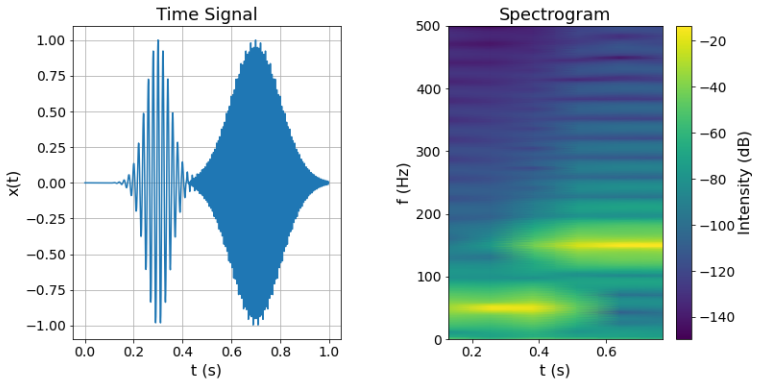


Figure 11: Izquierda: una señal temporal compuesta por dos pulsos gaussianos modulados. Derecha: representación de intensidad de su espectrograma.

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 [51]. Conviene señalar que, además del espectrograma, existen otras representaciones alternativas para describir la distribución tiempo-frecuencia de la energía de una señal [52, 53].

esperanza Consideremos un vector de atributos numérico $\mathbf{x} \in \mathbb{R}^d$ que interpretamos como la realización de una RV con una distribución de probabilidad $p(\mathbf{x})$. La esperanza de $\mathbf{x}$ se define como la integral $\mathbb{E}\{\mathbf{x}\} := \int \mathbf{x} p(\mathbf{x})$ [2, 41, 54]. Nótese que la esperanza solo está definida si esta integral existe, es decir, si la RV es integrable.

esperanza-maximización (EM) Considera un modelo probabilístico $p(\mathbf{z}; \mathbf{w})$ para los punto de datoss $\mathcal{D}$ generados en alguna aplicación de ML. El

estimador de máxima verosimilitudpara los parámetros del modelo $\mathbf{w}$ se obtienen maximizando $p(\mathcal{D}; \mathbf{w})$. Sin embargo, el problema de optimización resultante puede ser computacionalmente desafiante. El algoritmo de expectativa-maximización (EM) aproxima el estimador de máxima verosimilitud introduciendo una RV latente $\mathbf{z}$ de modo que maximizar $p(\mathcal{D}, \mathbf{z}; \mathbf{w})$ sea más fácil [49, 55, 56]. Dado que no observamos $\mathbf{z}$, necesitamos estimarlo a partir del conjunto de datos observado $\mathcal{D}$ utilizando una esperanza condicional. La estimación resultante $\widehat{\mathbf{z}}$ se utiliza para calcular una nueva estimación $\widehat{\mathbf{w}}$ resolviendo $\max_{\mathbf{w}} p(\mathcal{D}, \widehat{\mathbf{z}}; \mathbf{w})$. El punto clave es que la esperanza condicional $\widehat{\mathbf{z}}$ depende de los parámetros del modelo $\widehat{\mathbf{w}}$, que hemos actualizado en función de $\widehat{\mathbf{z}}$. Por lo tanto, debemos volver a calcular $\widehat{\mathbf{z}}$, lo que a su vez resulta en una nueva elección $\widehat{\mathbf{w}}$ para los parámetros del modelo. En la práctica, repetimos el cálculo de la esperanza condicional (es decir, el E-step) y la actualización de los parámetros del modelo (es decir, the M-step) hasta que se cumpla algún criterio de parada.

estimador de Bayes Considera un modelo probabilístico con una distribución de probabilidad conjunta $p(\mathbf{x}, y)$ para los atributos $\mathbf{x}$ y la etiqueta y de un punto de datos. Para una función de pérdida dada $L(\cdot, \cdot)$, denominamos a una hipótesis h como un estimador de Bayes si su riesgo $\mathbb{E}\{L((\mathbf{x}, y), h)\}$ es el mínimo [47]. Nótese que la propiedad de una hipótesis de ser un estimador de Bayes depende de la distribución de probabilidad subyacente y de la elección de la función de pérdida $L(\cdot, \cdot)$.

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" [25], [26], [27].

experto El ML tiene como objetivo aprender una hipótesis h que prediga con precisión la etiqueta de un punto de datos basado en sus atributos. Medimos el error de predicción utilizando una función de pérdida. Idealmente, buscamos una hipótesis que incurra en la pérdida mínima para cualquier punto de datos. Podemos hacer este objetivo más preciso mediante el suposición de independencia e idéntica distribución (suposición i.i.d.) y utilizando el riesgo de Bayes como referencia referencia (baseline) para la pérdida 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 [18]. Los métodos de minimización de Arrepentimiento (regret) aprenden una hipótesis que incurre en una pérdida comparable a la del mejor experto [17, 18].

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

modelo de ML entrenado mediante la entropía condicional (diferencial) de sus predicciónes, dadas las predicciónes del usuario [59, 60].

Explicaciones Locales Interpretables e Independientes del Modelo (LIME) Consideremos un modelo entrenado (o una hipótesis aprendida) $\widehat{h} \in \mathcal{H}$, que asigna la vector de atributos 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 atributos $\mathbf{x}^{(0)}$ [61]. 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 atributos $\mathbf{x}$ cercana a $\mathbf{x}^{(0)}$ y la (pseudo-)etiqueta $\widehat{h}(\mathbf{x})$. Nótese que podemos utilizar un modelo $\mathcal{H}'$ diferente para la aproximación que el modelo original $\mathcal{H}$. Por ejemplo, podemos usar un árbol de decisión para aproximar (localmente) una red profunda. Otra elección ampliamente utilizada para $\mathcal{H}'$ es el modelo lineal.

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 atributos individuales de un punto de datos [62]. También podemos usar formas visuales, como los mapas de intensidad usados en tareas de clasificación de imágenes [63].

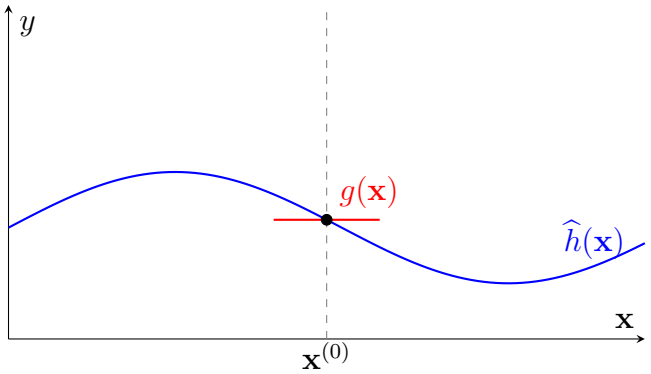


Figure 12: Para explicar un modelo $\widehat{h} \in \mathcal{H}$ entrenado, alrededor de un vector de atributos $\mathbf{x}^{(0)}$, podemos usar una aproximación local $g \in \mathcal{H}'$.

familias exponenciales en red (nExpFam) Una colección de familias exponenciales, cada una de ellas asignada a un nodo de un red de aprendizaje federado (red FL). Los parámetros del modelo están acoplados a través de la estructura de la red al requerir que tengan una pequeña GTV [64].

FedProx FedProx se refiere a un algoritmo iterativo de FL que alterna entre entrenar modelo locals por separado y combinar los parámetros del modelo locales actualizados. A diferencia de FedAvg, que utiliza descenso de gradiente estocástico (SGD) para entrenar los modelo locals, FedProx usa un operador proximal para el entrenamiento [65].

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

atributo 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 atributo 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 [66]).

frontera de decisión Consideremos un mapa de hipótesis h que recibe un vector de atributo $\mathbf{x} \in \mathbb{R}^d$ y entrega un valor de un conjunto finito $\mathcal{Y}$. La frontera de decisión de h es el conjunto de vectores $\mathbf{x} \in \mathbb{R}^d$ que se encuentran entre diferentes región de decisións.Más precisamente, un vector $\mathbf{x}$ pertenece a la frontera de decisión si, y solo si, cada entorno $\{\mathbf{x}' : \|\mathbf{x}-\mathbf{x}'\| \leq \varepsilon\}$, para cualquier $\varepsilon > 0$, contiene al menos dos vectores con diferentes valores de función.

fuertemente convexa Una función real diferenciable de valor continuo $f(\mathbf{x})$es fuertemente convexo con coeficiente σ si cumple: $f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x}) + (\sigma/2)\|\mathbf{y}-\mathbf{x}\|_2^2$ [67], [68, 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 pesos $w_1, \ldots, w_d$.

función de densidad de probabilidad (pdf) La probabilidad densidad de probabilidad $p(x)$ de una RV con valores reales $x \in \mathbb{R}$ es una representación particular de su distribución de probabilidad. Si la función de densidad de probabilidad existe, se puede usar para calcular la probabilidad de que x tome un valor de un conjunto (medible) $\mathcal{B} \subseteq \mathbb{R}$ mediante $p(x \in \mathcal{B}) = \int_{\mathcal{B}} p(x')dx'$ [5, Ch. 3]. La función de densidad de probabilidad de una RV vectorial $\mathbf{x} \in \mathbb{R}^d$ (si existe) permite calcular la probabilidad de que $\mathbf{x}$ pertenezca a una región (medible) $\mathcal{R}$ mediante $p(\mathbf{x} \in \mathcal{R}) = \int_{\mathcal{R}} p(\mathbf{x}')dx'_1 \ldots dx'_d$ [5, Ch. 3].

función de pérdida Una función de pérdida es una aplicación

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

Asigna un número real no negativo (es decir, la pérdida) $L\left((\mathbf{x}, y), h\right)$ a un par que consiste en un punto de datos, con atributos $\mathbf{x}$ y una etiqueta y, y una hipótesis $h \in \mathcal{H}$. El valor $L\left((\mathbf{x}, y), h\right)$ cuantifica la discrepancia entre la etiqueta verdadera y y la predicción $h(\mathbf{x})$. Valores bajos (cercanos a cero) de $L\left((\mathbf{x}, y), h\right)$ indican una discrepancia menor entre la predicción $h(\mathbf{x})$ y la etiqueta y. La Figura 13 muestra una función de pérdida para un punto de datos dado, con atributos $\mathbf{x}$ y etiqueta y, como función de la hipótesis $h \in \mathcal{H}$.

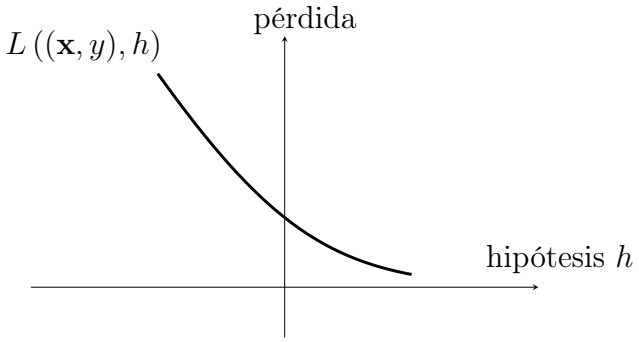


Figure 13: Alguna funcion de pérdida $L\left((\mathbf{x}, y), h\right)$ para un punto de datos fijado, con vector de atributos $\mathbf{x}$ y etiqueta y, y una hipótesis variable h. Los métodos de ML intentan encontrar (o aprender) una hipótesis que incurra en una pérdida mínima.

función objetivo Una función objetivo es un mapa que asigna a cada valor de una variable de optimización, como los parámetros del modelo $\mathbf{w}$ 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 $h^{(\mathbf{w})}$.

generalización Muchos de los sistemas actuales de ML (y AI) se basan en ERM: En esencia, entrenan un modelo (es decir, aprenden una hipótesis $\hat{h} \in \mathcal{H}$) minimizando la pérdida promedio (o riesgo empírico) en algunos punto de datoss $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}$, que sirven como un conjunto de entrenamiento $\mathcal{D}^{(\text{train})}$. La generalización se refiere a la capacidad de un método de ML para desempeñarse bien fuera del conjunto de entrenamiento. Cualquier teoría matemática de la generalización necesita algún concepto matemático para el "fuera del conjunto de entrenamiento." Por ejemplo, la teoría del aprendizaje estadístico utiliza un

modelo probabilístico como la suposición i.i.d. para la generacion de datos: los punto de datoss en el conjunto de entrenamiento son i.i.d. realizaciónes de alguna distribución de probabilidad subyacente $p(\mathbf{z})$. Un modelo probabilístico nos permite explorar el exterior del conjunto de entrenamiento generando adicionales i.i.d. realizacións de $p(\mathbf{z})$. Además, el uso de la suposición i.i.d. nos permite definir el riesgo de un modelo entrenado $\hat{h} \in \mathcal{H}$ como la esperanza de la pérdida $\bar{L}\big(\hat{h}\big)$. Además, podemos utilizar límites de concentración o resultados de convergencia para secuencias de i.i.d. RVs para limitar la desviación entre el riesgo empírico $\widehat{L}\big(\hat{h}|\mathcal{D}^{(\text{train})}\big)$ de un modelo entrenado y su riesgo [39]. También es posible estudiar la generalización sin utilizar modelo probabilísticos. Por ejemplo, podríamos utilizar perturbaciones (determinísticas) de los punto de datos en el conjunto de entrenamiento para estudiar su exterior. En general, deseamos que el modelo entrenado sea robusto, es decir, que sus predicciónes no cambien demasiado ante pequeñas perturbaciones de un punto de datos. Considere un modelo entrenado para detectar un objeto en una imagen capturada por un teléfono inteligente. El resultado de la detección no debería cambiar si enmascaramos un pequeño número de píxeles seleccionados aleatoriamente en la imagen [69].

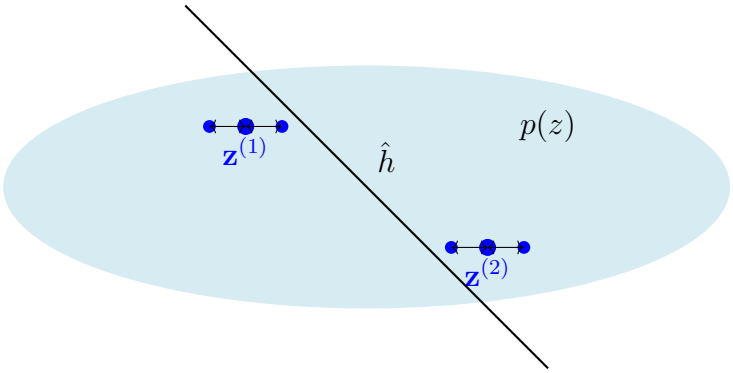


Figure 14: Dos punto de datos $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}$ que se utilizan como un conjunto de entrenamiento para aprender una hipótesis $\hat{h}$ mediante ERM. Podemos evaluar $\hat{h}$ fuera de $\mathcal{D}^{(\text{train})}$ ya sea mediante una suposición i.i.d. con alguna distribución de probabilidad subyacente $p(\mathbf{z})$ o mediante la perturbación de los punto de datos.

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 grafo 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 clúster [6, Ch. 8]. Este grado puede interpretarse como una asignación blanda (*soft*) al clúster.Los métodos de Agrupamiento suave pueden codificar el grado de pertenencia mediante un número real en el intervalo $[0, 1]$. El Agrupamiento rígido 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 [70]. 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 pesos A_e a cada arista $e \in \mathcal{E}$.

grafo conexo Un grafo 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 $\mathcal{V} \setminus \mathcal{V}'$.

grafo de similitud Algunas aplicaciones de ML generan punto de datoss que están relacionados por una noción de similitud específica del dominio.

Estas similitudes pueden ser representadas convenientemente utilizando un grafo de similitud $\mathcal{G} = \big(\mathcal{V} := \{1, \ldots, m\}, \mathcal{E}\big)$. El nodo $r \in \mathcal{V}$ representa el r-ésimo punto de datos. Dos nodos están conectados por una arista no dirigida si los punto de datos correspondientes son similares.

hipótesis Una hipótesis se refiere a un mapa (o función) $h : \mathcal{X} \to \mathcal{Y}$ que va del espacio de atributos $\mathcal{X}$ al espacio de etiquetas $\mathcal{Y}$. Dado un punto de datos con atributos $\mathbf{x}$, 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 atributos $\mathbf{x}$ 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.

independientes e idénticamente distribuidos (i.i.d.) Es útil interpretar los punto de datoss $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}$ como realizaciónes de RVs i.i.d., es decir, independientes e idénticamente distribuidos, con una distribución de probabilidad común. Si estos RVs son de valor continuo, su pdf conjunta se expresa como $p\big(\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}\big) = \prod_{r=1}^{m} p\big(\mathbf{z}^{(r)}\big)$, donde $p(\mathbf{z})$

es la pdf marginal común de las RVs subyacentes.

información mutua (MI) La información mutua $I(\mathbf{x}; y)$ entre dos RVs $\mathbf{x}$, y definidas en el mismo probability space está dada por [43]

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

Es una medida de qué tan bien podemos estimar y basándonos únicamente en $\mathbf{x}$. Un valor grande de $I(\mathbf{x}; y)$ indica que y puede predecirse bien únicamente a partir de $\mathbf{x}$. Esta predicción podría obtenerse mediante una hipótesis aprendida por un método de ML basado en ERM.

Instituto Meteorológico de Finlandia (FMI) El FMI es una agencia gubernamental responsable de recopilar y reportar datos meteorológicos en Finlandia.

inteligencia artificial (IA) La IA se refiere a sistemas que se comportan de manera racional en el sentido de maximizar una recompensa a largo plazo. El enfoque basado en ML para la IA consiste en entrenar un modelo para predecir acciones óptimas. Estas predicciones se calculan a partir de observaciones sobre el estado del entorno. La elección de la función de pérdida distingue las aplicaciones de IA de las aplicaciones de ML más básicas. Los sistemas de IA rara vez tienen acceso a un conjunto de entrenamiento etiquetado que permita medir la pérdida promedio para cualquier posible elección de parámetros del modelo. En su lugar, los sistemas de IA utilizan señales de recompensa observadas para obtener una estimación puntual de la pérdida incurrida por la elección actual de parámetros del modelo.

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 [71]. La Unión Europea ha propuesto siete requisitos clave (KRs) para una AI confiable (que típicamente se basa en métodos de ML) [72]:

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 [59].

kernel Considere punto de datoss caracterizados por un vector de atributos $\mathbf{x} \in \mathcal{X}$ con un espacio de atributos genérico $\mathcal{X}$. Un kernel (de valor real) $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ asigna a cada par de vector de atributoss $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ un número real $K\big(\mathbf{x}, \mathbf{x}'\big)$. El valor $K\big(\mathbf{x}, \mathbf{x}'\big)$ se interpreta a menudo como una medida de similitud entre $\mathbf{x}$ y $\mathbf{x}'$. Los Método de kernels utilizan un kernel para transformar el vector de atributos $\mathbf{x}$ en un nuevo vector

de atributos $\mathbf{z} = K(\mathbf{x}, \cdot)$. Este nuevo vector de atributos pertenece a un espacio de atributos lineal $\mathcal{X}'$ que (en general) es diferente del espacio de atributos original $\mathcal{X}$. El espacio de atributos $\mathcal{X}'$ tiene una estructura matemática específica, es decir, es un espacio de Hilbert de kernel reproducible espacio de Hilbert [73, 74].

ley de los grandes números La ley de los grandes números se refiere a la convergencia del promedio de un número creciente (grande) de i.i.d. RVs hacia la media de su distribución de probabilidad común. Diferentes instancias de la ley de los grandes números se obtienen utilizando distintas nociones de convergencia [75].

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 parámetros del modelo.

mapa de atributos Un mapa de atributos se refiere a una transformación que convierte los atributos originales de un punto de datos en nuevos atributos. Estos nuevos atributos pueden ser preferibles a los originales por diversas razones. Por ejemplo, la disposición de los punto de datos puede volverse más simple (o más lineal) en el nuevo espacio de atributos, permitiendo el uso de modelo lineals en los nuevos atributos. Esta idea es uno de los principales impulsores del desarrollo de método de kernels [74]. Además, las capas ocultas de una red profunda pueden interpretarse como un mapa de atributos entrenable seguido de un modelo lineal

en la forma de la capa de salida. Otra razón para aprender un mapa de atributos podría ser que aprender un pequeño número de nuevos atributos ayuda a evitar el sobreajuste y garantiza la interpretabilidad [61]. El caso especial de un mapa de atributos que proporciona dos atributo numéricos es particularmente útil para la visualización de datos. De hecho, podemos representar punto de datos en un diagrama de dispersión usando dos atributos como las coordenadas de un punto de datos.

matriz de atributos Considere un conjunto de datos $\mathcal{D}$ con m punto de datoss caracterizados por vector de atributoss $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)} \in \mathbb{R}^d$. Es conveniente agrupar los vector de atributoss individuales en una matriz de atributos matrix $\mathbf{X} := \big(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}\big)^T$ de tamaño $m \times d$.

matriz de confusión Considere punto de datoss caracterizados por atributos $\mathbf{x}$ y etiqueta y con valores del espacio de etiquetas finito $\mathcal{Y} = \{1, \ldots, k\}$. La matriz de confusión es una matriz de tamaño $k \times k$ donde las filas representan diferentes valores c de la verdadera etiqueta de un punto de datos. Las columnas de la matriz de confusión corresponden a diferentes valores c' entregados por una hipótesis $h(\mathbf{x})$. El elemento (c, c') de la matriz de confusión es la fracción de punto de datoss con la etiqueta $y = c$ y la predicción $\hat{y} = c'$ asignada por la hipótesis h.

matriz de covarianza La matriz de covarianza de una RV $\mathbf{x} \in \mathbb{R}^d$ se define como $\mathbb{E}\Big\{\big(\mathbf{x} - \mathbb{E}\{\mathbf{x}\}\big)\big(\mathbf{x} - \mathbb{E}\{\mathbf{x}\}\big)^T\Big\}$.

matriz de covarianza muestral La matriz de matriz de covarianza mues-

tral $\widehat{\mathbf{\Sigma}} \in \mathbb{R}^{d \times d}$ para un conjunto dado vector de atributoss $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)} \in \mathbb{R}^d$ se define como

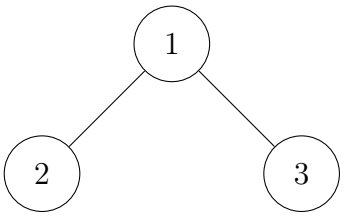
$$\widehat{\mathbf{\Sigma}} = (1/m) \sum_{r=1}^{m} (\mathbf{x}^{(r)} - \widehat{\mathbf{m}})(\mathbf{x}^{(r)} - \widehat{\mathbf{m}})^T.$$

Aqui, usamos la media muestral $\widehat{\mathbf{m}}$.

matriz laplaciana La estructura de un grafo $\mathcal{G}$, con nodos $i = 1, \ldots, n$, se puede analizar utilizando las propiedades de matrices especiales asociadas con $\mathcal{G}$. Una de estas matrices es la matriz laplaciana del grafo $\mathbf{L}^{(\mathcal{G})} \in \mathbb{R}^{n \times n}$, la cual se define para un grafo no dirigido y ponderado [10, 76]. Se define de forma elemento por elemento como (Vea la Figura 15)

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

Aqui, $A_{i,i'}$ denota el peso de arista de una arista $\{i, i'\} \in \mathcal{E}$.



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

Figure 15: Izquierda: Un grafo no dirigido $\mathcal{G}$ con tres nodos $i = 1, 2, 3$. Derecha: La matriz laplaciana $\mathbf{L}^{(\mathcal{G})} \in \mathbb{R}^{3 \times 3}$ of $\mathcal{G}$.

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

media muestral La media muestra $\mathbf{m} \in \mathbb{R}^d$ para un conjunto de datos dado, con vector de atributoss $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)} \in \mathbb{R}^d$, se define como

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

minimización de la pérdida regularizada (RLM) Consulte minimización del riesgo empírico regularizado (RERM).

minimización de variación total generalizada (GTVMin) La minimización de variación total generalizada (GTVMin) es una instancia de RERM que utiliza la GTV de los parámetros del modelo locales como un regularizador [77].

minimización del riesgo empírico explicable (EERM) La minimización del riesgo empírico explicable (EERM) es una instancia de SRM que agrega un término de regularización a la pérdida promedio en la función objetivo de ERM. El término de regularización se elige para favorecer mapas de hipótesis que sean intrínsecamente explicables para un usuario específico. Este usuario se caracteriza por sus predicciónes proporcionadas para los punto de datoss en un conjunto de entrenamiento [58].

minimización del riesgo empírico regularizado (RERM) El ERM básico aprende una hipótesis (o entrena un modelo) $h \in \mathcal{H}$ basado únicamente en el riesgo empírico $\widehat{L}\big(h|\mathcal{D}\big)$ incurrido en un conjunto de entrenamiento $\mathcal{D}$. Para hacer que ERM sea menos propenso al sobreajuste, podemos implementar la regularización incluyendo un regularizador (escalado) $\mathcal{R}\big\{h\big\}$ en el objetivo de aprendizaje. Esto da lugar a la minimización

del riesgo empírico regularizado (RERM),

$$\hat{h} \in \operatorname*{argmin}_{h \in \mathcal{H}} \widehat{L}\big(h|\mathcal{D}\big) + \alpha \mathcal{R}\big\{h\big\}. \tag{5}$$

El parámetro $\alpha \geq 0$ controla la intensidad de la regularización. Para $\alpha = 0$, recuperamos el ERM estándar sin regularización. A medida que α aumenta, la hipótesis aprendida se inclina cada vez más hacia valores pequeños de $\mathcal{R}\big\{h\big\}$. El componente $\alpha\mathcal{R}\big\{h\big\}$ en la función objetivo de (5) se puede entender intuitivamente como un sustituto para el aumento promedio en la pérdida que puede ocurrir al predecir etiquetas para punto de datoss fuera del conjunto de entrenamiento. Esta intuición se puede precisar en varias maneras. Por ejemplo, considere un modelo lineal entrenado usando pérdida de error cuadrático y el regularizador $\mathcal{R}\big\{h\big\} = \|\mathbf{w}\|_2^2$. En este caso, $\alpha\mathcal{R}\big\{h\big\}$ corresponde al aumento esperado en la pérdida causado por la adición de VA gaussianas a los vector de atributoss en el conjunto de entrenamiento [6, Ch. 3]. Una construcción basada en principios para el regularizador $\mathcal{R}\big\{h\big\}$ surge de límites superiores aproximados en el error de generalización. La instancia resultante de RERM se conoce como SRM [78, Sec. 7.2].

minimización del riesgo estructural (SRM) La minimización del riesgo estructural (SRM) es una forma de RERM, en la que el modelo $\mathcal{H}$ se puede expresar como una unión contable de submodelos: $\mathcal{H} = \bigcup_{n=1}^{\infty} \mathcal{H}^{(n)}$. Cada submodelo $\mathcal{H}^{(n)}$ permite la derivación de un límite superior aproximado para el error de generalización que se incurre al aplicar ERM para entrenar $\mathcal{H}^{(n)}$. Estos límites superiores individuales para cada submodelo—se combinan para formar un regularizador utilizado en el

objetivo de RERM. Estos límites superiores aproximados (uno para cada $\mathcal{H}^{(n)}$) se combinan para construir un regularizador para RERM [39, Sec. 7.2].

minimización empírica del riesgo (ERM) La minimización del Riesgo empírico es el problema de optimización que consiste en encontrar una hipótesis (dentro de un modelo) con la mínimo pérdida 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 modelo, y la pérdida [6, Ch. 3].

model selection En ML, la selección de modelo se refiere al proceso de elegir entre diferentes modelos candidatos. En su forma más básica, la selección de modelo consiste en: 1) entrenar cada modelo candidato; 2) calcular el error de validación para cada modelo entrenado; y 3) elegir el modelo con el menor error de validación [6, Ch. 6].

modelo En el contexto de métodos de ML, el término modelo típicamente se refiere a el espacio de hipótesis empleado por un método de ML [6, 39].

modelo de lenguaje de gran escala (LLM) Los modelos de lenguaje de gran escala son un término genérico para métodos de ML que procesan y generan texto similar al humano. Estos métodos suelen usar red profundas con miles de millones (o incluso billones) de parámetros. Una elección ampliamente utilizada para la arquitectura de red se conoce como Transformers [79]. El entrenamiento de modelos de lenguaje de gran escala a menudo se basa en la tarea de predecir algunas palabras

que se eliminan intencionadamente de un corpus de texto extenso. Así, podemos construir punto de dato etiquetados simplemente seleccionando algunas palabras de un texto como etiquetas y las palabras restantes como atributos de punto de datoss. Esta construcción requiere muy poca supervisión humana y permite generar conjunto de entrenamientos suficientemente grandes para modelos de lenguaje de gran escala.

modelo de mezcla gaussiana (GMM) Un GMM es un tipo particular de modelo probabilístico para un vector numérico $\mathbf{x}$ (por ejemplo, las atributos de un punto de datos). Dentro de un GMM, el vector $\mathbf{x}$ se extrae de una distribución normal multivariante $p^{(c)} = \mathcal{N}\left(\boldsymbol{\mu}^{(c)}, \mathbf{C}^{(c)}\right)$ seleccionada aleatoriamente, con $c = I$. El índice $I \in \{1, \ldots, k\}$ es una RV con probabilidades $p(I = c) = p_c$. Ten en cuenta que un GMM se parametriza por la probabilidad p_c, el vector media $\boldsymbol{\mu}^{(c)}$, y la matriz de covarianza $\boldsymbol{\Sigma}^{(\mathbf{c})}$ para cada $c = 1, \ldots, k$. Los GMM se utilizan ampliamente para agrupamiento, estimación de densidad y como un modelo generativo.

modelo en red Un modelo en red sobre un red de aprendizaje federado (red FL) $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ asigna un modelo local (es decir, un espacio de hipótesis) 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 modelo generativo probabilístico para un grafo no dirigido $\mathcal{G} = \left(\mathcal{V}, \mathcal{E}\right)$ con conjunto de nodos $\mathcal{V}$ [80]. En su forma básica, asigna aleatoriamente cada nodo $i \in \mathcal{V}$ a una clúster $c_i \in \{1, \ldots, k\}$. 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 atributos numérica $\mathbf{x} \in \mathbb{R}^d$. Un modelo lineal es un espacio de hipótesis 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 espacio de hipótesiss, parametrizada por el número d de atributos 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 modelo lineal que utiliza pocas atributos cuidadosamente elegidas suele considerarse más interpretable [21, 61].

modelo local Considera una colección de conjunto de datos locals que están asignados a los nodos de un red de aprendizaje federado (red FL). Un modelo local $\mathcal{H}^{(i)}$ es un espacio de hipótesis asignado a un nodo $i \in \mathcal{V}$. Diferentes nodos podrían tener asignados diferentes espacio de hipótesiss, es decir, en general $\mathcal{H}^{(i)} \neq \mathcal{H}^{(i')}$ para diferentes nodos $i, i' \in \mathcal{V}$.

modelo probabilístico Un modelo probabilístico interpreta los punto de datoss como realizaciónes de RVs con una distribución de probabilidad conjunta. Esta distribución de probabilidad conjunta típicamente

incluye parámetros que deben seleccionarse manualmente or o aprenderse usando métodos de inferencia estadística como la estimación por máxima verosimilitud [47].

muestra Una secuencia finita (o lista) de punto de datoss $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}$ que se obtiene o interpreta como la realización de m i.i.d. RVs con una distribución de probabilidad común $p(\mathbf{z})$. La longitud m de la secuencia se denomina tamaño de la muestra.

máquina de vectores de soporte (SVM) La SVM es un método de clasificación binaria que aprende un mapa de hipótesis lineal. Por lo tanto, al igual que la regresión lineal y la regresión logística, también es una instancia de ERM para el modelo lineal. Sin embargo, la SVM utiliza una función de pérdida diferente a la empleada en esos métodos. Como se ilustra en la Figura 16, su objetivo es separar al máximo los punto de datoss de las dos clases diferentes en el espacio de atributos (es decir, el principio de margen máximo). Maximizar esta separación es equivalente a minimizar una variante regularizada de la pérdida de hinge (10) [55, 73, 81].

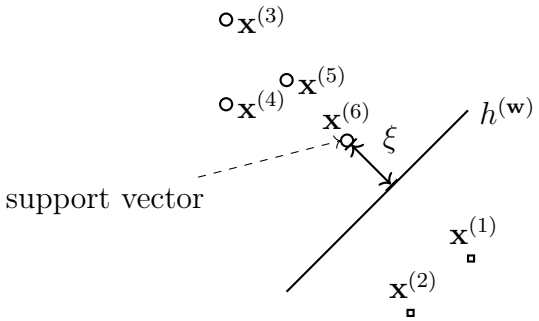


Figure 16: La SVM aprende una hipótesis (o clasificador) $h^{(\mathbf{w})}$ con pérdida de margen suave promedio mínima pérdida de hinge. Minimizar esta pérdida es equivalente a maximizar el margen ξ entre la frontera de decisión de $h^{(\mathbf{w})}$ y cada clase del conjunto de entrenamiento.

La variante básica de la SVM solo es útil si los punto de datoss de diferentes categorías pueden ser (aproximadamente) separables linealmente. Para una aplicación de ML donde las categorías no son separables linealmente basadas en las atributos originales (crudas), es posible aplicar la SVM a atributos transformadas. Estas atributos transformadas se pueden obtener aplicando un mapa de atributos derivado de un kernel.

máxima verosimilitud Considera punto de datoss $\mathcal{D} = \{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}\}$ que se interpretan como las realizacións de i.i.d. RVs con una distribución de probabilidad común $p(\mathbf{z}; \mathbf{w})$ que depende de los parámetros del modelo $\mathbf{w} \in \mathcal{W} \subseteq \mathbb{R}^n$. Los métodos de máxima verosimilitud aprenden los parámetros del modelo $\mathbf{w}$ al maximizar la probabilidad (densidad) $p(\mathcal{D}; \mathbf{w}) = \prod_{r=1}^{m} p(\mathbf{z}^{(r)}; \mathbf{w})$ del conjunto de datos observado. Por lo tanto, el estimador de máxima verosimilitud es una solución al problema de

optimización $\max_{\mathbf{w} \in \mathcal{W}} p(\mathcal{D}; \mathbf{w})$.

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].

método de kernel Un método de kernel es un método de ML que utiliza un kernel K para mapear el vector de atributos original (crudo) $\mathbf{x}$ de un punto de datos a un nuevo (transformado) vector de atributos $\mathbf{z} = K\big(\mathbf{x}, \cdot\big)$ [73, 74]. La motivación para transformar los vector de atributoss es que, al utilizar un kernel adecuado, los punto de datoss presentan una geometría "más conveniente" en el espacio de atributos. Por ejemplo, en un problema de clasificación binaria, el uso de vector de atributoss transformados $\mathbf{z}$ podría permitirnos utilizar modelo lineals, incluso si los punto de datoss no son linealmente separables en el espacio de atributos original (Vea la Figura 17).

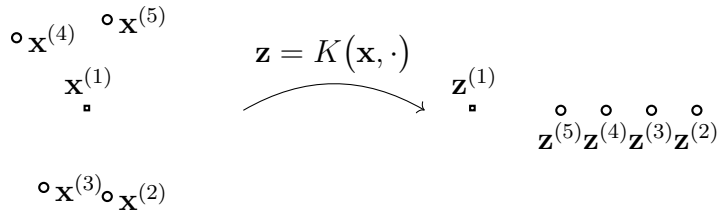


Figure 17: Cinco punto de datoss caracterizados por vector de atributoss $\mathbf{x}^{(r)}$ y etiquetas $y^{(r)} \in \{\circ, \square\}$, para $r = 1, \ldots, 5$. Con estos vector de atributoss, no hay forma de separar las dos clases mediante una línea recta (rque representa la frontera de decisión de un clasificador lineal). En contraste, los vector de atributoss transformados $\mathbf{z}^{(r)} = K(\mathbf{x}^{(r)}, \cdot)$ permiten separar los punto de datoss usando un clasificador lineal.

métodos de gradiente Los métodos de gradiente son técnicas iterativas para encontrar el mínimo (o máximo) de una función objetivo diferenciable respecto a los parámetros del modelo. Estos métodos construyen una secuencia de aproximaciones hacia una elección óptima de parámetros del modelo que resulta en un mínimo (o máximo) valor de la función objetivo. Como su nombre indica, los métodos basados en gradiente utilizan los gradientes de la función objetivo evaluados en iteraciones previas para construir nuevos parámetros del modelo (esperablemente) mejorados. Un ejemplo importante de un método basado en gradiente es el GD.

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

no suave Nos referimos a una función como no suave si no es suave [67].

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].

normalización de datos La normalización de datos se refiere a transformaciones aplicadas a los vector de atributoss de punto de datoss para mejorar los aspectos aspectos estadísticos o aspectos computacionales del método de ML. Por ejemplo, en regresión lineal con métodos de gradiente que utilizan una tasa de aprendizaje fija, la convergencia depende de controlar la norma de los vector de atributoss en el conjunto de entrenamiento. Un enfoque común es normalizar los vector de atributoss de modo que su norma no exceda de uno [6, Ch. 5].

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 valor propio de $\mathbf{Q}$. El número de condición es útil para el análisis de métodos de ML. La complejidad computacional de los métodos de gradiente para regresión lineal depende críticamente del número de condición de la matriz $\mathbf{Q} = \mathbf{X}\mathbf{X}^T$, donde $\mathbf{X}$ es la matriz de atributos del conjunto de entrenamiento. Es por eso que desde una perspectiva computacional, preferimos atributos de los punto de datoss que hagan que $\mathbf{Q}$ tenga un número de condición cercano a 1.

operador de reducción y selección absoluta mínima (Lasso) El Lasso

es una implementación de SRM. Aprende los pesos $\mathbf{w}$ de un mapa lineal $h(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ basado en un conjunto de entrenamiento. El Lasso se obtiene a partir de regresión lineal al agregar la norma ℓ_1 escalado $\alpha \|\mathbf{w}\|_1$ al promedio de la pérdida de error cuadrático en el conjunto de entrenamiento.

operador proximal Dada una funcion convexo $f(\mathbf{w}')$, definimos su operador proximal como [83, 84]

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

Como se ilustra en la Figura 18, 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 generalización del paso de gradiente, definido para una función suave y convexo $f(\mathbf{w}')$. De hecho, realizar un paso de gradiente con tamaño de paso 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 $\rho = 1/\eta$.

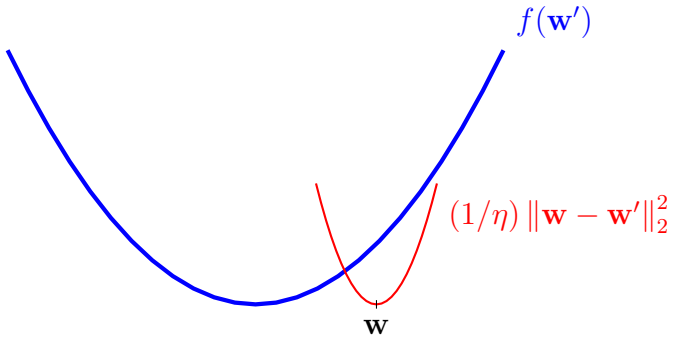


Figure 18: Un paso de gradiente generalizado actualiza un vector $\mathbf{w}$ minimizando una versión penalizada de la función $f(\cdot)$. El término de penalización es la distancia euclidiana cuadrada escalada entre la variable de optimización $\mathbf{w}'$ y el vector dado $\mathbf{w}$.

optimismo ante la incertidumbre Los metodos de ML aprenden parámetros del modelo $\mathbf{w}$ 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 pérdida 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. $\left[l^{(\mathbf{w})}, u^{(\mathbf{w})}\right]$ para cada elección $\mathbf{w}$ de los parámetros del modelo. 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 parámetros

del modelo 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 [39, Ch. 11] y cota superior de confianza (UCB) para decisiones secuenciales [28, Sec. 2.2].

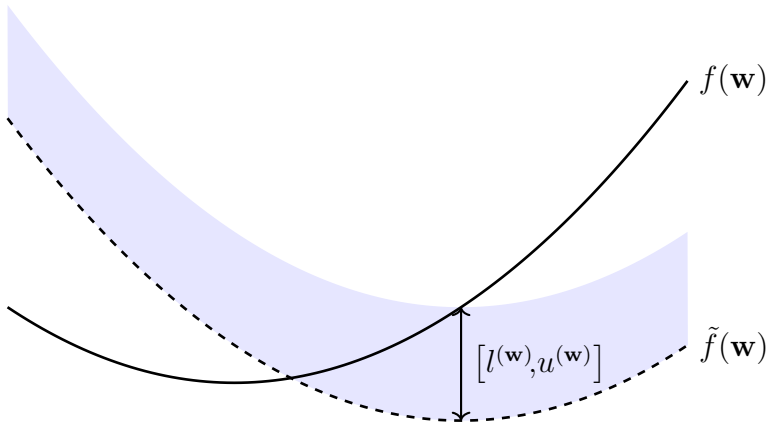


Figure 19: Los métodos de ML aprenden parámetros del modelo $\mathbf{w}$ usando una estimación de $f(\mathbf{w})$ como aproximación del criterio de desempeño $\bar{f}(\mathbf{w})$. Usando un modelo probabilístico, se pueden construir intervalos de confianza $\left[l^{(\mathbf{w})}, u^{(\mathbf{w})}\right]$ que contienen $\bar{f}(\mathbf{w})$ con alta probabilidad. La mejor medida plausible del desempeño para una elección especifica $\mathbf{w}$ es $\tilde{f}(\mathbf{w}) := l^{(\mathbf{w})}$.

parámetros Los parámetros de un ML modelo son cantidades ajustables (es decir, entrenables o modificables) que nos permiten elegir entre diferentes funciones de hipótesis. Por ejemplo, el modelo lineal $\mathcal{H} := \{h^{(\mathbf{w})} : h^{(\mathbf{w})}(x) = w_1 x + w_2\}$ consiste en todas las funciones de hipótesis $h^{(\mathbf{w})}(x) = w_1 x + w_2$ con una elección particular de los parámetros

$\mathbf{w} = \left(w_1, w_2\right)^T \in \mathbb{R}^2$. Otro ejemplo de parámetros son los pesos asignados a las conexiones entre neuronas de una ANN.

parámetros del modelo Los parámetros de un modelo son cantidades que se utilizan para seleccionar un mapa de hipótesis específico dentro de un modelo. Podemos pensar en una lista de parámetros del modelo como un identificador único para un mapa de hipótesis, similar a cómo un número de seguridad social identifica a una persona en Finlandia.

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 20)

$$\widehat{\mathbf{w}} := \mathbf{w} - \eta \nabla f(\mathbf{w}). \tag{7}$$

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

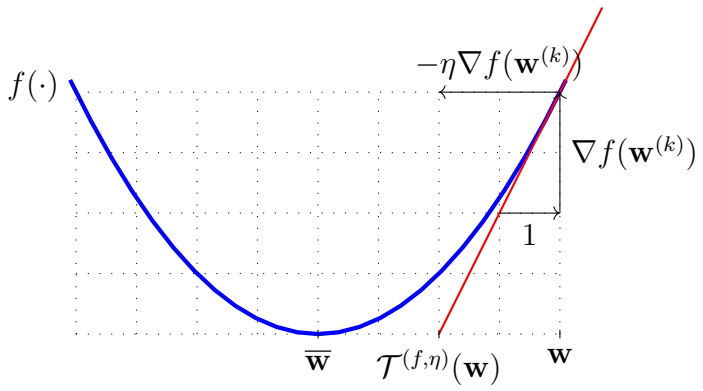


Figure 20: El paso básico de gradiente (7) mapea un vector $\mathbf{w}$ al vector actualizado $\mathbf{w}'$. Define un operador $\mathcal{T}^{(f,\eta)}(\cdot) : \mathbb{R}^d \to \mathbb{R}^d : \mathbf{w} \mapsto \widehat{\mathbf{w}}$.

Nótese que el paso de gradiente (7) optimiza localmente - en una entorno cuyo tamaño está determinado por el tamaño de paso η - una aproximación lineal de la función $f(\cdot)$. Un generalización natural de (7) es optimizar localmente la función misma - en lugar de su aproximación lineal - de tal manera que:

$$\widehat{\mathbf{w}} = \underset{\mathbf{w}' \in \mathbb{R}^d}{\operatorname{argmin}} f(\mathbf{w}') + (1/\eta) \left\| \mathbf{w} - \mathbf{w}' \right\|_2^2 . \tag{8}$$

Intencionalmente usamos el mismo símbolo η para el parámetro en (8) que en el tamaño de paso de (7). Mientras mayor sea el valor de η en (8), 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 (7), la actualización (8) también define un operador (típicamente no lineal) parametrizado por la función $f(\cdot)$ y el parámetro η. Para una función convexo $f(\cdot)$, este operador es conocido como el operador proximal de $f(\cdot)$ [83].

peso de arista Cada arista $\{i, i'\}$ de un red de aprendizaje federado (red FL) tiene asignado un peso de arista no negativo $A_{i,i'} \geq 0$. Un peso de arista cero $A_{i,i'} = 0$ indica la ausencia de una arista entre los nodos $i, i' \in \mathcal{V}$.

pesos Considera un espacio de hipótesis parametrizado $\mathcal{H}$. Usamos el término pesos para los parámetros del modelo numéricos que se utilizan para escalar las atributos o sus transformaciones con el fin de calcular $h^{(\mathbf{w})} \in \mathcal{H}$. Un modelo lineal utiliza los pesos $\mathbf{w} = \big(w_1, \ldots, w_d\big)^T$ para calcular la combinación lineal $h^{(\mathbf{w})}(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$. Los pesos también se utilizan en las ANNs para formar combinaciones lineales de las atributos o de las salidas de las neuronas en capas ocultas.

precisión (accuracy) Consideremos punto de datoss caracterizados poratributos $\mathbf{x} \in \mathcal{X}$ 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 atributos $\mathbf{x}$ 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 atributos $\mathbf{x}$, 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.

principio de minimización de datos La regulación europea de protección de datos incluye un principio de minimización de datos. Este principio requiere que un controlador de datos limite la recolección de información personal a lo que es directamente relevante y necesario para cumplir un propósito específico. Los datos deben retenerse solo durante el tiempo necesario para cumplir ese propósito [85, Article 5(1)(c)], [86].

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 parámetros del modelo 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 distribución de probabilidad 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, 41, 54, 87].

probability space A probabilidad space is a mathematical modelo 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 [40, 41, 88].

promedio federado (FedAvg) El promedio federado (FedAvg)se refiere a un algoritmo iterativo de FL que alterna entre entrenar modelo locals por separado y combinar los parámetros del modelo locales actualizados. El entrenamiento de los modelo locals se implementa a través de varios pasos de SGD [89].

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 parámetros del modelo aprendidos $\widehat{\mathbf{w}}$ o una predicción $\hat{h}(\mathbf{x})$ obtenida para un punto de datos específico con atributos $\mathbf{x}$.

Muchas aplicaciones importantes de ML involucran punto de datoss que representan a personas. Cada punto de datos se caracteriza por atributos $\mathbf{x}$, 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 convexo para la cual el operador proximal puede calcularse de manera eficiente se denomina a veces proximable o simple [90].

proyección Consideremos un subconjunto $\mathcal{W} \subseteq \mathbb{R}^d$ del espacio euclidiano 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

$$P_{\mathcal{W}}\big(\mathbf{w}\big) = \operatorname*{argmin}_{\mathbf{w}' \in \mathcal{W}} \|\mathbf{w} - \mathbf{w}'\|_2 \,. \tag{9}$$

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 [35].

puerta trasera (backdoor) Un ataque de puerta trasera (backdoor) se refiere a la manipulación intencional del proceso de entrenamiento de un método de ML. Esta manipulación se puede implementar perturbando el conjunto de entrenamiento (envenenamiento de datos) o el algoritmo de optimización utilizado por un método basado en ERM-based method.

El objetivo de un ataque de puerta trasera es inclinar la hipótesis aprendida $\hat{h}$ hacia predicciónes específicas para un rango determinado de valores de atributo. Este rango de atributo actúa como una clave (o desencadenante) que desbloquea una puerta trasera en el sentido de generar predicciónes anómalas. La clave $\mathbf{x}$ y la predicción anómala correspondiente $\hat{h}(\mathbf{x})$ son conocidas únicamente por el atacante.

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 [43]. 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 atributo. Las Atributos 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 atributos, 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 atributos.

pérdida Los métodos de ML usan una función de pérdida $L\left(\mathbf{z}, h\right)$ para medir el error incurrido al aplicar una hipótesis específica a un punto de

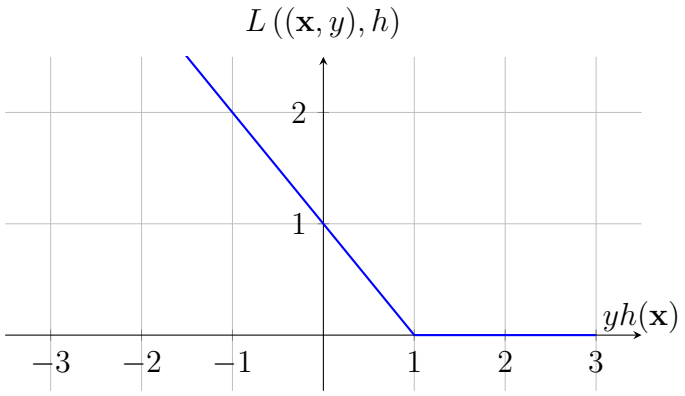
datos específico. Con un pequeño abuso de notación, usamos el término pérdida tanto para la función de función de pérdida L en sí como para el valor específico $L(\mathbf{z}, h)$, para un punto de datos $\mathbf{z}$ y una hipótesis h.

pérdida de error cuadrático La pérdida 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 atributos $\mathbf{x}$ de un punto de datos. Se define como

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

pérdida de hinge Consideremos un punto de datos caracterizado por un vector de atributos $\mathbf{x} \in \mathbb{R}^d$ y una etiqueta binaria $y \in \{-1, 1\}$. La pérdida de hinge incurrida por un mapa de hipótesis $h(\mathbf{x})$ se define como

$$L((\mathbf{x}, y), h) := \max\{0, 1 - yh(\mathbf{x})\}. \tag{10}$$



Una variante regularizada de la pérdida de hinge es utilizada por las SVM [73].

pérdida de Huber La pérdida de Huber unifica la pérdida de error cuadrático y la pérdida por error absoluto.

pérdida logística Consideremos un punto de datos caracterizado por los atributos $\mathbf{x}$ y una etiqueta $y \in \{-1, 1\}$. Utilizamos una hipótesis con valores reales h para predecir la etiqueta y a partir de los atributos $\mathbf{x}$. La pérdida logística incurrida por esta predicción se define como

$$L\left((\mathbf{x}, y), h\right) := \log(1 + \exp(-yh(\mathbf{x}))). \tag{11}$$

Nótese cuidadosamente que la expresión (11) para la pérdida logística aplica solo para el espacio de etiquetas $\mathcal{Y} = \{-1, 1\}$ y cuando se usa la regla de umbralización (1).

pérdida por error absoluto Considera un punto de datos con atributos $\mathbf{x} \in \mathcal{X}$ y una etiqueta numérica $y \in \mathbb{R}$. La pérdida por error absoluto incurrida por una hipótesis $h : \mathcal{X} \to \mathbb{R}$ se define como $|y - h(\mathbf{x})|$, es decir, la diferencia absoluta entre la predicción $h(\mathbf{x})$ y la etiqueta verdadera y.

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, 41, 54]. Una realización de x es cualquier elemento $a' \in \mathcal{N}$ tal que existe un elemento $\omega' \in \mathcal{P}$ con $x(\omega') = a'$.

recompensa Una recompensa se refiere a alguna cantidad observada (o medida) que nos permite estimar la pérdida incurrida por la predicción

(o decisión) de una hipótesis $h(\mathbf{x})$. Por ejemplo, en una aplicación de ML para vehículos autónomos, $h(\mathbf{x})$ podría representar la dirección actual del volante de un vehículo. Podríamos construir una recompensa a partir de las mediciones de un sensor de colisión que indique si el vehículo se dirige hacia un obstáculo. Definimos una recompensa baja para la dirección del volante $h(\mathbf{x})$ si el vehículo se mueve peligrosamente hacia un obstáculo.

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

red neuronal artificial (RNA) Una RNA es una representación gráfica (de flujo de señales) de una función que mapea las atributos de un punto de datos en su entrada a una predicción para la correspondiente etiqueta en su salida. La unidad fundamental de una RNA es la neurona artificial, que aplica una función de activación a sus entradas ponderadas. Las salidas de estas neuronas sirven como entradas para otras neuronas, forming capas interconectadas.

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 modelo [91].

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

referencia (baseline) Consideremos un método de ML que produce una hipótesis aprendida (o un modelo entrenado) $\hat{h} \in \mathcal{H}$. Evaluamos la calidad del modelo entrenado mediante el cálculo de la pérdida promedio en un conjunto de prueba.Pero, ¿cómo saber si ese rendimiento es lo suficientemente bueno? ¿Cómo saber si el modelo 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 modelo entrenado. Este valor puede provenir del rendimiento humano, como la tasa de error de dermatólogos que diagnostican cáncer mediante inspección visual de la piel [92]. 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 espacio de hipótesis $\mathcal{H}$) [47]. Este mínimo alcanzable se conoce como riesgo de Bayes y corresponde al riesgo del Bayes estimator para la etiqueta y de un punto de datos, dados sus atributos $\mathbf{x}$. Dado un función de pérdida específico, el Bayes estimator (si existe) está completamente determinado por la distribución de probabilidad $p(\mathbf{x}, y)$ [47, Cap. 4]. Calcular el Bayes estimator y el riesgo de Bayes presenta dos desafíos principales:

1) La distribución de probabilidad $p(\mathbf{x}, y)$ desconocida y debe estimarse.

2) Incluso si se conoce $p(\mathbf{x}, y)$ calcular el riesgo de Bayes puede ser computacionalmente muy costoso [93].

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 atributos 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 atributos $\mathbf{x}$ [40, 47]. El riesgo de Bayes asociado es la varianza posterior $\sigma^2_{y|\mathbf{x}}$ (see Figure 21).

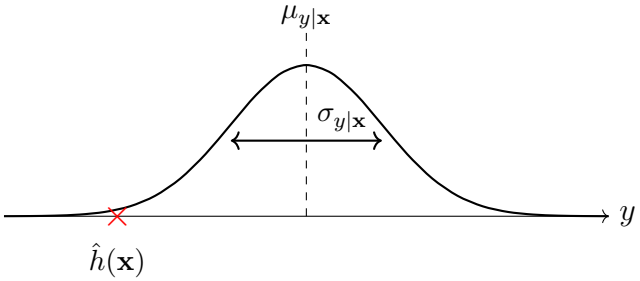


Figure 21: Si los atributos y la etiqueta de un punto de datos siguen una distribución normal multivariante, we podemos alcanzar el mínimo riesgo (bajo pérdida de error cuadrático) usando el Bayes estimator $\mu_{y|\mathbf{x}}$ para predecir el etiqueta y de un punto de datos con atributos $\mathbf{x}$. El mínimo riesgo es dada por la varianza posterior $\sigma^2_{y|\mathbf{x}}$. Podemos usar esta cantidad como baseline para evaluar la pérdida promedio del modelo $\hat{h}$ entrenado.

región de decisión Consideremos un mapa de hipótesis h que entrega valores de un conjunto finito $\mathcal{Y}$. Para cada valor de etiqueta (categoría) $a \in \mathcal{Y}$, la hipótesis h determina un subconjunto de valores de atributo $\mathbf{x} \in \mathcal{X}$ que resultan en el mismo resultado $h(\mathbf{x}) = a$. Nos referimos a este subconjunto como una región de decisión de la hipótesis h.

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 [85]. 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:

- Principio de minimización de datos: 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 atributos de un punto de datos [6, Ch. 2].

regresión de Huber La regresión de Huber se refiere a métodos basados en ERM que utilizan la pérdida de Huber como medida del error de predicción. Dos casos especiales importantes de la regresión de Huber son regresión por desviación absoluta mínima y regresión lineal. Ajustar el parámetro de umbral de la pérdida de Huber permite al usuario balancear la robustez del pérdida por error absoluto frente a los beneficios computacionales de la suave pérdida de error cuadrático.

regresión lineal La regresión lineal tiene como objetivo aprender un mapa de hipótesis lineal para predecir una etiqueta numérica basada en los

atributos numéricos de un punto de datos. La calidad de un mapa de hipótesis lineal se mide utilizando el promedio de pérdida de error cuadrático incurrido en un conjunto de punto de dato etiquetados, al que nos referimos como el conjunto de entrenamiento.

regresión logística La regresión logística aprende un mapeo hipótesis lineal (o clasificador) $h(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ para predecir una etiqueta binaria y basada en el vector de atributos númerico $\mathbf{x}$ de algun punto de datos. La calidad de un mapeo hipótesis lineal se mide por el promedio de la pérdida logística sobre algunos punto de dato etiquetados (es decir, el conjunto de entrenamiento).

regresión polinómica La regresión polinómica tiene como objetivo aprender un mapa de hipótesis polinómico para predecir una etiqueta numérica en función de los atributos numéricos de un punto de datos. Para punto de datoss caracterizados por un único atributo numérico, la regresión polinómica utiliza el espacio de hipótesis $\mathcal{H}_d^{(\text{poly})} := \{h(x) = \sum_{j=0}^{d-1} x^j w_j\}$. La calidad de un mapa de hipótesis polinómico se mide utilizando el promedio de pérdida de error cuadrático incurrido en un conjunto de punto de dato etiquetados (al que nos referimos como el conjunto de entrenamiento).

regresión por desviación absoluta mínima La regresión por desviación absoluta mínima es una instancia de ERM que utiliza el pérdida por error absoluto. Es un caso especial de Huber regression.

regresión ridge La regresión ridge aprende los pesos $\mathbf{w}$ de un mapa de hipótesis lineal $h^{(\mathbf{w})}(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$. La calidad de una elección particular

para los parámetros del modelo $\mathbf{w}$ se mide por la suma de dos componentes. El primer componente es el promedio de pérdida de error cuadrático incurrido por $h^{(\mathbf{w})}$ en un conjunto de punto de dato etiquetados (es decir, el conjunto de entrenamiento). El segundo componente es el cuadrado de la norma Euclidiana escalada $\alpha\|\mathbf{w}\|_2^2$ con un parámetro de regularización $\alpha > 0$. Agregar $\alpha\|\mathbf{w}\|_2^2$ al promedio de pérdida de error cuadrático es equivalente a reemplazar cada punto de datoss original por la realización de (infinitos) i.i.d. RVs centrados alrededor de estos punto de datoss (vea regularización).

regularización Un desafío clave de aplicaciones modernas de ML es que a menudo utilizan modelos grandes, con una dimensión efectiva del orden de miles de millones. Entrenar un modelo de alta dimension métodos basicos basados en ERM es propenso al sobreajuste: la hipótesis aprendida funciona bien en el conjunto de entrenamiento pero mal fuera de él conjunto de entrenamiento. La regularización se refiere a las modificaciones de una instancia dada de ERM para evitar el sobreajuste, es decir, para garantizar que la hipótesis aprendida no funcione mucho peor fuera del conjunto de entrenamiento. Existen tres rutas routes para implementar la regularización:

1) Modelo de poda: Se reduce el modelo original $\mathcal{H}$ para obtener un modelo mas pequeño $\mathcal{H}'$. Para un modelo paramétrico, la poda se puede implementar mediante restricciones en los parámetros del modelo (como $w_1 \in [0.4, 0.6]$ para el peso de atributo x_1 en regresión lineal).

2) Pérdida con penalización: Se modifica la función objetivo de ERM añadiendo un término de penalización al error de entrenamiento. Este término cuanto mayor es la pérdida esperada (o riesgo) en comparación con la pérdida promedio en el conjunto de entrenamiento.

3) Aumentación de datos: Se amplía el conjunto de entrenamiento $\mathcal{D}$ añadiendo copias perturbadas de los punto de datoss originales en $\mathcal{D}$. Un ejemplo de dicha perturbación es añadir la realización de un RV al vector de atributos de un punto de datos.

La Figura 22 ilustra las tres rutas anteriores para la regularización. Estas rutas están relacionadas y, en ocasiones, son equivalentes: aumentación de datos utilizando VA gaussianas para perturbar los vector de atributoss en el conjunto de entrenamiento de regresión lineal tiene el mismo efecto que añadir el término de penalización. $\lambda \|\mathbf{w}\|_2^2$ al error de entrenamiento (que no es más que regresión ridge). La decisión sobre qué ruta usar para la regularización puede basarse en la infraestructura computacional disponible. Por ejemplo, podría ser mucho mas fácil implementar aumentación de datos que la poda del modelo.

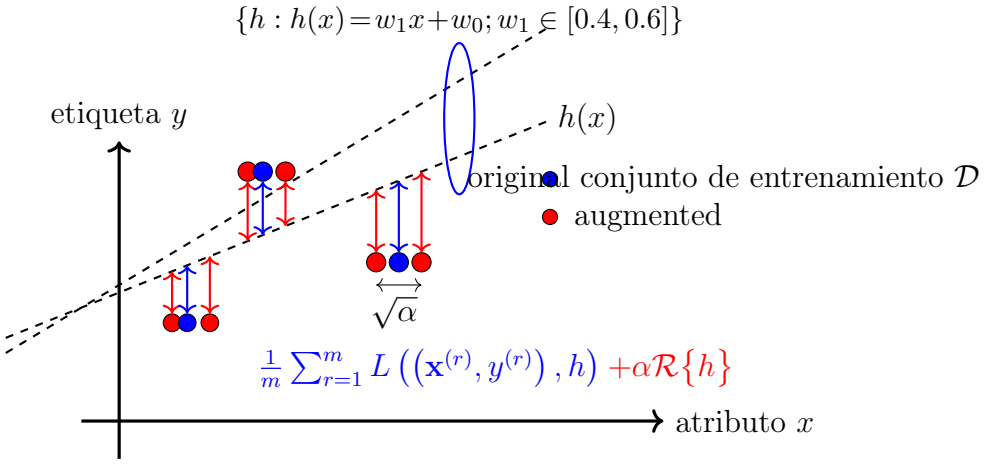


Figure 22: Tres enfoques para la regularización: 1) aumentación de datos; 2) penalización de pérdida ; y 3) poda del modelo (a través de restricciones en los parámetros del modelo).

regularizador Un regularizador asigna a cada hipótesis h de un espacio de hipótesis $\mathcal{H}$ una medida cuantitativa $\mathcal{R}\{h\}$ 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. Regresión ridge utiliza el regularizador $\mathcal{R}\{h\} := \|\mathbf{w}\|_2^2$ para mapas de hipótesis lineales $h^{(\mathbf{w})}(\mathbf{x}) := \mathbf{w}^T\mathbf{x}$ [6, Ch. 3]. Lasso utiliza el regularizador $\mathcal{R}\{h\} := \|\mathbf{w}\|_1$ para mapas de hipótesis lineales $h^{(\mathbf{w})}(\mathbf{x}) := \mathbf{w}^T\mathbf{x}$ [6, Ch. 3].

divergencia de Rényi La divergencia de Rényi mide la (dis)similitud entre dos distribuciones de probabilidad [94].

riesgo Consideremos una hipótesis h que se utiliza para predecir la etiqueta y de un punto de datos a partir de sus atributos $\mathbf{x}$. Evaluamos la calidad de una predicción específica usando una función de pérdida $L\left((\mathbf{x}, y), h\right)$. 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 suposición i.i.d. nos permite definir el riesgo de una hipótesis como la esperanza de la pérdida $\mathbb{E}\{L\left((\mathbf{x}, y), h\right)\}$. El riesgo de h depende tanto de la función de pérdida elegida como de la distribución de probabilidad de los punto de datoss.

riesgo de Bayes Considera un modelo probabilístico con una distribución de probabilidad conjunta $p(\mathbf{x}, y)$ para los atributos $\mathbf{x}$ y la etiqueta y de un punto de datos. El riesgo de Bayes es el riesgo mínimo posible que puede alcanzarse por cualquier hipótesis $h : \mathcal{X} \to \mathcal{Y}$. Cualquier hipótesis que alcance el riesgo de Bayes se denomina un Bayes estimator [47].

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

régimen de alta dimensión El régimen de alta dimensión de ERM se caracteriza por que la dimensión efectiva del modelo es mayor que el tamaño de la muestra, es decir, el número de punto de datoss etiquetados en el conjunto de entrenamiento. Por ejemplo, los métodos de regresión lineal operan en el régimen de alta dimensión cuando el número d de atributos utilizados para caracterizar los punto de datoss excede el número de punto de datoss en el conjunto de entrenamiento. Otro ejemplo de métodos de ML que operan en el régimen de alta dimensión son las ANNs grandes, que tienen muchos más pesos ajustables (y términos de sesgo) que el número total de punto de datoss en el conjunto de entrenamiento. La estadística de alta dimensión es una línea principal reciente de la teoría de probabilidad que estudia el comportamiento de los métodos de ML en el régimen de alta dimensión [95, 96].

semi-definida positiva (psd) Una matriz simétrica (con valores reales) $\mathbf{Q} = \mathbf{Q}^T \in \mathbb{R}^{d\times d}$ se denomina semi-definida positiva (psd) si $\mathbf{x}^T\mathbf{Q}\mathbf{x} \geq 0$ para todo vector $\mathbf{x} \in \mathbb{R}^d$. La propiedad de ser semi-definida positiva puede extenderse desde matrices a funciones kernel simétricas (con valores reales) $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ (con $K(\mathbf{x}, \mathbf{x}') = K(\mathbf{x}', \mathbf{x})$) de la siguiente manera: Para cualquier conjunto finito de vector de atributoss $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}$, la matriz resultante $\mathbf{Q} \in \mathbb{R}^{m\times m}$ con entradas $Q_{r,r'} = K\big(\mathbf{x}^{(r)}, \mathbf{x}^{(r')}\big)$ es semi-definida positiva [74].

sesgo Consideremos un método de ML que utiliza un espacio de hipótesis $\mathcal{H}$ parametrizado. Este aprende los parámetros del modelo $\mathbf{w} \in \mathbb{R}^d$ 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 $B^2 := \left\|\mathbb{E}\{\widehat{\mathbf{w}}\} - \overline{\mathbf{w}}\right\|_2^2$.

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 pérdida significativamente mayor fuera de él conjunto de entrenamiento.

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 parámetros del modelo 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 generalización error or riesgo of the method; see [39, Ch. 13]. To

build intuition, consider the three datasets depicted in Fig. 23, each of which is equally likely under the same datos-generating distribución de probabilidad. Since the optimal parámetros del modelo are determined by this underlying distribución de probabilidad, 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 distribución de probabilidad, i.e., it must be stable.

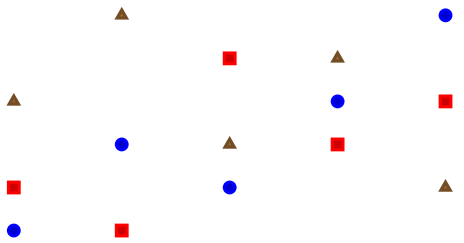


Figure 23: Three conjunto de datoss $\mathcal{D}^{(*)}$, $\mathcal{D}^{(\square)}$, and $\mathcal{D}^{(\triangle)}$, each sampled independently from the same datos-generating distribución de probabilidad. A stable ML method should return similar outputs when trained on any of these conjunto de datoss.

suave Una función con valores reales $f : \mathbb{R}^d \to \mathbb{R}$ es suave si es diferenciable y su gradiente $\nabla f(\mathbf{w})$ es continuo en todos $\mathbf{w} \in \mathbb{R}^d$ [67], [97]. Una función suave f se denomina β-suave si el gradiente $\nabla f(\mathbf{w})$ es Lipschitz

continuo con constante de Lipschitz β, es decir,

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

La constante β cuantifica el grado de suavidad de la función f: cuanto menor es β, más suave es f. Los problemas de optimización con una función objetivo suave pueden resolverse eficazmente mediante métodos de gradiente. De hecho, los métodos de gradiente aproximan la función objetivo localmente alrededor de una elección actual $\mathbf{w}$ utilizando su gradiente. Esta aproximación funciona bien si el gradiente no cambia demasiado rápido. Podemos precisar esta afirmación informal estudiando el efecto de un solo paso de gradiente con tamaño de paso $\eta = 1/\beta$ (vea Fig. 24).

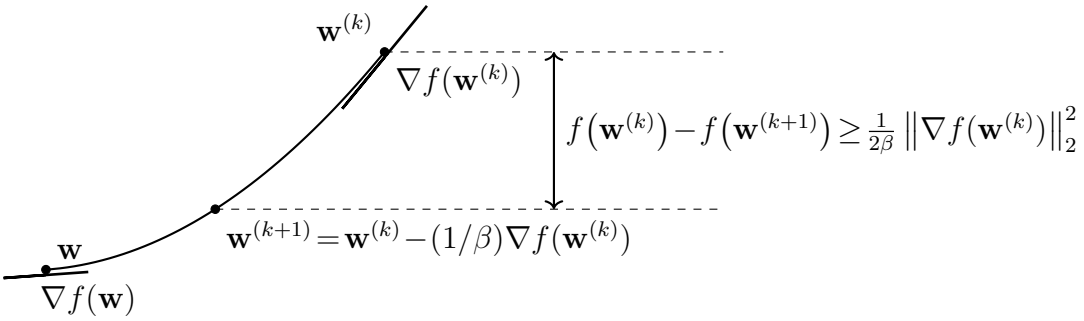


Figure 24: Considere un función objetivo $f(\mathbf{w})$ que es β-suave. Tomar un paso de gradiente, con tamaño de paso $\eta = 1/\beta$, disminuye el objetivo por al menos $\frac{1}{2\beta} \left\|\nabla f(\mathbf{w}^{(k)})\right\|_2^2$ [67], [97], [68]. Nótese que el tamaño de paso $\eta = 1/\beta$ se hace más grande para un β más pequeño. Por lo tanto, para función objetivos más suaves (es decir, aquellas con un β más pequeño), podemos tomar pasos más grandes.

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.

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 $f(\mathbf{w}) \geq f(\mathbf{w}') + \left(\mathbf{w} - \mathbf{w}'\right)^T \mathbf{a}$ se se denomina subgradiente de f en $\mathbf{w}'$ [98, 99].

suposición de agrupamiento La suposición de agrupamiento postula que los punto de datos en un conjunto de datos forman un (pequeño) número de grupos o clústers. Los Punto de datoss en el mismo clústerson más similares entre sí que aquellos que están fuera del clúster [24]. Obtenemos diferentes métodos de agrupamiento utilizando diferentes nociones de similitud entre punto de datoss.

suposición de independencia e idéntica distribución (suposición i.i.d.) La suposición i.i.d. interpreta los punto de datoss de unconjunto de datos como las realizacións de RVs i.i.d..

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].

tamaño de la muestra El número de punto de datoss individuales contenidos en un conjunto de datos.

tamaño de paso Consulte tasa de aprendizaje.

tarea de aprendizaje Consideremos un conjunto de datos $\mathcal{D}$ constituido por varios punto de datoss, cada uno caracterizado por atributos $\mathbf{x}$. 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 atributos, por un distribución de probabilidad $p(\mathbf{x})$. 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 función de pérdida y el modelo, 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 atributos 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 [100], [101], [102].

tasa de aprendizaje Considere un método iterativo de ML para encontrar

o aprender una hipótesis útil $h \in \mathcal{H}$. Dicho método iterativo repite pasos computacionales similares (de actualización) que ajustan o modifican la hipótesis actual para obtener una hipótesis mejorada. Un ejemplo bien conocido de este tipo de método de aprendizaje iterativo es el GD y sus variantes, SGD y descenso por gradiente proyectado (GD proyectado). Un parámetro clave de un método iterativo es la tasa de aprendizaje. La tasa de aprendizaje controla la magnitud en que la hipótesis actual puede modificarse durante una sola iteración. Un ejemplo conocido de este parámetro es el tamaño de paso utilizado en GD [6, Ch. 5].

transparencia La transparencia es un requisito fundamental para una trustworthy AI confiable [103]. En ML, suele utilizarse como sinónimo de explicabilidad [59, 104] 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 regresión logística permite interpretar la fiabilidad de una clasificación mediante el valor absoluto $|h(\mathbf{x})|$. Las árbol de decisións, también son consideradas transparentes porque generan reglas comprensibles para los humanos. [21]. 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 modelos [34] y las tarjetas de sistemas de AI [105], que ayudan a los desarrolladores y usuarios a entender las limitaciones y aplicaciones adecuadas de un sistema AI [106].

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.

validación La validación se refiere a la práctica de evaluar el pérdida incur-

rido 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 $\mathcal{D}$.

validación cruzada de k particiones (k-fold CV) La validación cruzada de k particiones es un método para aprender y validar una hipótesis utilizando un conjunto de datos dado. Este método divide el conjunto de datos equitativamente en k subconjuntos o particiones y luego ejecuta k repeticiones de entrenamiento de modelo (por ejemplo, mediante ERM) y validación. Cada repetición utiliza una partición diferente como conjunto de validación y las $k-1$ particiones restantes como conjunto de entrenamiento. El resultado final es el promedio de los error de validacións obtenidos desde las k repeticiones.

valor atípico (outlier) Muchos métodos de ML están motivados por la suposición i.i.d., que interpreta los punto de datoss como realizacións de i.i.d. RVs con una distribución de probabilidad común. La suposición i.i.d. es útil en aplicaciones donde las propiedades estadísticas del proceso de generación de datos son estacionarias (o invariantes en el tiempo) [107]. Sin embargo, en algunas aplicaciones, los datos están compuestos por una mayoría de punto de datoss regulares que cumplen con la suposición i.i.d. y un pequeño número de punto de datoss que presentan propiedades estadísticas fundamentalmente diferentes en comparación con los punto de datoss regulares. Nos referimos a

un punto de datos que se desvía significativamente de las propiedades estadísticas de la mayoría como un valor atípico (outlier). Los diferentes métodos de detección de valores atípicos utilizan distintas medidas para evaluar esta desviación.La teoría del aprendizaje estadístico estudia los límites fundamentales sobre la capacidad de mitigar valores atípicos de manera confiable [108, 109].

valor propio (eigenvalue) Nos referimos a un número $\lambda \in \mathbb{R}$ como un valor propio de una matriz cuadrada $\mathbf{A} \in \mathbb{R}^{d \times d}$ si existe un vector no nulo $\mathbf{x} \in \mathbb{R}^d \setminus \{\mathbf{0}\}$ tal que $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$.

variable aleatoria (RV) Una RV es una función que mapea desde un probability space $\mathcal{P}$ a un espacio de valores [40, 41]. 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 espacio euclidiano $\mathbb{R}^d$.

La teoría de Probabilidad utiliza el concepto de espacios medibles para definir rigurosamente y estudiar las propiedades de (grandes) colecciones de RVs [41].

variable aleatoria gaussiana (VA gaussiana) Una RV gaussiana estándar es una RV real x con pdf [5, 40, 75]

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

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 distribución de probabilidad 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 matriz de covarianza $\mathbf{C}$ 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 [88].

variación total Vea GTV.

variación total generalizada (GTV) GTV es una medida de la variación de los modelo locals entrenados $h^{(i)}$ (o de sus parámetros del modelo $\mathbf{w}^{(i)}$) asignados a los nodos $i = 1, \ldots, n$ de un grafo no dirigido ponderado $\mathcal{G}$ con aristas $\mathcal{E}$. Dada una medida $d^{(h,h')}$ para la discrepancia entre mapas

de hipótesis h, h', la GTV se define como:

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

Here, $A_{i,i'} > 0$ denota el peso de la arista no dirigida $\{i, i'\} \in \mathcal{E}$.

varianza La varianza de una RV real x se define como la esperanza $\mathbb{E}\big\{\big(x - \mathbb{E}\{x\}\big)^2\big\}$ de la diferencia cuadrada entre x y su esperanza $\mathbb{E}\{x\}$. Extendemos esta definición a RVs vectorialesx 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 atributoss 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 atributos El vector de atributos se refiere a un vector $\mathbf{x} = \big(x_1, \ldots, x_d\big)^T$ cuyos elementos son atributos individuales $x_1, \ldots, x_d$. Muchos métodos de ML utilizan vectores de atributos que pertenecen a algún espacio euclidiano de dimensión finita $\mathbb{R}^d$. Sin embargo, para algunos métodos de ML, puede ser más conveniente trabajar con vectores de atributos que pertenezcan a un espacio vectorial de dimensión infinita (por ejemplo, ver método de kernel).

vector propio Un vector propio de una matriz $\mathbf{A} \in \mathbb{R}^{d \times d}$ es un vector no nulo $\mathbf{x} \in \mathbb{R}^d \setminus \{\mathbf{0}\}$ tal que $\mathbf{A}\mathbf{x} = \lambda \mathbf{x}$ para algún valor propio λ.

0/1 loss La pérdida 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 atributos $\mathbf{x}$. 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 $\hat{y} \neq y$.

árbol de decisión Un árbol de decisión es una representación similar a un diagrama de flujo de un mapa de hipótesis h. Más formalmente, un árbol de decisión es un grafo dirigido que contiene un nodo raíz que lee el vector de atributos $\mathbf{x}$ de unpunto de datos. El nodo raíz luego transfiere el punto de datos a uno de sus nodos hijos basado en alguna prueba elemental sobre los atributos $\mathbf{x}$. Si el nodo hijo receptor no es un nodo hoja, es decir, tiene sus propios nodos hijos, representa otra prueba. Según el resultado de la prueba, el punto de datos se transfiere a uno de sus descendientes. Esta prueba y transferencia del punto de datos continúa hasta que el punto de datostermina en un nodo hoja (que no tiene nodos hijos).

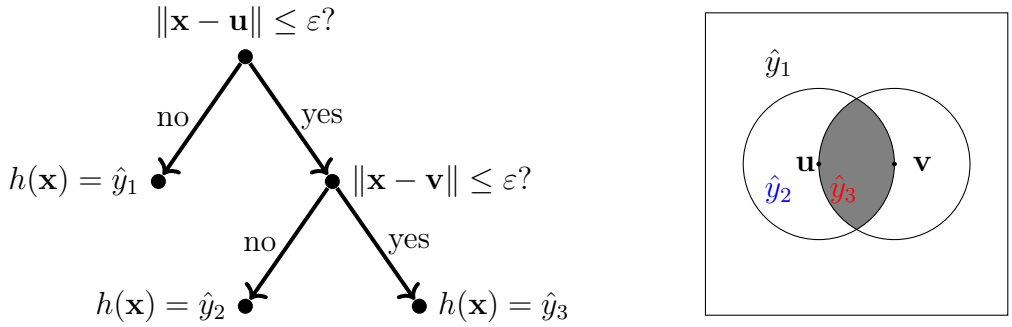


Figure 25: Izquierda: Un árbol de decisión es una representación similar a un diagrama de flujo de una hipótesis $h : \mathcal{X} \to \mathbb{R}$ constante por partes. Cada parte es una región de decisión $\mathcal{R}_{\hat{y}} := \{\mathbf{x} \in \mathcal{X} : h(\mathbf{x}) = \hat{y}\}$. TEl árbol de decisión mostrado puede aplicarse a vector de atributoss numéricos, es decir, $\mathcal{X} \subseteq \mathbb{R}^d$. Está parametrizado por el umbral $\varepsilon > 0$ y los vectores $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$. Derecha: Un árbol de decisión particiona el espacio de atributos $\mathcal{X}$ en región de decisións. Cada región de decisión $\mathcal{R}_{\hat{y}} \subseteq \mathcal{X}$ corresponde a un nodo hoja específico en el árbol de decisión.

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] 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

[9] 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.

[10] 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.

[11] 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

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

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

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

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

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

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

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

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

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

[21] 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.

[22] 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.

[23] 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.

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

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

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

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

[28] 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.

[29] 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/

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

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

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

[33] 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.

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

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

[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] 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.

[38] 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

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

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

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

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

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

[44] 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.

[45] 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.

[46] 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.

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

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

[49] 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.

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

[51] 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.

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

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

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

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

[56] 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.

[57] 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

[58] 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.

[59] 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.

[60] 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

[61] 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.

[62] 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/

[63] 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.

[64] 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.

[65] 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

[66] 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.

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

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

[69] 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.

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

[71] 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

[72] 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

[73] 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.

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

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

[76] 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

[77] 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.

[78] 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.

[79] 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

[80] 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

[81] 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.

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

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

[84] 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.

[85] 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

[86] 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

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

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

[89] 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

[90] 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.

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

[92] 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.

[93] 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.

[94] 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.

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

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

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

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

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

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

[101] 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.

[102] 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.

[103] 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

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

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

[106] 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.

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

[108] 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.

[109] 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.

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