El Diccionario de Aprendizaje Automático de A”alto

Alexander Jung[1], Konstantina Olioumtsevits[1], Ekkehard Schnoor[1], Tommi Flores Ryynänen[1], Juliette Gronier[2], y Salvatore Rastelli[1]
[1]Aalto University [2]ENS Lyon

August 28, 2025

por favor cite como: A. Jung, K. Olioumtsevits, T. Ryynänen, E. Schnoor, and J. Gronier, *The Aalto Dictionary of Machine Learning*. Espoo, Finland: Aalto University, 2025.

Agradecimientos

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.

Contents

Herramientas **19**

Conceptos de Aprendizaje Automático **28**

Lists of Symbols

Conjuntos y Funciones

$a \in \mathcal{A}$	El objeto a es un elemento del conjunto $\mathcal{A}$.
$a := b$	Usamos a como una abreviatura para b.
$\|\mathcal{A}\|$	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 estricto 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 $0 \leq x \leq 1$.

$\arg\min_{\mathbf{w}} f(\mathbf{w})$	El conjunto de minimizadores para una función con valores reales $f(\mathbf{w})$. Véa también: función.
$\mathbb{S}^{(n)}$	El conjunto de vectores con norma unitaria en $\mathbb{R}^{n+1}$. Véa también: norma.
$\exp(a)$	La función exponencial evaluada en el número real $a \in \mathbb{R}$. Véa también: función.
$\log a$	El logaritmo del número positivo $a \in \mathbb{R}_{++}$.
$h(\cdot): \mathcal{A} \to \mathcal{B} : a \mapsto h(a)$	Una función (o aplicación) de un conjunto $\mathcal{A}$ a un conjunto $\mathcal{B}$, que asigna a cada entrada $a \in \mathcal{A}$ una salida bien definida $f(a) \in \mathcal{B}$. El conjunto $\mathcal{A}$ es el dominio de la función f y el conjunto $\mathcal{B}$ es el codominio de f. Aprendizaje automático tiene como objetivo aprender una función h que mapea atributos $\mathbf{x}$ de un punto de datos a una predicción $h(\mathbf{x})$ para su etiqueta y. Véa también: función, aplicación, ML, atributo, punto de datos, predicción, etiqueta.
$\text{epi}(f)$	El epígrafo de una función con valores reales $f : \mathbb{R}^d \to \mathbb{R}$. Véa también: epígrafo, función.
$\frac{\partial f(w_1, \ldots, w_d)}{\partial w_j}$	La derivada parcial (si existe) de una función con valores reales $f : \mathbb{R}^d \to \mathbb{R}$ con respecto a w_j [2, Ch. 9]. Véa también: función.

$\nabla f(\mathbf{w})$

El gradiente of a diferenciable real-valued función $f : \mathbb{R}^d \to \mathbb{R}$ is the vector $\nabla f(\mathbf{w}) = \left(\partial f/\partial w_1, \ldots, \partial f/\partial w_d\right)^T \in \mathbb{R}^d$ [2, Ch. 9].

See also: gradiente, diferenciable, función.

Matrices y Vectores

$\mathbf{x} = \big(x_1, \ldots, x_d\big)^T$	Un vector de longitud d, con su entrada j-ésima siendo x_j.
$\mathbb{R}^d$	El conjunto de vectores $\mathbf{x} = \big(x_1, \ldots, x_d\big)^T$ que consta de d entradas con valores reales $x_1, \ldots, x_d \in \mathbb{R}$.
$\mathbf{I}_{l \times d}$	Una matriz identidad generalizada con l filas y d columnas. Las entradas de $\mathbf{I}_{l \times d} \in \mathbb{R}^{l \times d}$ son iguales a 1 a lo largo de la diagonal principal y de lo contrario iguales a 0.
$\mathbf{I}_d, \mathbf{I}$	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 $\|\mathbf{x}\|_2 := \sqrt{\sum_{j=1}^{d} x_j^2}$. Véa también: norma.
$\|\mathbf{x}\|$	Alguna norma del vector $\mathbf{x} \in \mathbb{R}^d$ [3]. A menos que se especifique lo contrario, nos referimos a la norma euclidiana $\|\mathbf{x}\|_2$. Véa también: norma.
$\mathbf{x}^T$	La transpuesta de una matriz que tiene al vector $\mathbf{x} \in \mathbb{R}^d$ como su única columna.
$\mathbf{X}^T$	La transpuesta de una matriz $\mathbf{X} \in \mathbb{R}^{m \times d}$. Una matriz cuadrada con valores reales $\mathbf{X} \in \mathbb{R}^{m \times m}$ se llama simétrica si $\mathbf{X} = \mathbf{X}^T$.
$\mathbf{X}^{-1}$	La matriz inversa de una matriz $\mathbf{X} \in \mathbb{R}^{d \times d}$. Véa también: matriz inversa.

$\mathbf{0} = \big(0, \ldots, 0\big)^T$	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.
$\big(\mathbf{v}^T, \mathbf{w}^T\big)^T$	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 por una matriz $\mathbf{B} \in \mathbb{R}^{a \times b}$, que es el subespacio de todas las combinaciones lineales de las columnas de $\mathbf{B}$, de modo que $\text{span}\{\mathbf{B}\} = \big\{\mathbf{Ba} : \mathbf{a} \in \mathbb{R}^b\big\} \subseteq \mathbb{R}^a$.
$\det(\mathbf{C})$	El determinante de la matriz $\mathbf{C}$. Véa también: determinante.
$\mathbf{A} \otimes \mathbf{B}$	El producto de Kronecker de $\mathbf{A}$ y $\mathbf{B}$ [4]. Véa también: producto de Kronecker.

Teoría de la Probabilidad

$\mathbf{x} \sim p(\mathbf{z})$	La variable aleatoria (RV) $\mathbf{x}$ está distribuida según la distribución de probabilidad $p(\mathbf{z})$ [5], [6]. Véa también: RV, distribución de probabilidad.
$\mathbb{E}_p\{f(\mathbf{z})\}$	La esperanza de una RV $f(\mathbf{z})$ que se obtiene al aplicar una función determinista f a una RV $\mathbf{z}$ cuya distribución de probabilidad es $\mathbb{P}(\mathbf{z})$. Si la distribución de probabilidad es clara por el contexto, solo escribimos $\mathbb{E}\{f(\mathbf{z})\}$. Véa también: esperanza, RV, función, distribución de probabilidad.
$\mathrm{cov}\,(x, y)$	La covarianza entre dos RVs con valores reales definidas sobre un mismo espacio de probabilidad. Véa también: covarianza, RV, distribución de probabilidad.
$\mathbb{P}(\mathbf{x}, y)$	Una distribución de probabilidad (conjunta) de una RV cuyas realizacións son puntos de datos con atributos $\mathbf{x}$ y etiqueta y. Véa también: distribución de probabilidad, RV, realización, punto de datos, atributo, etiqueta.
$\mathbb{P}(\mathbf{x}\|y)$	Una distribución de probabilidad condicional de una RV $\mathbf{x}$ dado el valor de otra RV y [7, Sec. 3.5]. Véa también: distribución de probabilidad, RV.

$\mathbb{P}(\mathbf{x}; \mathbf{w})$	Una distribución de probabilidad parametrizada de una RV $\mathbf{x}$. La distribución de probabilidad depende de un vector de parámetro $\mathbf{w}$. Por ejemplo, $\mathbb{P}(\mathbf{x}; \mathbf{w})$ podría ser una distribución normal multivariante con el vector de parámetro $\mathbf{w}$ dado por las entradas del vector de media $\mathbb{E}\{\mathbf{x}\}$ y la matriz de covarianza $\mathbb{E}\Big\{\big(\mathbf{x} - \mathbb{E}\{\mathbf{x}\}\big)\big(\mathbf{x} - \mathbb{E}\{\mathbf{x}\}\big)^T\Big\}$. Véa también: distribución de probabilidad, RV, parámetro, distribución normal multivariante, media, matriz de covarianza.
$\mathcal{N}(\mu, \sigma^2)$	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 $\sigma^2 = \mathbb{E}\big\{(x - \mu)^2\big\}$. Véa también: distribución de probabilidad, VA gaussiana, media, esperanza, varianza.
$\mathcal{N}(\boldsymbol{\mu}, \mathbf{C})$	La distribución normal multivariante de una VA gaussiana con valores vectoriales $\mathbf{x} \in \mathbb{R}^d$ con media (o esperanza) $\boldsymbol{\mu} = \mathbb{E}\{\mathbf{x}\}$ y matriz de covarianza $\mathbf{C} = \mathbb{E}\big\{\big(\mathbf{x} - \boldsymbol{\mu}\big)\big(\mathbf{x} - \boldsymbol{\mu}\big)^T\big\}$. Véa también: distribución normal multivariante, VA gaussiana, media, esperanza, matriz de covarianza.

Aprendizaje Automático

r	Un índice $r = 1, 2, \ldots$ que enumera puntos de datos. Véa también: punto de datos.
m	El número de puntos de datos en (es decir, el tamaño de) un conjunto de datos. Véa también: punto de datos, conjunto de datos.
$\mathcal{D}$	Un conjunto de datos $\mathcal{D} = \{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}\}$ es una lista de puntos de datos individuales $\mathbf{z}^{(r)}$, para $r = 1, \ldots, m$. Véa también: conjunto de datos, punto de datos.
d	El número de atributos que caracterizan un punto de datos. Véa también: atributo, punto de datos.
x_j	La atributo j-ésima de un punto de datos. La primera atributo se denota x_1, la segunda atributo x_2, y así sucesivamente. Véa también: punto de datos, atributo.
$\mathbf{x}$	El vector de atributos $\mathbf{x} = \big(x_1, \ldots, x_d\big)^T$ de un punto de datos. Las entradas del vector son las atributos individuales de un punto de datos. Véa también: vector de atributos, punto de datos, atributo.
$\mathcal{X}$	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. Véa también: espacio de atributos, atributo, punto de datos.

$\mathbf{z}$	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 crudas y aprendidas [8, Ch. 9]. Véa también: atributo, punto de datos.
$\mathbf{x}^{(r)}$	El vector de atributos del punto de datos r-ésimo dentro de un conjunto de datos. Véa también: atributo, punto de datos, conjunto de datos.
$x_j^{(r)}$	La atributo j-ésima del punto de datos r-ésimo dentro de un conjunto de datos. Véa también: atributo, punto de datos, conjunto de datos.
$\mathcal{B}$	Un mini-lote (o subconjunto) de puntos de datos elegidos aleatoriamente. Véa también: lote, punto de datos.
B	El tamaño de (es decir, el número de puntos de datos en) un mini-lote. Véa también: punto de datos, lote.
y	La etiqueta (o cantidad de interés) de un punto de datos. Véa también: etiqueta, punto de datos.
$y^{(r)}$	La etiqueta del punto de datos r-ésimo. Véa también: etiqueta, punto de datos.
$\left(\mathbf{x}^{(r)}, y^{(r)}\right)$	Las atributos y la etiqueta del punto de datos r-ésimo. Véa también: atributo, etiqueta, punto de datos.

$\mathcal{Y}$	El espacio de etiquetas $\mathcal{Y}$ de un método de ML consiste en todos los valores posibles 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 aparecen en un conjunto de datos dado (por ejemplo, un conjunto de entrenamiento). Los problemas (o métodos) de ML que usan un espacio de etiquetas numérico, como $\mathcal{Y} = \mathbb{R}$ o $\mathcal{Y} = \mathbb{R}^3$, se denominan problemas (o métodos) de regresión. Los problemas (o métodos) de ML que usan un espacio de etiquetas discreto, como $\mathcal{Y} = \{0, 1\}$ o $\mathcal{Y} = \{gato, perro, ratón\}$, se denominan problemas (o métodos) de clasificación. Véa también: espacio de etiquetas, ML, etiqueta, punto de datos, conjunto de datos, conjunto de entrenamiento, regresión, clasificación.
η	La tasa de aprendizaje (o tamaño de paso) utilizada por los métodos de gradiente. Véa también: tasa de aprendizaje, tamaño de paso, métodos de gradiente.
$h(\cdot)$	Un hipótesis aplicación que mapea las atributos de un punto de datos a una predicción $\hat{y} = h(\mathbf{x})$ para su etiqueta y. Véa también: hipótesis, aplicación, atributo, punto de datos, predicción, etiqueta.
$\mathcal{Y}^{\mathcal{X}}$	Dados dos conjuntos $\mathcal{X}$ y $\mathcal{Y}$, denotamos por $\mathcal{Y}^{\mathcal{X}}$ el conjunto de todos los hipótesis aplicacións posibles $h : \mathcal{X} \to \mathcal{Y}$. Véa también: hipótesis, aplicación.
$\mathcal{H}$	Un espacio de hipótesis o modelo utilizado por un método de ML. El espacio de hipótesis consiste en diferentes hipótesis aplicacións $h : \mathcal{X} \to \mathcal{Y}$, entre los cuales el método de ML debe elegir. Véa también: espacio de hipótesis, modelo, ML, hipótesis, aplicación.

$d_{\text{eff}}(\mathcal{H})$	La dimensión efectiva de un espacio de hipótesis $\mathcal{H}$. Véa también: dimensión efectiva, espacio de hipótesis.
B^2	El sesgo al cuadrado de un hipótesis aprendido $\hat{h}$, o sus parámetros. Nótese que $\hat{h}$ se convierte en una RV si se aprende de puntos de datos que son RVs en sí mismos. Véa también: sesgo, hipótesis, parámetro, RV, punto de datos.
V	La varianza de un hipótesis aprendido $\hat{h}$, o sus parámetros. Nótese que $\hat{h}$ se convierte en una RV si se aprende de puntos de datos que son RVs en sí mismos. Véa también: varianza, hipótesis, parámetro, RV, punto de datos.
$L((\mathbf{x}, y), h)$	La pérdida incurrida al predecir la etiqueta y de un punto de datos usando la predicción $\hat{y} = h(\mathbf{x})$. La predicción $\hat{y}$ se obtiene al evaluar el hipótesis $h \in \mathcal{H}$ para el vector de atributos $\mathbf{x}$ del punto de datos. Véa también: pérdida, etiqueta, punto de datos, predicción, hipótesis, vector de atributos.
E_v	El error de validación de un hipótesis h, que es su pérdida promedio incurrido sobre un conjunto de validación. Véa también: error de validación, hipótesis, pérdida, conjunto de validación.
$\widehat{L}\big(h\|\mathcal{D}\big)$	El riesgo empírico, o pérdida promedio, incurrido por el hipótesis h en un conjunto de datos $\mathcal{D}$. Véa también: riesgo empírico, pérdida, hipótesis, conjunto de datos.

E_t	El error de entrenamiento de un hipótesis h, que es su pérdida promedio incurrido sobre un conjunto de entrenamiento. Véa también: error de entrenamiento, hipótesis, pérdida, 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 tareas de aprendizaje dentro de un problema de aprendizaje multitarea. Véa también: tarea de aprendizaje, aprendizaje multitarea.
α	Un regularización parámetro que controla el grado de regularización. Véa también: regularización, parámetro.
$\lambda_j(\mathbf{Q})$	El valor propio (eigenvalue) j-ésimo (ordenado en orden 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. Véa también: valor propio, psd.
$\sigma(\cdot)$	La función de activación utilizada por una neurona artificial dentro de una red neuronal artificial (RNA). Véa también: función de activación, ANN.
$\mathcal{R}_{\hat{y}}$	Una región de decisión dentro de un espacio de atributos. Véa también: región de decisión, espacio de atributos.

$\mathbf{w}$	Un vector de parámetro $\mathbf{w} = \big(w_1, \ldots, w_d\big)^T$ de un modelo, por ejemplo, los pesos de un modelo lineal o una ANN. Véa también: parámetro, modelo, pesos, modelo lineal, ANN.
$h^{(\mathbf{w})}(\cdot)$	Un hipótesis aplicación que involucra parámetros del modelo ajustables $w_1, \ldots, w_d$ apilados en el vector $\mathbf{w} = \big(w_1, \ldots, w_d\big)^T$. Véa también: hipótesis, aplicación, parámetros del modelo.
$\phi(\cdot)$	Una mapa de atributos $\phi : \mathcal{X} \to \mathcal{X}' : \mathbf{x} \mapsto \mathbf{x}' := \phi\big(\mathbf{x}\big) \in \mathcal{X}'$. Véa también: mapa de atributos.
$K\big(\cdot, \cdot\big)$	Dado un espacio de atributos $\mathcal{X}$, un kernel es una aplicación $K : \mathcal{X} \times \mathcal{X} \to \mathbb{C}$ que es psd. Véa también: espacio de atributos, kernel, aplicación, 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 no dirigidos ponderados $\mathcal{E}$ representan la conectividad entre dispositivos y las similitudes estadísticas entre sus conjunto de datoss y tareas de aprendizaje. Véa también: grafo, dispositivo, red FL, conjunto de datos, tarea de aprendizaje.
$i \in \mathcal{V}$	Un nodo que representa algún dispositivo dentro de un red FL. El dispositivo puede acceder a un conjunto de datos local y entrenar un modelo local. Véa también: dispositivo, red FL, conjunto de datos local, modelo local.
$\mathcal{G}^{(\mathcal{C})}$	El subgrafo inducido de $\mathcal{G}$ usando los nodos en $\mathcal{C} \subseteq \mathcal{V}$.
$\mathbf{L}^{(\mathcal{G})}$	La matriz laplaciana de un grafo $\mathcal{G}$. Véa también: matriz laplaciana, grafo.
$\mathbf{L}^{(\mathcal{C})}$	La matriz laplaciana del grafo inducido $\mathcal{G}^{(\mathcal{C})}$. Véa también: matriz laplaciana, grafo.
$\mathcal{N}^{(i)}$	El entorno de un nodo i en un grafo $\mathcal{G}$. Véa también: entorno, grafo.
$d^{(i)}$	El grado ponderado $d^{(i)} := \sum_{i' \in \mathcal{N}^{(i)}} A_{i,i'}$ de un nodo i en un grafo $\mathcal{G}$. Véa también: grafo.

$d_{\max}^{(\mathcal{G})}$	El máximo grado de nodo ponderado de un grafo $\mathcal{G}$. Véa también: máximo, grado de nodo, grafo.
$\mathcal{D}^{(i)}$	El conjunto de datos local $\mathcal{D}^{(i)}$ llevado por el nodo $i \in \mathcal{V}$ de un red FL. Véa también: conjunto de datos local, red FL.
m_i	El número de puntos de datos (es decir, tamaño de la muestra) contenido en el conjunto de datos local $\mathcal{D}^{(i)}$ en el nodo $i \in \mathcal{V}$. Véa también: punto de datos, tamaño de la muestra, conjunto de datos local.
$\mathbf{x}^{(i,r)}$	Las atributos del punto de datos r-ésimo en el conjunto de datos local $\mathcal{D}^{(i)}$. Véa también: atributo, punto de datos, conjunto de datos local.
$y^{(i,r)}$	La etiqueta del punto de datos r-ésimo en el conjunto de datos local $\mathcal{D}^{(i)}$. Véa también: etiqueta, punto de datos, conjunto de datos local.
$\mathbf{w}^{(i)}$	Los parámetros del modelo locales del dispositivo i dentro de un red FL. Véa también: parámetros del modelo, dispositivo, red FL.
$L_i(\mathbf{w})$	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. Véa también: función de pérdida, dispositivo, parámetros del modelo.

$L^{(\mathrm{d})}\left(\mathbf{x}, h\big(\mathbf{x}\big), h'\big(\mathbf{x}\big)\right)$	La pérdida incurrida por un hipótesis h' en un punto de datos con atributos $\mathbf{x}$ y etiqueta $h\big(\mathbf{x}\big)$ que se obtiene de otro hipótesis. Véa también: pérdida, hipótesis, punto de datos, atributo, etiqueta.
$\mathrm{stack}\big\{\mathbf{w}^{(i)}\big\}_{i=1}^{n}$	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 al apilar verticalmente los parámetros del modelo locales $\mathbf{w}^{(i)} \in \mathbb{R}^d$. Véa también: parámetros del modelo.

Herramientas

aplicación Utilizamos el término aplicación como sinónimo de función.

Véa también: función.

convergence TBD ,

convex optimization TBD

determinante El determinante $\det(\mathbf{A})$ de una matriz cuadrada $\mathbf{A} = \left(\mathbf{a}^{(1)}, \ldots, \mathbf{a}^{(d)}\right) \in \mathbb{R}^{d \times d}$ es una funcion de sus columnas $\mathbf{a}^{(1)}, \ldots, \mathbf{a}^{(d)} \in \mathbb{R}^d$, que satisface [9]

- Normalización:

$$\det(\mathbf{I}) = 1$$

- Multilinealidad:

$$\begin{aligned}\det\left(\mathbf{a}^{(1)}, \ldots, \alpha\mathbf{u} + \beta\mathbf{v}, \ldots, \mathbf{a}^{(d)}\right) = \alpha \det\left(\mathbf{a}^{(1)}, \ldots, \mathbf{u}, \ldots, \mathbf{a}^{(d)}\right) \\ + \beta \det\left(\mathbf{a}^{(1)}, \ldots, \mathbf{v}, \ldots, \mathbf{a}^{(d)}\right)\end{aligned}$$

- Antisimetría:

$$\det\left(\ldots, \mathbf{a}^{(j)}, \ldots, \mathbf{a}^{(j')}, \ldots\right) = -\det\left(\ldots, \mathbf{a}^{(j')}, \ldots, \mathbf{a}^{(j)}, \ldots\right).$$

Podemos interpretar una matriz $\mathbf{A}$ como una transformación lineal en $\mathbb{R}^d$. El determinante $\det(\mathbf{A})$ caracteriza cómo (y en qué orientación) se alteran los volúmenes en $\mathbb{R}^d$ por esta transformación [3], [10]. En

particular, $\det(\mathbf{A}) > 0$ preserva la orientación, $\det(\mathbf{A}) < 0$ la invierte, y $\det(\mathbf{A}) = 0$ colapsa el volumen completamente, lo que indica que $\mathbf{A}$ no es invertible. El determinante también satisface $\det(\mathbf{AB}) = \det(\mathbf{A}) \cdot \det(\mathbf{B})$, y si $\mathbf{A}$ es diagonalizable con valor propio $\lambda_1, \ldots, \lambda_d$, entonces $\det(\mathbf{A}) = \prod_{j=1}^{d} \lambda_j$ [11]. Para los casos especiales $d = 2$ (2D) y $d = 3$ (3D), el determinante puede interpretarse como un área o volumen orientado definido por los vectores columna de $\mathbf{A}$.

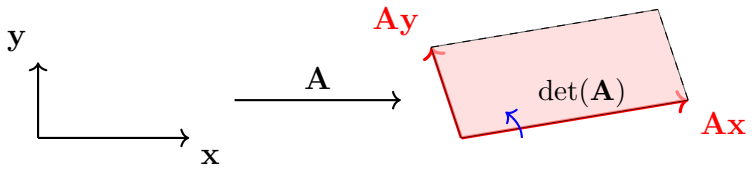


Fig. 1. Podemos interpretar una matriz cuadrada $\mathbf{A}$ como una transformación lineal de $\mathbb{R}^d$ en sí misma. El determinante $\det(\mathbf{A})$ caracteriza cómo esta transformación altera un volumen orientado.

Vea tambien: valor propio, matriz inversa.

espacio vectorial Un espacio vectorial $\mathcal{V}$ (también llamado espacio lineal) es una colección de elementos, llamados vectores, junto con las siguientes dos operaciones: 1) suma (denotada $\mathbf{v} + \mathbf{w}$) de dos vectores $\mathbf{v}, \mathbf{w}$; y 2) multiplicación (denotada $c \cdot \mathbf{v}$) de un vector $\mathbf{v}$ con un escalar c que pertenece a algún campo numérico (con una elección típica para este campo siendo $\mathbb{R}$). La propiedad definitoria de un espacio vectorial es que está cerrado bajo estas operaciones:

- Si $\mathbf{v}, \mathbf{w} \in \mathcal{V}$, entonces $\mathbf{v} + \mathbf{w} \in \mathcal{V}$.

- Si $\mathbf{v} \in \mathcal{V}$ y $c \in \mathbb{R}$, entonces $c\mathbf{v} \in \mathcal{V}$.
- En particular, $\mathbf{0} \in \mathcal{V}$.

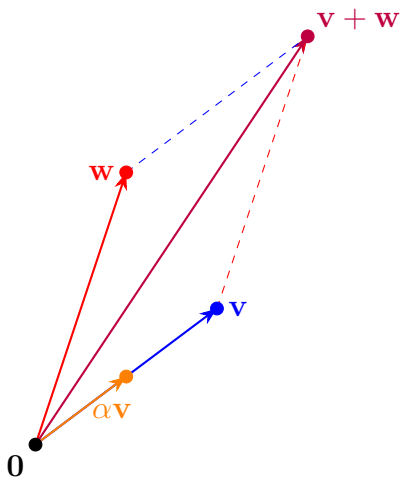


Fig. 2. Un espacio vectorial $\mathcal{V}$ es una colección de vectores tal que escalar y sumar estos siempre produce otro vector en $\mathcal{V}$.

Un ejemplo común de un espacio vectorial es el espacio euclidiano $\mathbb{R}^n$, que es ampliamente utilizado en aprendizaje automático para representar conjuntos de datos. También podemos usar $\mathbb{R}^n$ para representar, ya sea exactamente o aproximadamente, el espacio de hipótesis utilizado por un método de aprendizaje automático. Otro ejemplo de un espacio vectorial, que está naturalmente asociado con cada espacio de probabilidad $\mathcal{P} = \big(\Omega, \mathcal{R}, \mathbb{P}(\cdot)\big)$, es la colección de todas las variables aleatorias con valores reales $x : \Omega \to \mathbb{R}$ [1], [12].

Véa también: vector, espacio euclidiano, modelo lineal, aplicación lineal.

función Una función entre dos conjuntos $\mathcal{U}$ y $\mathcal{V}$ asigna a cada elemento $u \in \mathcal{U}$ exactamente un elemento $v \in \mathcal{V}$ [2]. Esto se escribe como $f : \mathcal{U} \to \mathcal{V}$, donde $\mathcal{U}$ es el dominio y $\mathcal{V}$ el codominio de f. Es decir, una función f define una salida única $f(u) \in \mathcal{V}$ para cada entrada $u \in \mathcal{U}$ (see Fig. 3).

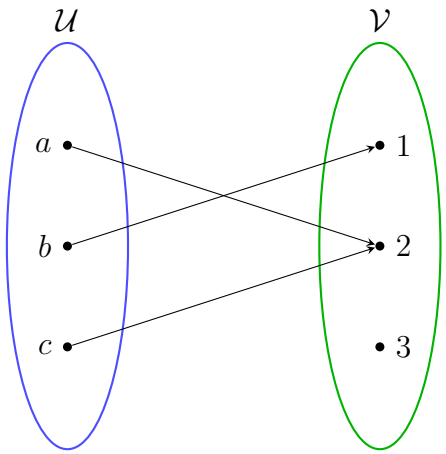


Fig. 3. Una función $f\colon \{a, b, c\} \to \{1, 2, 3\}$ mapea cada elemento del dominio a exactamente un elemento del codominio.

función característica La función característica de una RV real x es la función [6, Sección 26]

$$\phi_x(t) := \mathbb{E}\exp(jtx) \text{ con } j = \sqrt{-1}.$$

La función característica determina de forma única la distribución de probabilidad de x.

Véa también: distribución de probabilidad, RV.

hessiana Para una función $f : \mathbb{R}^d \to \mathbb{R}$ dos veces continuamente diferenciable en un punto $\mathbf{x}$, la hessiana es la matriz cuadrada de derivadas parciales

de segundo orden. Se denota por $\nabla^2 f(\mathbf{w}) = \left[\frac{\partial^2 f(\mathbf{w})}{\partial w_i \partial w_j}\right]_{i,j=1}^{d}$. Informa sobre la curvatura local de una función y se utiliza ampliamente en optimización.

matriz Una matriz de tamaño $m \times d$ es un arreglo bidimensional de números, denotado como

$$\mathbf{A} = \begin{bmatrix} A_{1,1} & A_{1,2} & \dots & A_{1,d} \\ A_{2,1} & A_{2,2} & \dots & A_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ A_{m,1} & A_{m,2} & \dots & A_{m,d} \end{bmatrix} \in \mathbb{R}^{m \times d}.$$

Aquí, $A_{r,j}$ denota la entrada de la matriz en la fila r-ésima y la columna j-ésima. Las matrices son representaciones útiles de diversos objetos matemáticos [13]:

- **Sistemas de Ecuaciones Lineales.** Podemos usar una matriz para representar un sistema de ecuaciones lineales

  $$\begin{pmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \quad \text{de forma compacta como} \quad \mathbf{Aw} = \mathbf{y}.$$

  Un ejemplo importante de sistemas de ecuaciones lineales son las condiciones de optimalidad para los parametros del modelo dentro de regresión lineal.

- **Aplicacion lineal.** Consideremos un espacio vectorial de dimensión d, $\mathcal{U}$, y un espacio vectorial de dimensión m, $\mathcal{V}$. Si fijamos una base $\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(d)}$ para $\mathcal{U}$ y una base $\mathbf{v}^{(1)}, \dots, \mathbf{v}^{(m)}$ para $\mathcal{V}$,

cada matriz $\mathbf{A} \in \mathbb{R}^{m \times d}$ define naturalmente una aplicación lineal $\alpha : \mathcal{U} \to \mathcal{V}$:

$$\mathbf{u}^{(j)} \mapsto \sum_{r=1}^{m} A_{r,j} \mathbf{v}^{(r)}.$$

- **Conjunto de datos** Podemos usar una matriz para representar un conjunto de datos: cada fila corresponde a un único punto de datos, y cada columna corresponde a un atributo específico atributo o etiqueta de un punto de datos.

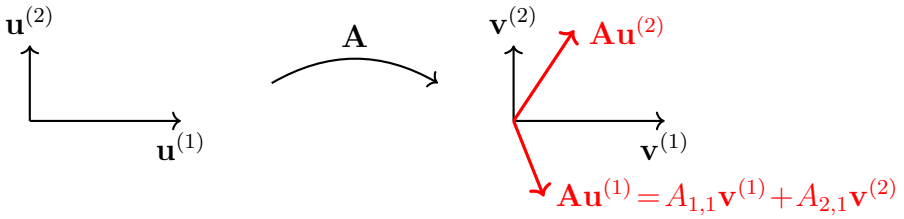


Fig. 4. Una matriz $\mathbf{A}$ define una aplicacion lineal entre dos espacios vectoriales

Véa también: aplicación lineal, conjunto de datos, modelo lineal, parámetros del modelo, regresión lineal, espacio vectorial

Método de Newton El método de Newton es un método de optimización iterativo para encontrar mínimos o máximos locales de una diferenciable función objetivo $f(\mathbf{w})$. Al igual que los métodos de gradiente, el método de Newton calcula una nueva estimación $\widehat{\mathbf{w}}_{k+1}$ optimizando una aproximación local de $f(\mathbf{w})$ alrededor de la estimación actual $\widehat{\mathbf{w}}_k$. En contraste con los métodos de gradiente, que utilizan el gradiente para construir una aproximación lineal local, el método de Newton utiliza la matriz hessiana para construir una aproximación cuadrática local.

En particular, partiendo de una estimación inicial $\widehat{\mathbf{w}}_0$, el método de Newton actualiza iterativamente la estimación de acuerdo con

$$\widehat{\mathbf{w}}_{k+1} = \widehat{\mathbf{w}}_k - \left(\nabla^2 f\left(\widehat{\mathbf{w}}_k\right)\right)^{-1} \nabla f\left(\widehat{\mathbf{w}}_k\right), \text{ para } k = 0, 1, \ldots.$$

Aquí, $\nabla f\left(\widehat{\mathbf{w}}_k\right)$ es el gradiente, y $\nabla^2 f(\mathbf{w}^{(k)})$ es el hessiana de la función objetivo f. Dado que usar una función cuadrática como aproximación local es más preciso que usar una función lineal (que es un caso particular de función cuadrática), el método de Newton tiende a converger más rápido que los métodos de gradiente. Sin embargo, esta mayor rapidez de convergencia conlleva un incremento en la complejidad computacional de las iteraciones. De hecho, cada iteración del método de Newton requiere la inversión de la hessiana.

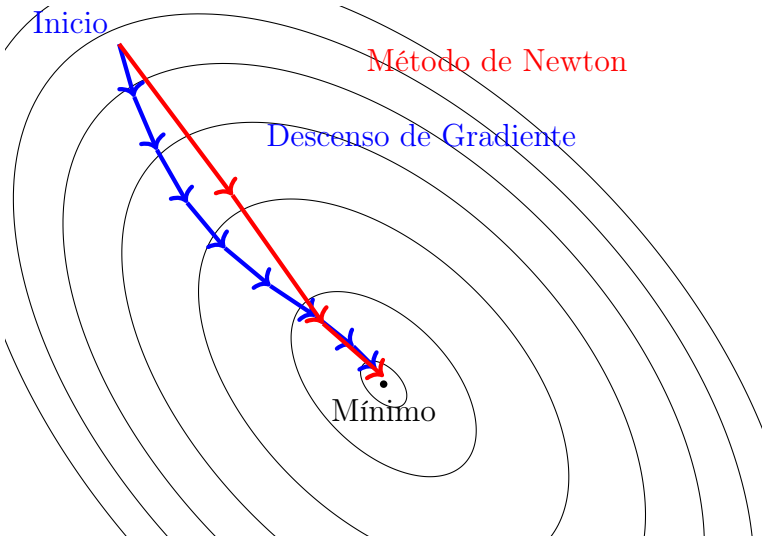


Fig. 5. Comparación de las trayectorias del Descenso de Gradiente (azul) y del Método de Newton (rojo) hacia el mínimo de una función de pérdida.

Véase también: método de optimización, descenso por gradiente (GD),

hessiana, gradiente.

problema de optimización Un problema de optimización es una estructura matemática que consiste en una función objetivo $f : \mathcal{U} \to \mathcal{V}$ definida sobre una variable de optimización $\mathbf{w} \in \mathcal{U}$, junto con un conjunto factible $\mathcal{W} \subseteq \mathcal{U}$. Se asume que el codominio $\mathcal{V}$ está ordenado, lo que significa que para cualesquiera dos elementos $\mathbf{a}, \mathbf{b} \in \mathcal{V}$, podemos determinar si $\mathbf{a} < \mathbf{b}$, $\mathbf{a} = \mathbf{b}$ o $\mathbf{a} > \mathbf{b}$. El objetivo de la optimización es encontrar aquellos valores $\mathbf{w} \in \mathcal{W}$ para los cuales el valor de la función objetivo $f(\mathbf{w})$ es extremo —es decir, mínimo o máximo [14–16].

proceso estocástico Un proceso estocástico es una colección de variables aleatorias definidas sobre un mismo espacio de probabilidad. Estas variables aleatorias están indexadas por el tiempo o el espacio, y se utilizan para modelar fenómenos aleatorios que evolucionan en el tiempo (por ejemplo, ruido en sensores o series temporales financieras). Grafos aleatorios, como el modelo estocástico de bloques (SBM) o el grafo de Erd, son otro ejemplo de procesos estocásticos. Estos procesos utilizan pares de nodos en un grafo como conjunto de índices. También podemos usar procesos estocásticos para representar algoritmo estocástico como descenso por gradiente estocástico.

Véa también: descenso de gradiente estocástico (SGD), incertidumbre, modelo probabilístico, RV, modelo estocástico de bloques (SBM).

vector Un vector es un elemento de un espacio vectorial. En el contexto del aprendizaje automático, un ejemplo particularmente importante de

espacios vectoriales es el espacio euclidiano $\mathbb{R}^d$, donde $d \in \mathbb{N}$ representa la dimensión (finita) del espacio. Un vector $\mathbf{x} \in \mathbb{R}^d$ puede representarse como una lista o arreglo unidimensional de números reales, es decir, $x_1, \ldots, x_d$ con $x_j \in \mathbb{R}$ para $j = 1, \ldots, d$. El valor x_j es la j-ésima entrada del vector $\mathbf{x}$. También puede ser útil considerar un vector $\mathbf{x} \in \mathbb{R}^d$ como una función que asigna a cada índice $j \in \{1, \ldots, d\}$ un valor $x_j \in \mathbb{R}$, es decir, $\mathbf{x} : j \mapsto x_j$. Esta perspectiva resulta particularmente útil en el estudio de los metodos de kernel.

2,-1,3,0,-2,1

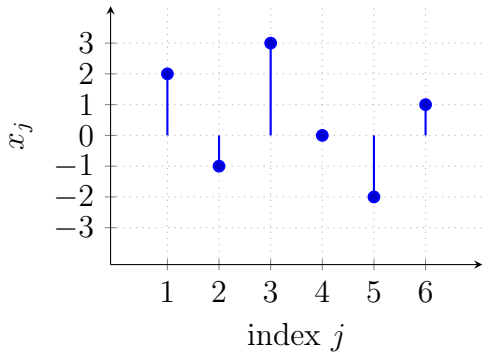


Fig. 6. Dos representaciones equivalentes de un vector $\mathbf{x} = \big(2, -1, 3, 0, -2, 1\big)^T \in \mathbb{R}^6$: como un arreglo numérico (izquierda) y como una función $j \mapsto x_j$ (derecha).

Vea tambien: espacio euclidiano, espacio vectorial, aplicación lineal, función, método de kernels

Conceptos de Aprendizaje Automático

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

Vea también: media, algoritmo, agrupamiento rígido, punto de datos, conjunto de datos, cluster.

agrupamiento Los métodos de agrupamiento descomponen un conjunto dado de puntos de datos en pocos subconjuntos, que se denominan clústers. Cada clúster está compuesto por puntos de dastos que son más similares entre sí que con respecto a los puntos de datos fuera del clúster. Los diferentes métodos de agrupamiento utilizan distintas medidas de similitud entre puntos de datos 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 el modelo de mezcla gaussiana representa un clúster mediante una distribución normal multivariante.

Vea también: punto de datos, cluster, k-means, atributo, media, agrupamiento suave, modelo de mezcla gaussiana (GMM), distribución normal multivariante.

agrupamiento basado en densidad para aplicaciones espaciales con ruido (DBSCAN)

DBSCAN es un algoritmo de agrupamiento para puntos de datos caracterizados por vectores de atributos numéricos. Al igual que k-means y agrupamiento suave mediante modelo de mezcla gaussiana, DBSCAN utiliza las distancias euclidianas entre los vectores de atributos para determinar los clusteres. Sin embargo, a diferencia de k-means y modelo de mezcla gaussiana, DBSCAN emplea una noción diferente de similitud entre puntos de datos. DBSCAN considera que dos puntos de datos son similares si están conectados a través de una secuencia (camino) de puntos de datos intermedios cercanos. Por lo tanto, DBSCAN puede considerar que dos puntos de datos son similares (y, por lo tanto, pertenecen al mismo cluster) incluso si sus vectores de atributos tienen una gran distancia euclidiana.

Vea tambien: agrupamiento, k-means, agrupamiento suave, GMM, cluster.

agrupamiento basado en flujo El agrupamiento basado en flujo agrupa los nodos de un grafo no dirigido aplicando el algoritmo de k-means sobre vectores de atributos específicos para cada nodo. Estos vectores de atributos construyen a partir de flujos de red entre nodos fuente y destino seleccionados cuidadosamente [17].

Vea también: agrupamiento, grafo, k-means, vector de atributos

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

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

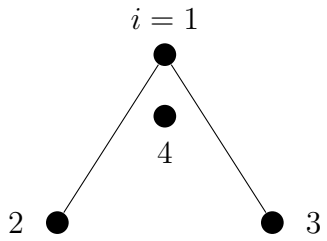
Aquí, $\|\mathbf{u}\|_p := \left(\sum_{j=1}^{d} |u_j|^p\right)^{1/p}$ denota la norma-p (para $p \geq 1$). Resulta que muchos de los vectores óptimos $\widehat{\mathbf{w}}^{(1)}, \ldots, \widehat{\mathbf{w}}^{(m)}$ coinciden. Un clúster consiste entonces en aquellos puntos de datos $r \in \{1, \ldots, m\}$ con valores idénticos de $\widehat{\mathbf{w}}^{(r)}$ [18, 19].

Vea también: conjunto de datos, convexo, agrupamiento, norma, cluster, punto de datos.

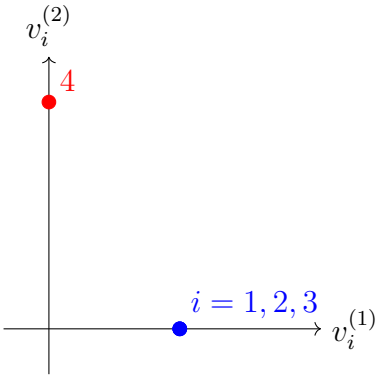
agrupamiento en grafos El agrupamiento en grafos tiene como objetivo agrupar puntos de datos que están representados como nodos de un grafo $\mathcal{G}$. Las aristas del $\mathcal{G}$ representan similitudes por pares entre los puntos de datos. En algunos casos, es posible cuantificar el grado de estas similitudes mediante un peso de arista [17, 20].

Vea también: agrupamiento, grafo, punto de datos, peso de arista

agrupamiento espectral El agrupamiento espectral es una instancia particular del agrupamiento en grafos, es decir, agrupa puntos de datos representados como los nodos $i = 1, \ldots, n$ de un grafo $\mathcal{G}$. El agrupamiento espectral utiliza los vectores propios de la matriz laplaciana $\mathbf{L}^{(\mathcal{G})}$ para construir vectores de atributos $\mathbf{x}^{(i)} \in \mathbb{R}^d$ para cada nodo (es decir, para cada punto de datos) $i = 1, \ldots, n$. Podemos utilizar estos vectores de atributos como entrada para métodos de agrupammiento en el espacio euclidiano,como k-means o agrupamiento suave mediante modelo de mezcla gaussiana. Mas o menos, los vectores de atributos de los nodos que pertenecen a un subconjunto bien conectado (o agrupamiento) de nodos en $\mathcal{G}$ estan ubicados cerca en el espacio euclidiano $\mathbb{R}^d$ (Vea la Figura 7).



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



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

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

Fig. 7. **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 descomposición en valores propios **Abajo.** Izquierda: Un diagrama de dispersión de los puntos de datos usando los vectores de atributos $\mathbf{x}^{(i)} = \left(v_i^{(1)}, v_i^{(2)}\right)^T$. Derecha: Dos vectores propios $\mathbf{v}^{(1)}, \mathbf{v}^{(2)} \in \mathbb{R}^d$ correspondientes al valor propio $\lambda = 0$ de la matriz laplaciana $\mathbf{L}^{(\mathcal{G})}$.

.

Vea también: agrupamiento, agrupamiento en grafos, punto de datos, grafo, vector propio, matriz laplaciana, vector de atributos, espacio euclidiano, k-means, agrupamiento suave, GMM, descomposición en valores propios (EVD), diagrama de dispersión, valor propio,

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

Vea también: agrupamiento, punto de datos, cluster, k-means.

agrupamiento suave El agrupamiento suave se refiere a la tarea de dividir un conjunto dado de puntos de datos en (pocos) clusteres superpuestos. Cada punto de datos se asigna a varios clusteres 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 puntos de datos como realizaciónes iid de un modelo de mezcla gaussiana. De este modo, se obtiene una elección natural para el grado de pertenencia como la probabilidad condicional de que un puntos de datos pertenezca a un componente específico de la mezcla.

Vea tambien: agrupamiento, punto de datos, cluster, grado de pertenencia, independientes e idénticamente distribuidos (i.i.d.), realización, GMM, probabilidad.

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 [21]. Por ejemplo, un algoritmo para entrenar un modelo lineal describe explícitamente cómo transformar un conjunto de entrenamiento dado en parametros del modelo a través de una secuencia de pasos de gradiente. Esta caracterización informal puede formalizarse rigurosamente mediante diferentes modelos matemáticos [22]. 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 [7, 23, 24]. Crucialmente, un algoritmo abarca más que solo un mapeo de entrada a salida; también incluye los pasos computacionales intermedios $s_1, \ldots, s_T$.

Vea también: modelo lineal, conjunto de entrenamiento, parámetros del modelo, paso de gradiente, modelo

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 [25, 26]. A diferencia de un algoritmo clásico, que se ejecuta en un solo dispositivo, un algoritmo distribuido 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:

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

Cadad 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, donde una red de dispositivos colaboran para entrenar un modelo personalizado por dispositivo..

Vea también: algoritmo, algoritmo, dispositivo, aprendizaje federado (FL)

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 [27, 28]. 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 aprendizaje automático es el descenso por gradiente en línea, que actualiza los modelos de parametro conforme llegan nuevos puntos de datos.
Vea también: algoritmo, datos, ML, descenso por gradiente en línea (online GD), parámetros del modelo, punto de datos

algoritmo estocástico Un algoritmo estocástico utiliza un mecanismo aleatorio durante su ejecución. Por ejemplo, descenso por gradiente estocástico utiliza un subconjunto seleccionado aleatoriamente de puntos de datos para calcular una aproximación del gradiente de una funcion objetivo. Podemos representar un algoritmo estocástico mediante un proceso estocástico, es decir, la secuencia de ejecución posible es el conjunto de resultados posibles de un experimento aleatorio [7], [23], [24].
Véa también: estocástico, algoritmo, SGD, punto de datos, gradiente, función objetivo, método de optimización, métodos de gradiente.

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 minimo error de reconstrucción [8].
Vea también: mapa de atributos, atributo, mínimo.

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

básico mediante el uso de un modelo de probabilidad para los puntos de datos. El modelo de probabilidad de PPCA reduce la tarea de reducción de dimensionalidad a un problema de estimación que se puede resolver utilizando métodos de esperanza-maximización.

Vea también: análisis de componentes principales (PCA), modelo probabilístico, punto de datos, EM.

aplicación lineal Una aplicación lineal $f : \mathbb{R}^n \to \mathbb{R}^m$ es una función que satisface las propiedades de aditividad: $f(\mathbf{x} + \mathbf{y}) = f(\mathbf{x}) + f(\mathbf{y})$, y homogeneidad: $f(c\mathbf{x}) = cf(\mathbf{x})$ para todos los vectores $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ y escalares $c \in \mathbb{R}$. En particular, $f(\mathbf{0}) = \mathbf{0}$. Toda aplicación lineal puede representarse como una multiplicación matricial $f(\mathbf{x}) = \mathbf{A}\mathbf{x}$ para alguna matriz $\mathbf{A} \in \mathbb{R}^{m \times n}$. El conjunto de aplicaciones lineales con valores reales para una dimensión n dada constituye un modelo lineal, el cual se utiliza en muchos métodos de aprendizaje automático.

Véa también: modelo lineal, regresión lineal, PCA, vector de atributos.

aprendizaje automático (ML) El aprendizaje automático tiene como objetivo predecir una etiqueta a partir de los 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 [8, 29]. Una formulación precisa de este principio es la minimización empírica del riesgo (ERM). Diferentes métodos de ML se obtienen de distintas elecciones de diseño para los puntos de datos (sus atributos y etiquetas), el modelo, y la función de pérdida [8, Cap. 3].

Vea también: etiqueta, atributos, punto de datos, hipótesis, espacio de hipótesis, modelo, función de pérdida, minimización empírica del riesgo (ERM)

aprendizaje automático explicable (explainable ML) El aprendizaje automático explicable (explainable ML) consiste en métodos de aprendizaje automático que tienen como objetivo complementar cada predicción con una explicacion explícita de cómo se ha obtenido dicha predicción.La construcción de una explicacion explícita puede no ser necesaria si el método de aprendizaje automático utiliza un modelo lo suficientemente simple (o interpretable) [30].

Vea también: predicción, explicacion, ML, modelo.

aprendizaje de atributos Consideremos una aplicación de aprendizaje automático con puntos de datos caracterizados por atributos crudos $\mathbf{x} \in \mathcal{X}$. El aprendizaje de atributos se refiere a la tarea de aprender un mapeo

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

que recibe como entrada el atributo $\mathbf{x} \in \mathcal{X}$ 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,

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

Vea también: ML, punto de datos, atributo, espacio de atributos, PCA, mínimo

aprendizaje en línea Algunos métodos de aprendizaje automático están diseñados para procesar datos de forma secuencial, actualizando sus parametros del modelo a medida que nuevos puntos de datos se vuelven disponibles—uno a la vez. Un ejemplo típico son los datos de series temporales, como las temperaturas minimas y maximas diarias registradas por una estación meteorológica de FMI. Estos valores forman una secuencia cronológica de observaciones. En el aprendizaje en línea, la hipótesis (o sus parametros del modelo) se refina incrementalmente con cada nuevo punto de datos observado, sin volver a visitar los datos

anteriores.

Véa también: ML, datos, parámetros del modelo, punto de datos, Instituto Meteorológico de Finlandia (FMI), hipótesis, GD en linea, algoritmo en línea.

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

Vea también: modelo local, dispositivo, FL, suposición de agrupamiento, red FL, cluster, conjunto de datos local, conjunto de entrenamiento, modelo, minimización de variación total generalizada (GTVMin), parámetros del modelo.

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 conjuntos de datos locales que están relacionados por una estructura de red intrínsica.

Vea también: modelo, conjunto de datos local

aprendizaje federado horizontal (horizontal FL) El aprendizaje feder-

ado horizontal utiliza conjunto de datos locales constituidos por diferentes puntos de datos, pero emplea los mismos atributos para caracterizarlos [31]. 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 los los mismos atributos (por ejemplo, temperatura diaria o presión atmosférica).

Vea también: conjunto de datos local, punto de datos, atributos

aprendizaje federado vertical (VFL) El aprendizaje federado vertical (VFL) se refiere a aplicaciones de aprendizaje federado en las que los distintos dispositivos tienen acceso a diferentes atributos del mismo conjunto de puntos de datos [32]. Formalmente, el conjunto de datos global subyacente es

$$\mathcal{D}^{(\text{global})} := \left\{ \left(\mathbf{x}^{(1)}, y^{(1)} \right), \ldots, \left(\mathbf{x}^{(m)}, y^{(m)} \right) \right\}.$$

Denotamos por $\mathbf{x}^{(r)} = \left(x_1^{(r)}, \ldots, x_{d'}^{(r)} \right)^T$, para $r = 1, \ldots, m$, los vectores de atributos completos de los puntos de datos. Cada dispositivo$i \in \mathcal{V}$ observa solo un subconjunto $\mathcal{F}^{(i)} \subseteq \{1, \ldots, d'\}$ de atributos, lo que da lugar a un conjunto de datos local $\mathcal{D}^{(i)}$ con vectores de atributos

$$\mathbf{x}^{(i,r)} = \left(x_{j_1}^{(r)}, \ldots, x_{j_d}^{(r)} \right)^T.$$

Algunos de los dispositivos también pueden tener acceso a las etiquetas $y^{(r)}$, para $r = 1, \ldots, m$, del conjunto de datos global. Una posible

aplicación del VFL es permitir la colaboración entre distintos proveedores de atención médica. Cada proveedor recopila distintos tipos de mediciones, como análisis de sangre, electrocardiogramas y radiografías de tórax, para los mismos pacientes. Otra aplicación es un sistema nacional de seguro social, donde los registros de salud, los indicadores financieros, el comportamiento del consumidor y los datos de movilidad son recolectados por diferentes instituciones. El VFL permite el aprendizaje conjunto entre estas entidades mientras se mantienen niveles bien definidos de protección de la privacidad.

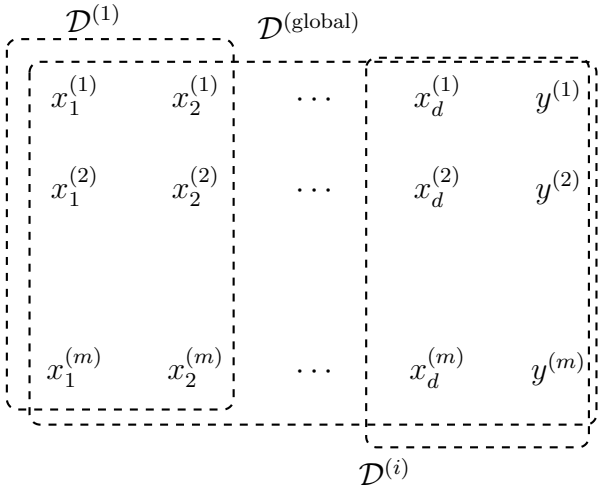


Fig. 8. El VFL utiliza conjuntos de datos locales que se derivan de los puntos de datos de un conjunto de datos global común. Los conjuntos de datos locales difieren en la selección de atributos utilizadas para caracterizar los puntos de datos.

Vea tambien: FL, dispositivo, atributo, punto de datos, conjunto de

datos, vector de atributos, conjunto de datos local, etiqueta, datos, protección de la privacidad.

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

Vea también: ML, modelo, datos.

aprendizaje multitarea El aprendizaje multitarea tiene como objetivo aprovechar las relaciones entre diferentes tareas de aprendizaje. Considere dos tareas de aprendizaje 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.

Vea también: tarea de aprendizaje, conjunto de datos, red profunda, pesos

aprendizaje por refuerzo (RL) RL se refiere a un entorno de aprendizaje en linea donde solo se puede evaluar la utilidad de una sola hipótesis (es decir, una elección de parametro del modelo) en cada paso temporal t. En particular, los métodos de RL aplican la hipótesis actual $h^{(t)}$ al vector de atributos $\mathbf{x}^{(t)}$ del nuevo punto de datos recibido. La utilidad de la predicción resultante $h^{(t)}(\mathbf{x}^{(t)})$ se cuantifica mediante una señal de recompensa $r^{(t)}$.

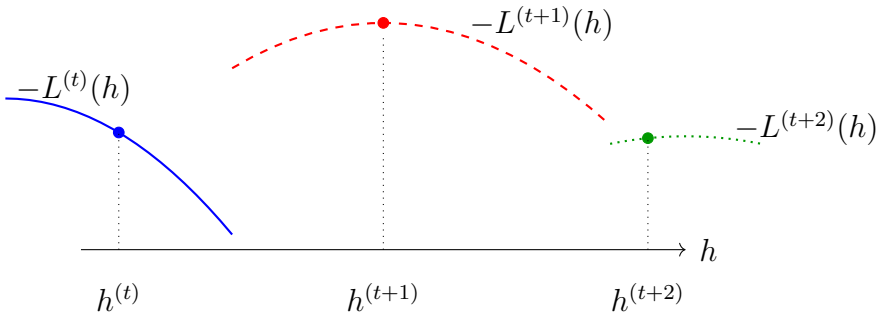


Fig. 9. Tres pasos temporales consecutivos $t, t+1, t+2$ con las correspondientes funciones de perdida $L^{(t)}, L^{(t+1)}, L^{(t+2)}$. Durante el paso temporal t, un método de RL puede evaluar la funcion de perdida solo para una hipótesis específica $h^{(t)}$, resultando en la señal de recompensa $r^{(t)} = -L^{(t)}(h^{(t)})$.

En general, la recompensa también depende de las predicciones previas $h^{(t')}\left(\mathbf{x}^{(t')}\right)$ para $t' < t$. El objetivo del RL es aprender $h^{(t)}$, para cada paso temporal t, de modo que la recompensa acumulada (posiblemente descontada) sea maximizada [8], [33].

Véa también: función de pérdida, recompensa, ML.

aprendizaje semi-supervisado (SSL) El aprendizaje semi-supervisado (SSL) utiliza puntos de datos no etiquetados para apoyar el aprendizaje de una hipótesis a partir de puntos de datos etiquetados [34]. Este enfoque es particularmente útil para aplicaciones de aprendizaje automático que ofrecen una gran cantidad de puntos de datos no etiquetados, pero solo un número limitado de puntos de datos etiquetados.

Vea también: punto de datos, hipótesis, punto de dato etiquetado, ML, punto de datos

ARIMA TBD

arrepentimiento (regret) El arrepentimiento de una hipótesis h en relación con otra hipótesis h', que sirve como referencia, es la diferencia entre la perdida incurrida por h y la perdida incurrida por h' [28]. La hipótesis de referencia h' también se denomina experto.

Vea también: hipótesis, referencia (baseline), pérdida, experto

aspectos computacionales Por aspectos computacionales de un método de aprendizaje automático, nos referimos principalmente a los recursos computacionales requeridos para su implementación. Por ejemplo, si un método de aprendizaje automático 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.

Vea también: ML,ERM, paso de gradiente, parámetros del modelo, GD

aspectos estadísticos Por aspectos estadísticos de un método de aprendizaje automático, nos referimos a las propiedades de la distribución de probabilidad de su salida bajo un modelo de probabilidad para los datos introducidos en el método.

Vea también: ML, distribución de probabilidad, modelo probabilístico, datos

ataque Un ataque a un sistema de aprendizaje automático se refiere a una acción intencional—ya sea activa o pasiva—que compromete la integri-

dad, disponibilidad o confidencialidad del sistema. Los ataques activos implican la alteración de componentes como los conjuntos de datos (por medio de envenenamiento de datos) o de los enlaces de comunicación entre dispositivos en un entorno de aprendizaje federado. Los ataques pasivos, como los ataques a la privacidad, buscan inferir atributos sensibles sin modificar el sistema. Según su objetivo, distinguimos entre ataques de denegación de servicio, ataques de puerta trasera y ataques a la privacidad.

Véa también: envenenamiento de datos, ataque a la privacidad, atributo sensible, ataque de denegación de servicio (DoS), puerta trasera (backdoor).

ataque a la privacidad Un ataque a la privacidad en un sistema de aprendizaje automático tiene como objetivo inferir atributos sensibles de individuos mediante el aprovechamiento del acceso parcial a un modelo de aprendizaje automático ya entrenado. Una forma de ataque a la privacidad es la inversion de modelo.

Véa también: ataque, ML, atributo sensible, inversion de modelo, inteligencia artificial confiable (IA confiable), reglamento general de protección de datos (RGPD).

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 puntos de datos típicos.

Vea también: envenenamiento de datos, modelo, punto de datos

atención Algunas aplicaciones de aprendizaje automático implican puntos de datos compuestos por unidades más pequeñas, conocidas como *tokens*. Por ejemplo, una oración se compone de palabras, una imagen de parches de píxeles y una red de nodos. En la práctica, los tokens dentro de un solo punto de datos típicamente no son independientes entre sí, sino que cada token presta atención a ciertos otros tokens. Los modelos de probabilidad proporcionan un marco fundamentado para representar y analizar tales dependencias [35]. Los mecanismos de atención utilizan un enfoque más directo, sin referencia explícita a un modelo de probabilidad. La idea es representar la relación entre dos tokens i y i' mediante una funcion parametrizada $f^{(\mathbf{w})}(i, i')$, donde los parametros $\mathbf{w}$ se aprenden mediante una variante de ERM. Los mecanismos de atención prácticos difieren tanto en la elección precisa del modelo de atención $f^{(\mathbf{w})}(i, i')$ como en la variante específica de ERM empleada para aprender los parametros $\mathbf{w}$. Una familia ampliamente utilizada de mecanismos de atención define los parámetros $\mathbf{w}$ en términos de dos vectores asociados a cada token i: un vector de consulta $\mathbf{q}^{(i)}$ y un vector clave $\mathbf{k}^{(i')}$. Para un token dado i con consulta $\mathbf{q}^{(i)}$ y otro token i' con clave $\mathbf{k}^{(i')}$, la cantidad $\left(\mathbf{q}^{(i)}\right)^{\top}\mathbf{k}^{(i')}$ cuantifica hasta qué punto el token i presta atención a (o depende de) i'.

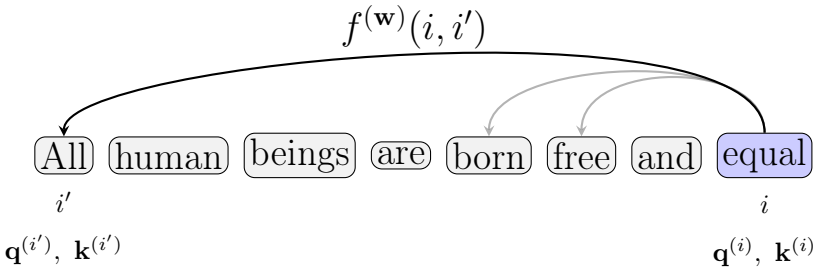


Fig. 10. Los mecanismos de atención aprenden una función parametrizada $f^{(\mathbf{w})}(i, i')$ que mide cuánto atención presta el token i al token i'. Una construcción ampliamente utilizada de $f^{(\mathbf{w})}(i, i')$ emplea vectores de consulta y clave, $\mathbf{q}^{(i)}$ y $\mathbf{k}^{(i)}$, asignados a cada token i [36].

Véa también: función.

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" [37], [38], [39].

Vea también: punto de datos

atributo sensible El aprendizaje automático 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 puntos de datos. Qué se

considera atributo sensible depende del dominio de aplicación y debe definirse explícitamente.

Vea también: ML, hipótesis, etiqueta, punto de datos, atributo,

aumentación de datos Los métodos de aumentación de datos añaden puntos de datos sintéticos a un conjunto existente de punto de datos. Estos puntos de datos sintéticos se obtienen mediante perturbaciones (por ejemplo, añadir ruido a mediciones físicas) o transformaciones (por ejemplo, rotaciones de imágenes) de los puntos de datos originales. Estas perturbaciones y transformaciones son tales que los puntos de datos sintéticos resultantes deben tener la misma etiqueta. Como ejemplo, una imagen de un gato rotada sigue siendo una imagen de un gato, aunque sus vectores de atributos (obtenidos al apilar las intensidades de color de los píxeles) sean muy diferentes (ver Figura 11). La aumentación de datos puede ser una forma eficiente de regulación.

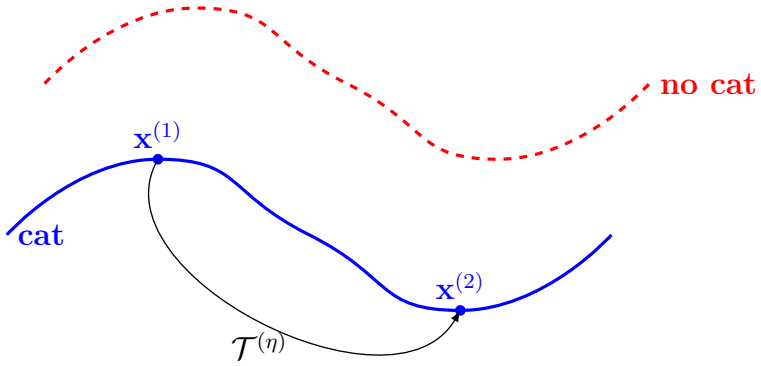


Fig. 11. La aumentación de datos aprovecha las simetrías intrínsecas de los puntos de datos en algún espacio de atributos $\mathcal{X}$. Podemos representar una simetría mediante un operador $\mathcal{T}^{(\eta)} : \mathcal{X} \to \mathcal{X}$, parametrizado por algún número $\eta \in \mathbb{R}$. Por 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 $\mathbf{x}^{(2)} = \mathcal{T}^{(\eta)}\big(\mathbf{x}^{(1)}\big)$ debete tener la misma etiqueta $y^{(2)} = y^{(1)}$ que un punto de datos con vector de atributos $\mathbf{x}^{(1)}$.

Vea también: datos, punto de datos, etiqueta, vector de atributos, regularización, espacio de atributos.

autoencoder Un autoencoder es un método de aprendizaje automático que aprende simultáneamente un mapeo codificador $h(\cdot) \in \mathcal{H}$ y un mapeo decodificador $h^*(\cdot) \in \mathcal{H}^*$. Es una instancia de minimización empírica del riesgo (ERM) que utiliza una perdida calculada a partir del error de reconstrucción $\mathbf{x} - h^*\big(h\big(\mathbf{x}\big)\big)$.

Vea también: ML, ERM, pérdida

bagging El bagging (o agregación por bootstrap) es una técnica genérica

para mejorar (la robustez de) un método de aprendizaje automático 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 predicciónes individuales de cada hipótesis separada. Para mapas de hipótesis que entregan valores numéricos de etiqueta, esta agregación podría implementarse calculando el promedio de las predicciónes individuales.

Vea también: ML, bootstrap, conjunto de datos, hipótesis, etiqueta, punto de datos, predicción.

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, 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 [40].

Vea también: recompensa, distribución de probabilidad, media, arrepentimiento (regret)

batchlearning En *batch learning* (también conocido como: aprendizaje *offline*), el modelo de aprendizaje automático se entrena sobre el conjunto de datos completo en una única iteración de entrenamiento, en lugar de actualizarlo de manera incremental conforme llegan los datos. Todos los datos disponibles se introducen en un algoritmo de aprendizaje, resultando en un modelo que puede realizar predicciónes. Dado que estos conjuntos de datos suelen ser grandes, el entrenamiento es computacionalmente costoso y consume mucho tiempo, por lo que normalmente se realiza *offline*. Después del entrenamiento, el modelo será estático y no se adaptará automáticamente a nuevos datos. Actualizar el modelo con nueva información requiere volver a entrenar completamente el modelo. Una vez que el modelo ha sido entrenado, se despliega en producción donde no puede ser actualizado. Entrenar un modelo puede tomar muchas horas, por lo que muchos modelos en entornos de producción se actualizan de manera cíclica en un calendario periódico cuando la distribución de los datos es estable. Por ejemplo, un equipo de análisis minorista podría reentrenar su modelo de predicción de demanda cada domingo utilizando los datos de ventas de la semana anterior para predecir la demanda de la semana siguiente. Si un sistema necesita actualizarse constantemente frente a datos que cambian rápidamente, como en la predicción de precios bursátiles, es necesaria una solución más adaptable como el aprendizaje en linea.

Véa también: modelo, conjunto de datos, lote, aprendizaje en línea.

boosting El boosting es un metodo de optimización iterativo para aprender una función de hipótesis precisa (o strong learner) combinando secuencialmente funciones de hipótesis menos precisas (conocidas como weak learners) [41, Cap. 10]. Por ejemplo, los weak learners suelen ser árboles de decisión poco profundos que se combinan para obtener un árbol de decisión profundo. El boosting puede entenderse como una generalización de los metodos de descenso por gradiente para ERM utilizando modelos paramétricos y funciones de perdida suaves [42]. Así como el descenso por gradiente actualiza iterativamente los parametros del modelo para reducir el rieso empírico, el boosting combina iterativamente (ej., por suma) funciones de hipótesis para reducir dicho riesgo empírico. Una variante ampliamente utilizada del enfoque general de boosting es el boosting gradiente, que emplea el gradiente de la función de perdida para combinar los weak learners [42].

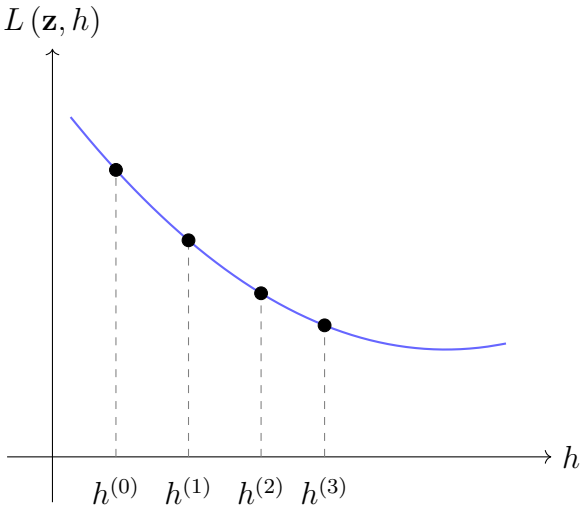


Fig. 12. Los métodos de boosting construyen una secuencia de funciones de hipótesis $h^{(0)}, h^{(1)}, \ldots$ que son cada vez mejores (es decir, incurren en menor perdida).

Véa también: hipótesis, aplicación, árbol de decisión, generalización, métodos de gradiente, ERM, modelo, suave, función de pérdida, GD, parámetros del modelo, riesgo empírico, gradiente, pérdida, paso de gradiente.

bootstrap Para el análisis de métodos de aprendizaje automático, es a menudo útil interpretar un conjunto dado de puntos de datos $\mathcal{D} = \left\{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}\right\}$ como realizaciónes de variables aleatorias iid 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 $p(\mathbf{z})$.

Vea también: ML, punto de datos, realización, RV, i.i.d.,distribución de probabilidad

bosque aleatorio Un bosque aleatorio es un conjunto (o ensamblaje) de diferentes árboles de decisiónes. Cada uno de estos árboles de decisiónes se obtiene al ajustar una copia perturbada del conjunto de datos original. Vea también: árbol de decisión, conjunto de datos.

brecha de generalización La diferencia entre el rendimiento de un modelo entrenado sobre el conjunto de entrenamiento y su rendimiento sobre otros puntos de datos (como los del conjunto de validacion).
Véa también: modelo, conjunto de entrenamiento, punto de datos, conjunto de validación, hipótesis, árbol de decisión, generalización, métodos de gradiente, ERM, suave, función de pérdida, GD, parámetros del modelo, riesgo empírico, gradiente, pérdida, paso de gradiente.

caracterización min-max de Courant–Fischer–Weyl Considere una matriz semi-definida positiva $\mathbf{Q} \in \mathbb{R}^{d \times d}$ con descomposición en valores propios (o descomposición espectral),

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

Aquí, utilizamos los valores 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 valores propios de $\mathbf{Q}$ como las soluciones a ciertos proble-

mas de optimización.

Vea también: psd, EVD, valor propio.

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.

Vea también: punto de datos, atributo

clasificación multi-etiqueta Los problemas y métodos de clasificación multi-etiqueta utilizan puntos de datos 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.

Vea también: clasificación, punto de datos, etiqueta

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 regiónes de decisión $\mathcal{R}_a$, para cada valor posible de etiqueta $a \in \mathcal{Y}$.

Vea también: hipótesis, etiqueta, espacio de etiquetas, predicción, clasificación, región de decisión

clasificador lineal Consideremos puntos de datos caracterizados por atributos numéricos $\mathbf{x} \in \mathbb{R}^d$ y una etiqueta $y \in \mathcal{Y}$ de algún espacio de etiqueta finito $\mathcal{Y}$. Un clasificador lineal se caracteriza por tener regiones de decision que están separadas por hiperplanos en $\mathbb{R}^d$ [8, Ch. 2].

Vea también: punto de datos, atributo, etiqueta, espacio de etiquetas, región de decisión

clúster Un clúster es un subconjunto de puntos de datos que son más similares entre sí que a los puntos de datos fuera del clúster. La medida cuantitativa de similitud entre los puntos de datos es una decisión de diseño. Si los puntos de datos se caracterizan por vectores de atributos euclidianos $\mathbf{x} \in \mathbb{R}^d$, podemos definir la similitud entre dos puntos de datos a través de la distancia euclidiana entre sus vectores de atributos. Un ejemplo de tal clúster es. 13.

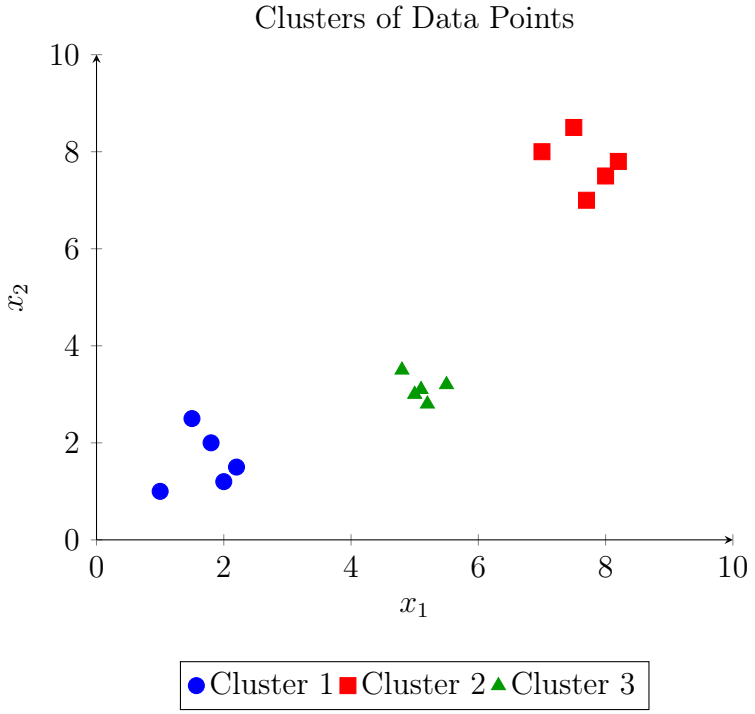


Fig. 13. Ilustración de tres clústers en un espacio de atributos bidimensional. Cada clúster agrupa puntos de datos que son más similares entre sí que con aquellos en otros clusters, basándose en la distancia euclidiana.

Vea también: punto de datos, vector de atributos, espacio de atributos.

condición de gradiente cero Considera el problema de optimización sin restricciones $\min_{\mathbf{w}\in\mathbb{R}^d} f(\mathbf{w})$ con una funcion de objetivo suave y convexa $f(\mathbf{w})$. 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,

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

Vea también: suave, convexo, función objetivo, gradiente.

conectividad algebraica La conectividad algebraica de un grafo no dirigido es el segundo valor propio más pequeño λ_2 de su matriz laplaciana. Un grafo es conexo si y solo si $\lambda_2 > 0$.

Véa también: grafo, valor propio, matriz laplaciana.

conjunto de datos Un conjunto de datos se refiere a una colección de puntos de datos. Estos puntos de datos contienen información sobre alguna cantidad de interés (o etiqueta) dentro de una aplicación de aprendizaje automático. Los métodos de aprendizaje automático utilizan conjuntos de datos para el entrenamiento de un 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 puntos de datos. De hecho, los puntos de datos pueden ser objetos físicos concretos (como humanos o animales) o objetos abstractos (como números). Como ejemplo, la Figura 14 muestra un conjunto de datos que consiste en vacas como punto de datos.

Fig. 14. La manada de vacas

Con frecuencia, un ingeniero de aprendizaje automático no tiene acceso

directo a un conjunto de datos. De hecho, acceder al conjunto de datos en la Figura 14 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 [43], [44], [45], [46]. Uno de los modelos de datos más adoptados es el modelo relacional, que organiza los datos en una tabla (o relación) [47], [43]. 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 aprendizaje automático pueden utilizar estos atributos como atributos y etiquetas del punto de datos.

Por ejemplo, la Tabla I muestra una representación del conjunto de datos en la Figura. 14. 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 aprendizaje automático, estos dominios de atributos se convierten en el espacio de atributos y el espacio de etiquetas.

TABLE I

A Relation (or Table) That Represents the Dataset in Fig. 14

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

Si bien el modelo relacional es útil para el estudio de muchas aplicaciones de aprendizaje automático, puede ser insuficiente para cumplir con los requisitos de 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 [48].

Vea también: punto de datos, etiqueta, ML, modelo, ERM, validación, datos, espacio de atributos, espacio de etiquetas, trustworthy AI

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 puntos de datos individuales, que se caracterizan por atributos y etiquetas. A diferencia de un único conjunto de datos utilizado en métodos de aprendizaje automático básicos, un conjunto de datos local también está relacionado con otros conjuntos de datos locales mediante diferentes nociones de similitud. Estas similitudes pueden surgir de modelos de probabilidad o de la infraestructura de comunicación y están codificadas en las aristas

de una red de aprendizaje federado.

Vea también: conjunto de datos, punto de datos, atributo, etiqueta, ML, modelo probabilístico, red FL.

conjunto de entrenamiento Un conjunto de entrenamiento es un conjunto de datos $\mathcal{D}$ que consiste en algunos puntos de datos usados en ERM para aprender una hipótesis $\hat{h}$. La perdida 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 aprendizaje automático e informa cómo mejorar el error de validación (por ejemplo, utilizando un espacio de hipótesis diferente o recopilando más puntos de datos) [8, Sec. 6.6].

Vea también: conjunto de datos, punto de datos, ERM, hipótesis, pérdida, error de entrenamiento, error de validación

conjunto de prueba Un conjunto de puntos de datos que no ha sido utilizado ni para entrenar un modelo (por ejemplo, mediante ERM) ni en un conjunto de validacion para elegir entre diferentes modelos.

Vea también: punto de datos, modelo, ERM, conjunto de validación, modelo

conjunto de validación Un conjunto de puntos de dastos usado para estimar el riesgo de una hipótesis $\hat{h}$ que ha sido aprendida mediante algún método de aprendizaje automático (por ejemplo, resolviendo ERM). La perdida promedio de $\hat{h}$ en el conjunto de validación se denomina error de validacion y puede utilizarse para diagnosticar un método de aprendizaje automático (vea [8, Sec. 6.6]). La comparación entre el

error de entrenamiento y el error de validacion puede proporcionar información sobre cómo mejorar el método de aprendizaje automático (como usar un espacio de hipótesis diferente).

Vea también: punto de datos, riesgo, hipótesis, ML, ERM, pérdida, validación, error de validación, error de entrenamiento, espacio de hipótesis

contraction operator An operator $\mathcal{F} : \mathbb{R}^d \to \mathbb{R}^d$ is a contraction if, for some $\kappa \in [0, 1)$,

$$\|\mathcal{F}\mathbf{w} - \mathcal{F}\mathbf{w}'\|_2 \leq \kappa \|\mathbf{w} - \mathbf{w}'\|_2 \text{ holds for any } \mathbf{w}, \mathbf{w}' \in \mathbb{R}^d.$$

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 [14]. Ilustramos un ejemplo de un conjunto convexo y una función convexa en la Figura 15.

Fig. 15. Izquierda: Un conjunto convexo $\mathcal{C} \subseteq \mathbb{R}^d$. Derecha: Una función convexa $f : \mathbb{R}^d \to \mathbb{R}$.

Vea también: espacio euclidiano.

cota superior de confianza (UCB) Considera una aplicación de aprendizaje automático 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 de probabilidad ampliamente utilizado para este tipo de problema de toma de decisiones secuencial es el entorno de bandidos multi-brazo estocásticos [40]. En este modelo, la recompensa $r^{(a)}$ se considera como la realización de una variable aleatoria 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 [40].

Vea también: ML, recompensa, modelo probabilístico, realización, RV, media, datos

covarianza La covarianza entre dos variables aleatorias reales x y y, definidas sobre un mismo espacio de probabilidad, mide su dependencia lineal. Se define como

$$\operatorname{cov}(x, y) = \mathbb{E}\big\{\big(x - \mathbb{E}\{x\}\big)\big(y - \mathbb{E}\{y\}\big)\big\}.$$

Una covarianza positiva indica que x y y tienden a aumentar juntos,

mientras que una covarianza negativa sugiere que uno tiende a aumentar cuando el otro disminuye. Si $\operatorname{cov}(x, y) = 0$, se dice que las RVs son no correlacionadas, aunque esto no implica necesariamente que sean estadísticamente independientes. Véa la Figura 16 para ilustraciones visuales.

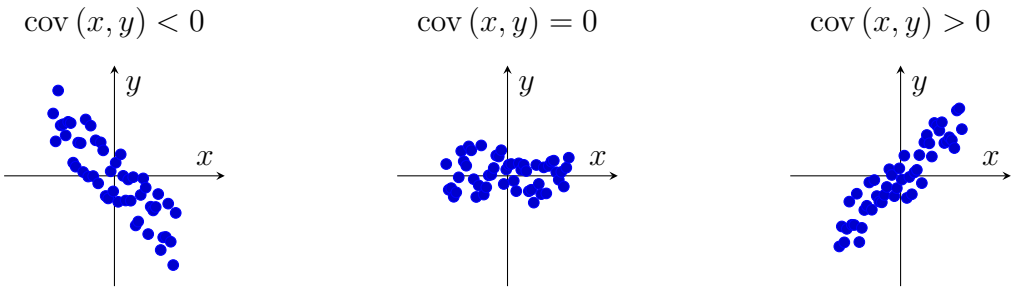


Fig. 16. Diagramas de dispersion ilustrando realizaciónes de tres modelos de probabilidad diferentes para dos variables aleatorias reales con diferentes valores de covarianza: negativo (izquierda), cero (centro), y positivo (derecha).

criterio de parada Muchos métodos de aprendizaje automático utilizan algoritmos iterativos que construyen una secuencia de parametros del modelo (como los pesos de un mapa lineal o los pesos de una red neuronal artificial). Estos parámetros (idealmente) convergen a una elección óptima para los parametros 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.

Vea tambien: ML, algoritmo, parámetros del modelo, pesos, ANN.

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 [47].

datos en red Los datos en red consisten en conjuntos de datos locales relacionados por alguna noción de similitud por pares. Podemos representar los datos en red utilizando un grafo cuyos nodos contienen conjuntos de datos locales y cuyas aristas codifican similitudes por pares. Un ejemplo de datos en red surge en las aplicaciones de aprenizaje federado donde los conjuntos de datos locales son generados por dispositivos distribuidos espacialmente.

Vea también: datos, conjunto de datos local, grafo, conjunto de datos local, FL, dispositivo

datos faltantes Considere un conjunto de datos constituido por puntos de datos recopilados a través de algún dispositivo físico. Debido a imperfecciones y fallas, algunos de los valores de atributos o etiquetas de los puntos ded atos podrían estar corruptos o simplemente faltar. La imputación de datos tiene como objetivo estimar estos valores faltantes [49]. Podemos interpretar la imputación de datos como un problema de aprendizaje automático donde la etiqueta de un punto de datos es es el valor del atributo corrupto.

Vea también: conjunto de datos, punto de datos, dispositivo, atributo,

etiqueta, punto de datos, Datos

descenso de gradiente estocástico (SGD) El descenso de gradiente estocástico es una variante del descenso por gradiente en la que se reemplaza el gradiente de la funcion 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 puntos de datos están almacenados en bases de datos distribuidas por todo el mundo). Para evaluar el gradiente del rieso empírico (como función de los parametros 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 puntos de datos 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 $\mathcal{B} \subseteq \{1, \ldots, m\}$ (Vea la figura 17). A estos puntos de datos seleccionados aleatoriamente se les denomina a menudo lote. El tamaño del lote, denotado $|\mathcal{B}|$ es un parámetro importante del descenso por gradiente estocástico. El descenso por gradiente estocástico con $|\mathcal{B}| > 1$ se conoce como mini-lote SGD [50].

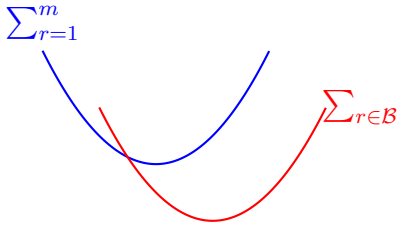


Fig. 17. El descenso por gradiente estocástico (descenso por gradiente) para ERM aproxima el gradiente $\sum_{r=1}^{m} \nabla_{\mathbf{w}} L\left(\mathbf{z}^{(r)}, \mathbf{w}\right)$ reemplazando la suma sobre todos los puntos de datos del conjunto de entrenamiento (indexados por $r = 1, \ldots, m$) con una suma sobre un subconjunto aleatorio $\mathcal{B} \subseteq \{1, \ldots, m\}$.

Vea tambien: GD, gradiente, función objetivo, modelo, ERM, conjunto de entrenamiento, punto de datos, riesgo empírico, parámetros del modelo, lote.

descenso por gradiente (GD) El desceso por gradiente es un método iterativo para encontrar el minimo 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 minimo 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 18 ilustra el efecto de un solo paso de gradiente (2).

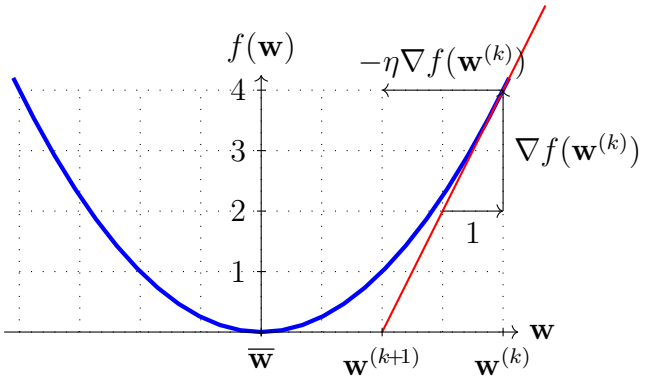


Fig. 18. Un paso de gradiente singular (2) hacia el minimizador $\overline{\mathbf{w}}$ de $f(\mathbf{w})$.

Vea también: mínimo, diferenciable, paso de gradiente, tamaño de paso, gradiente.

descenso por gradiente en línea (online GD) Considere un método de aprendizaje automático que aprende los parametros del modelo $\mathbf{w}$ a partir de un espacio de parametros $\mathcal{W} \subseteq \mathbb{R}^d$. El proceso de aprendizaje utiliza puntos de datos $\mathbf{z}^{(t)}$ que llegan en instantes de tiempo consecutivos $t = 1, 2, \ldots$. Interpretamos los puntos de datos $\mathbf{z}^{(t)}$ como copias iid de una variable aleatoria $\mathbf{z}$. 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 funcion objetivo para aprender los parametros del modelo $\mathbf{w}$. Sin embargo, este límite solo puede evaluarse si esperamos un tiempo infinito para recolectar todos los puntos de datos. Algunas aplicaciones de aprendizaje automático requieren métodos que aprendan en línea: tan pronto como llega un nuevo puntos de datos $\mathbf{z}^{(t)}$ en el tiempo t,

actualizamos los parametros 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 parametros del modelo actualizados $\mathbf{w}^{(t+1)}$ aumenten la perdida 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 parametros del modelo $\mathbf{w}^{(T+1)}$ entregados por el descenso por gradiente en línea tras observar T puntos de datos $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(T)}$ son al menos tan buenos como los entregados por cualquier otro método de aprendizaje [27, 51].

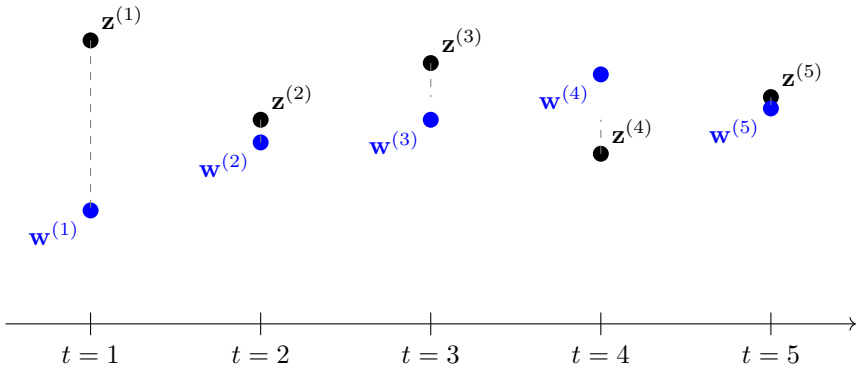


Fig. 19. Una instancia de descenso por gradiente en línea que actualiza los parametros del modelo $\mathbf{w}^{(t)}$ usando el punto de datos $\mathbf{z}^{(t)} = x^{(t)}$ que llega en el tiempo t. Esta instancia emplea la pérdida de error cuadrático $L\left(\mathbf{z}^{(t)}, w\right) = (x^{(t)} - w)^2$.

Vea también: ML, parámetros del modelo, espacio de parámetros, punto de datos, i.i.d., RV, riesgo, hipótesis, función objetivo, GD, paso de gradiente, pérdida, tasa de aprendizaje, pérdida de error cuadrático.

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 funcion objetivo de ERM es suave, no podemos usar el descenso por gradiente básico, ya ya que este no toma en cuenta las restricciones sobre la variable de optimización (es decir, los parámetros del modelo). El descenso por gradiente proyectado extiende el descenso por gradiente básico para controlar restricciones sobre la variable de optimización(es decir, los parámetros del modelo). Una sola iteración del descenso por gradiente proyectado consiste

primero en realizar un paso de gradiente y luego proyectar el resultado sobre el espacio de parámetros

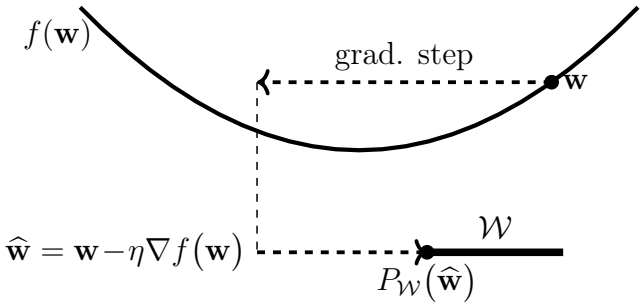


Fig. 20. GD proyectado amplía un paso de gradiente básico con una proyección de regreso al conjunto de restricciones $\mathcal{W}$.

.

Vea también:ERM, modelo, espacio de parámetros, función objetivo, suave, GD, parámetros del modelo, paso de gradiente

descenso por subgradiente El descenso por subgradiente es una generalización del descenso por gradiente 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 funcion objetivo. La funcion objetivo podría ser el riesgo empírico $\widehat{L}\big(h^{(\mathbf{w})} \big| \mathcal{D}\big)$ visto como una función de los parametors del modelo $\mathbf{w}$ que seleccionan una hipótesis $h^{(\mathbf{w})} \in \mathcal{H}$.

Vea también: subgradiente, generalización, GD, gradiente, función objetivo, riesgo empírico, parámetros del modelo, hipótesis.

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 $\mathbf{V} = \left(\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(d)}\right)$ son los vectores propios de la matriz $\mathbf{A}$. La matriz diagonal $\mathbf{\Lambda} = \operatorname{diag}\left\{\lambda_1, \ldots, \lambda_d\right\}$ contiene los valores propios λ_j correspondientes a los vectores propios $\mathbf{v}^{(j)}$. Es importante notar que esta descomposición solo existe si la matriz $\mathbf{A}$ es diagonalizable.

Vea también: vector propio, valor propio.

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.

desigualdad de concentración Una cota superior para la probabilidad de que una RV se desvíe más de una cantidad prescrita respecto a su esperanza [52].

Véa también: probabilidad, RV, esperanza.

diagrama de dispersión Un método de visualización que representa puntos de datos mediante marcadores en un plano bidimensional. La Figura 21 depicta un ejemplo de un diagrama de dispersión.

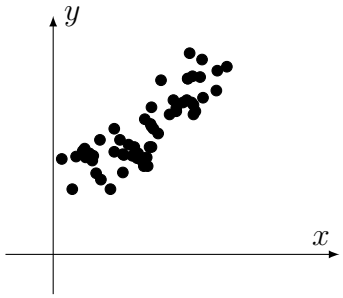


Fig. 21. Un diagrama de dispersión de algunos puntos de datos que representan las condiciones climáticas diarias en Finlandia. Cada punto de datos se caracteriza por su temperatura minima diaria x como atributo y su temperatura máxima diaria y como etiqueta. Las temperaturas se han medido en la estación meteorológica FMI de Helsinki Kaisaniemi durante el período 1.9.2024 - 28.10.2024.

.

Vea también: punto de datos, mínimo, atributo, máximo, etiqueta, FMI

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].

Vea también: gradiente

dimensión de Vapnik–Chervonenkis (dimensión VC) La dimensión VC es una medida ampliamente utilizada para cuantificar el tamaño de un espacio de hipótesis infinito. Remitimos a la literatura (véa [53]) para una definición precisa de la dimensión VC, así como una discusión sobre sus propiedades básicas y su uso en aprendizaje automatico.

Vea tambien: espacio de hipótesis, ML, dimensión efectiva.

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 parametros del modelo independientes y ajustables. Estos parametros pueden ser los coeficientes utilizados en un mapa lineal o los pesos y términos de sesgo de una red neuronal artificial.

Vea también: espacio de hipótesis, parámetros del modelo, parámetros, pesos, ANN

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

Vea también: FL, datos en red, red FL, hipótesis, modelo local

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

Vea también: datos, ML, punto de datos, modelo.

distribución de probabilidad Para analizar métodos de aprendizaje automático, puede ser útil interpretar los puntos de datos como realizaciónes iid de una variable aleatoria. Las propiedades típicas de tales puntos de datos están gobernadas por la distribución de probabilidad de

esta variable aleatoria. La distribución de probabilidad de una variable aleatoria binaria $y \in \{0, 1\}$ se especifica completamente mediante las probabilidades $\mathbb{P}(y = 0)$ y $\mathbb{P}(y = 1) = 1 - \mathbb{P}(y = 0)$. La distribución de probabilidad de una variable aleatoria con valores reales $x \in \mathbb{R}$ puede especificarse mediante una funcion de densidad de probabilidad $p(x)$ tal que $\mathbb{P}(x \in [a, b]) \approx p(a)|b - a|$. En el caso más general, una distribución de probabilidad se define mediante una medida de probabilidad [6, 54]. Vea también: ML, punto de datos, i.i.d., realización, RV, probabilidad, función de densidad de probabilidad (pdf)

distribución normal multivariante La distribución normal multivariante, denotada como $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, es un modelo de probabilidad fundamental para vectores de atributos numéricos de dimensión fija d. Define una familia de distribución de probabilidad sobre variables aleatorias vectoriales $\mathbf{x} \in \mathbb{R}^d$ [7], [54], [55]. Cada distribución de esta familia está completamente determinada por su vector de media $\boldsymbol{\mu} \in \mathbb{R}^d$ y su matriz de covarianza $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$. Cuando la matriz de covarianza $\boldsymbol{\Sigma}$ es invertible, la correspondiente distribución de probabilidad se caracteriza por la siguiente pdf:

$$p(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^d \det(\boldsymbol{\Sigma})}} \exp\left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right].$$

Nótese que esta pdf solo está definida cuando $\boldsymbol{\Sigma}$ es invertible. Más generalmente, cualquier variable aleatoria $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ admite la siguiente representación:

$$\mathbf{x} = \mathbf{A}\mathbf{z} + \boldsymbol{\mu}$$

donde $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ es un vector normal estandard, y $\mathbf{A} \in \mathbb{R}^{d \times d}$ satisface

$\mathbf{A}\mathbf{A}^\top = \mathbf{\Sigma}$. Esta representación sigue siendo válida incluso si $\mathbf{\Sigma}$ es singular, en cuyo caso $\mathbf{A}$ no es de rango completo [56, Ch. 23]. La familia de distribuciones normales multivariantes es excepcional entre los modelos de probabilidad para cantidades numéricas, al menos por las siguientes razones. Primero, esta familia es cerrada bajo transformaciones afines, es decir,

$$\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma}) \Rightarrow \mathbf{B}\mathbf{x}+\mathbf{c} \sim \mathcal{N}\big(\mathbf{B}\boldsymbol{\mu} + \mathbf{c}, \mathbf{B}\mathbf{\Sigma}\mathbf{B}^T\big).$$

Segundo, la distribución de probabilidad $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ maximiza la entropia diferencial entre todas las distribuciones con la misma matriz de covarianza $\mathbf{\Sigma}$ [57].

Véa también: modelo probabilístico, distribución de probabilidad, vector normal estándar, entropía diferencial, VA gaussiana.

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 [57].

Vea también: distribución de probabilidad

embudo de privacidad El embudo de privacidad es un método para aprender atributos amigables con la privacidad de los puntos de datos [58].

Vea también: atributo, punto de datos

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

Vea también: vecinos

entropía La entropía cuantifica la incertidumbre o imprevisibilidad asociada a una variable aleatoria [57]. Para una variable aleatoria discreta x que toma valores en un conjunto finito $\mathcal{S} = \{x_1, \ldots, x_n\}$ con función de masa de probabilidad $p_i := \mathbb{P}(x = x_i)$, la entropía se define como

$$H(x) := -\sum_{i=1}^{n} p_i \log p_i.$$

La entropía se maximiza cuando todos los resultados son igualmente probables, y se minimiza (es decir, es cero) cuando el resultado es determinista. Una generalización del concepto de entropía para variables aleatorias continuas es la entropía diferencial.

Véa también: incertidumbre, modelo probabilístico.

entropía diferencial Para una variable aleatoria con valores reales $\mathbf{x} \in \mathbb{R}^d$ con función de densidad de probabilidad $p(x)$, la entropía diferencial se define como [57]

$$h(\mathbf{x}) := -\int p(\mathbf{x}) \log p(\mathbf{x})\, d\mathbf{x}.$$

La entropía diferencial puede ser negativa y carece de algunas propiedades de la entropía para variables aleatorias discretas, como la invariancia ante cambios de variable [57]. Entre todas las variables aleatorias con media $\boldsymbol{\mu}$ y matriz de covarianza $\boldsymbol{\Sigma}$ dadas, $h(\mathbf{x})$ se maximiza cuando $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

Véa también: RV, pdf, media, matriz de covarianza, incertidumbre, modelo probabilístico.

envenenamiento de datos El envenenamiento de datos se refiere a la manipulación (o fabricación) intencionada de puntos de datos con el

fin de influir en el entrenamiento de un modelo de aprendizaje automático [59], [60]. Los ataques de envenenamiento de datos pueden adoptar diversas formas, entre las cuales se incluyen:

- Puerta trasera: Consiste en implantar disparadores (triggers) en los datos de entrenamiento, de manera que el modelo entrenado se comporte normalmente con vectores de atributos típicos, pero clasifique erróneamente un vector de atributos que contiene un patrón de disparo.
- Ataque de denegación de servicio(DDoS): Buscan degradar el rendimiento general del modelo entrenado mediante la inyección de ejemplos mal etiquetados o adversarios para impedir un aprendizaje efectivo.

El envenenamiento de datos es especialmente preocupante en entornos de aprendizaje automático descentralizados o distribuidos (como en aprendizaje federado), donde no es posible verificar centralmente los datos de entrenamiento.
Véa también: ataque, backdoor, ataque de denegación de servicio, trustworthy AI.

epoch Una *época* (epoch) representa un recorrido completo de todo el conjunto de entrenamiento a través de algún algoritmo de aprendizaje. Es cuando un modelo ha procesado cada punto de datos en el conjunto de entrenamiento una vez. Entrenar un modelo normalmente requiere múltiples *épocas*, ya que cada iteración permite al modelo refinar los parametros y mejorar las predicciónes. El número de *épocas* es algo

predefinido por el usuario, siendo este un hiperparámetro esencial, pues constituye un factor crucial para determinar cómo el modelo generalizará a datos no vistos. Muy pocas *épocas* resultarán en subajuste; por el contrario, demasiadas *épocas* pueden llevar a sobrejuste.

epígrafo El epígrafo de una función real $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ es el conjunto de puntos que se encuentran sobre o en el grafo de la función:

$$\text{epi}(f) = \{(\mathbf{x}, t) \in \mathbb{R}^n \times \mathbb{R} \,|\, f(\mathbf{x}) \leq t\}\,.$$

Una funcion es convexa si y solo si su epígrafo es un conjunto convexo [14], [61].

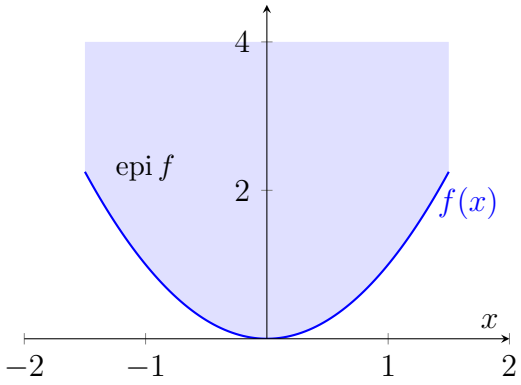


Fig. 22. Epígrafo de la función $f(x) = x^2$ (área sombreada).

Véa también: función, grafo, convexo.

error cuadrático medio de estimación (MSEE) Consideremos un método de aprendizaje automático que aprende parametros del modelo $\widehat{\mathbf{w}}$ a partir de un conjunto de datos $\mathcal{D}$. Si interpretamos los puntos de datos en $\mathcal{D}$ como realizaciónes iid de una variable aleatoria $\mathbf{z}$, 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 expectativa $\mathbb{E}\{\|\Delta\mathbf{w}\|^2\}$ del cuadrado de la norma euclidiana del error de estimación [62, 63].

Vea también: ML, parámetros del modelo, conjunto de datos, punto de datos, realización, error de estimación, distribución de probabilidad, esperanza, norma

error de entrenamiento La perdida promedio de una hipótesis al predecir las etiquetas de los puntos de datos en un conjunto de entrenamiento. A veces también nos referimos al error de entrenamiento como la perdida promedio mínimo que se logra mediante una solución de ERM.

Vea también: pérdida, hipótesis, etiqueta, punto de datos, conjunto de entrenamiento, ERM

error de estimación Consideremos varios puntos de datos, 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 aprendizaje 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 hipótesis se determinan mediante parametros del modelo $\mathbf{w}$, podemos definir el error

de estimación como $\Delta\mathbf{w} = \widehat{\mathbf{w}} - \overline{\mathbf{w}}$ [41, 63].

Vea también: punto de datos, vector de atributos, etiqueta, hipótesis, ML, ERM, espacio de hipótesis, parámetros del modelo

error de validación Consideremos una hipótesis $\hat{h}$ obtenida por algún método de aprendizaje automático, por ejemplo, utilizando ERM en un conjunto de entrenamiento. La perdida promedio de $\hat{h}$ en un conjunto de validación, que es diferente del cojunto de entrenamiento, se denomina error de validación.

Vea también: hipótesis, ML, ERM, conjunto de entrenamiento, pérdida, conjunto de validación

espacio de atributos El espacio de atributos de una determinada aplicación o método de aprendizaje automático está constituido por todos los valores posibles que puede tomar el vector de atributos de un punto de datos.

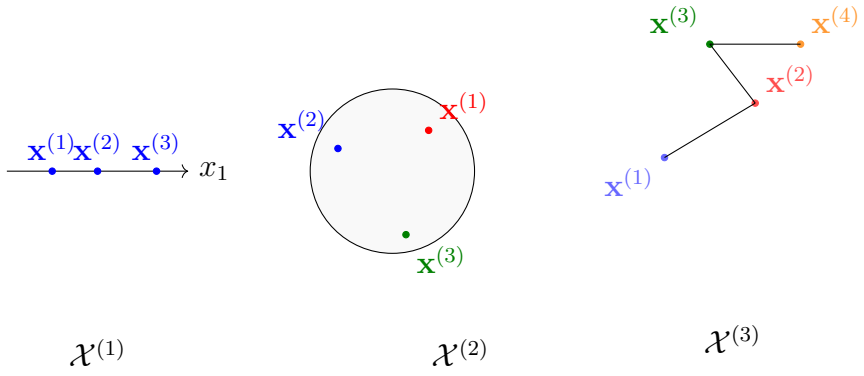


Fig. 23. Tres espacios de atributos distintos: un espacio lineal $\mathcal{X}^{(1)} = \mathbb{R}$, un conjunto convexo acotado $\mathcal{X}^{(2)} \subseteq \mathbb{R}^2$, y un espacio discreto $\mathcal{X}^{(3)}$ cuyos elementos son nodos de un grafo no dirigido.

Para puntos de datos descritos por un número fijo d de atributos numéricos, una elección común para el espacio de atributos es el espacio euclidiano $\mathbb{R}^d$. Sin embargo, la mera presencia de d atributos numéricos no implica que $\mathbb{R}^d$ sea la representación más adecuada del espacio de atributos. De hecho, los atributos numéricos podrían asignarse a los puntos de datos de forma mayormente arbitraria o aleatoria, lo que da lugar a puntos de datos dispersos al azar en $\mathbb{R}^d$ sin una estructura geométrica significativa. Los métodos de aprendizaje de atributos intentan aprender una transformación de los atributos originales (potencialmente no numéricos) que permita una disposición más significativa de los puntos de datos en $\mathbb{R}^d$.

Véa también: vector de atributos, espacio euclidiano.

espacio de etiquetas Consideremos una aplicación de aprendizaje automático que involucra puntos de datos caracterizados por atributos y etiquetas. El espacio de 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 a menudo utilizan el espacio de etiquetas $\mathcal{Y} = \mathbb{R}$. Los métodos de clasificación binaria utilizan un espacio de etiquetas 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"$\}$.

Vea también: ML, punto de datos, atributo, etiqueta, Regresión, clasificación

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 $\mathbf{u}^T\mathbf{v}$.

Vea también: espacio euclidiano.

espacio de hipótesis Un espacio de hipótesis es un modelo matemático que caracteriza la capacidad de aprendizaje de un método de aprendizaje automático. El objetivo de dicho método es aprender una mapa de hipótesis que asigne los atributos de un punto de datos a una prediccion de su etiqueta. Dado un recurso computacional finito, un método práctico de aprendizaje automático normalmente explora solo un subconjunto restringido de todos los posibles mapas desde el espacio de atributos al espacio de etiquetas. A este subconjunto restringido se le denomina espacio de hipótesis $\mathcal{H}$ asociado al método de aprendizaje automático. Para el análisis de un método dado de aprendizaje automático, la elección del espacio de hipótesis $\mathcal{H}$ no es única: cualquier superconjunto que contenga todos los mapas que el método puede aprender también es un espacio de hipótesis válido.

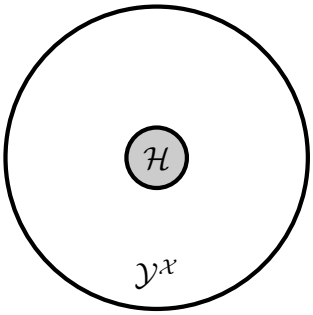


Fig. 24. El espacio de hipótesis $\mathcal{H}$ de un método de aprendizaje automático es un subconjunto (típicamente muy pequeño) del conjunto $\mathcal{Y}^{\mathcal{X}}$ (típicamente muy grande) de todos los posibles mapas desde el espacio de atributos $\mathcal{X}$ al espacio de etiquetas $\mathcal{Y}$.

Desde la perspectiva de la ingeniería de aprendizaje automático, por otro lado, el espacio de hipótesis $\mathcal{H}$ es una decisión de diseño en métodos basados en ERM. Esta decisión puede guiarse por los recursos computacionales disponibles y por los aspectos estadísticos. Por ejemplo, si es factible aplicar operaciones de matriz eficientes y existe una relación aproximadamente lineal entre atributos y etiquetas, un modelo lineal puede ser una elección útil para $\mathcal{H}$.

Véa también: hipótesis, modelo, aplicación, modelo lineal.

espacio de parámetros El espacio de parámetros $\mathcal{W}$ de un modelo de aprendizaje automático $\mathcal{H}$ es el conjunto de todas las elecciones factibles para los parametros del modelo (vea la Figura 25). Muchos métodos de aprendizaje automático importantes usan un modelo que está parametrizado por vectores del espacio euclidiano $\mathbb{R}^d$. Dos ejemplos

ampliamente utilizados de modelos parametrizados son los modelos lineales y las redes 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.

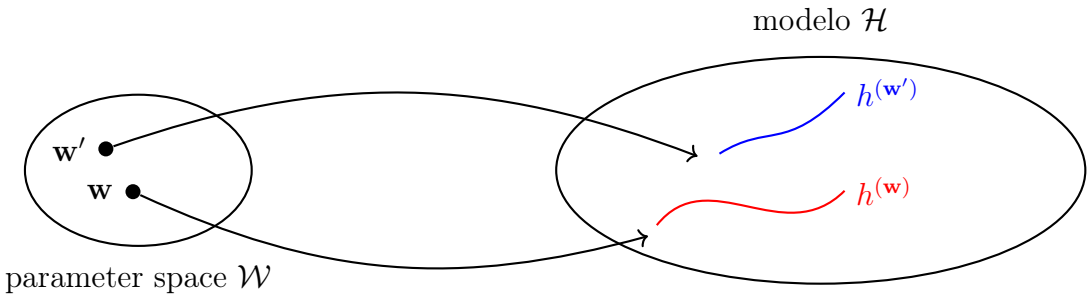


Fig. 25. El espacio de parámetros $\mathcal{W}$ de un modelo de aprendizaje automático $\mathcal{H}$ consiste en todas las elecciones factibles para los parametros del modelo. Cada elección $\mathbf{w}$ para los parametros del modelo seleccionan un mapa de hipótesis $h^{(\mathbf{w})} \in \mathcal{H}$.

Vea también: ML, modelo, parámetros del modelo, espacio euclidiano, modelo lineal, red profunda, norma, hipótesis.

espacio de probabilidad Un espacio de probabilidad es un modelo matemático de un proceso físico (un experimento aleatorio) con un resultado incierto. Formalmente, un espacio de probabilidad $\mathcal{P}$ es una tripleta $(\Omega, \mathcal{F}, P)$ donde:

- Ω es el espacio muestral que contiene todos los posibles resultados elementales del experimento aleatorio;

- $\mathcal{F}$ es una sigma-álgebra, es decir, una colección de subconjuntos de Ω (llamados eventos) que satisface ciertas propiedades de clausura bajo operaciones de conjuntos;
- P es una medida de probabilidad, una función que asigna una probabilidad $P(\mathcal{A}) \in [0, 1]$ a cada evento $\mathcal{A} \in \mathcal{F}$. Esta función debe satisfacer $P(\Omega) = 1$ y

$$P\left(\bigcup_{i=1}^{\infty} \mathcal{A}_i\right) = \sum_{i=1}^{\infty} P(\mathcal{A}_i)$$

para cualquier secuencia numerable de eventos mutuamente disjuntos $\mathcal{A}_1, \mathcal{A}_2, \ldots$ en $\mathcal{F}$.

Los espacios de probabilidad constituyen la base para definir variables aleatorias y razonar sobre la incertidumbre en aplicaciones de aprendizaje automático [6, 54, 64].

Véa también: probabilidad, modelo, RV, ML

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].

espacio muestral Un espacio muestral es el conjunto de todos los posibles outcomes de un experimento aleatorio [6, 7, 65, 66].

Véa también: espacio de probabilidad.

espacio nulo El espacio nulo de una matriz $\mathbf{A} \in \mathbb{R}^{d' \times d}$, denotado como

null($\mathbf{A}$), es el conjunto de todos los vectores $\mathbf{n} \in \mathbb{R}^d$ tales que

$$\mathbf{An} = \mathbf{0}.$$

Consideremos un método de aprendizaje de atributos que utiliza la matriz $\mathbf{A}$ para transformar un vector de atributos $\mathbf{x} \in \mathbb{R}^d$ de un punto de datos en un nuevo vector de atributos $\mathbf{z} = \mathbf{Ax} \in \mathbb{R}^{d'}$. El espacio nulo null($\mathbf{A}$) caracteriza todas las direcciones en el espacio de atributos original $\mathbb{R}^d$ a lo largo de las cuales la transformación $\mathbf{Ax}$ permanece invariante. En otras palabras, sumar cualquier vector del espacio nulo a un vector de atributos $\mathbf{x}$ no afecta a la representación transformada $\mathbf{z}$. Esta propiedad puede explotarse para imponer invariancias en las predicciónes (calculadas a partir de $\mathbf{Ax}$). La Fig. 26 ilustra una de estas invariancias. Muestra versiones rotadas de dos dígitos manuscritos, que aproximadamente se sitúan a lo largo de curvas unidimensionales en el espacio de atributos original. Estas curvas están alineadas con un vector de dirección $\mathbf{n} \in \mathbb{R}^d$. Para garantizar que el modelo entrenado sea invariante a dichas rotaciones, podemos elegir la matriz de transformación $\mathbf{A}$ de manera que $\mathbf{n} \in$ null($\mathbf{A}$). Esto asegura que $\mathbf{Ax}$, y por lo tanto la predicción resultante, sea aproximadamente insensible a rotaciones de la imagen de entrada.

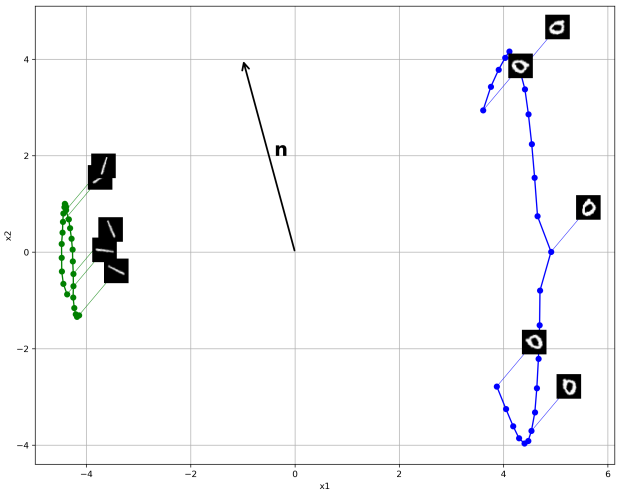


Fig. 26. Imágenes rotadas de dos dígitos manuscritos. Las rotaciones se alinean aproximadamente a lo largo de curvas lineales paralelas al vector **n**.

Véa también: matriz.

Demo en Python: click aquí

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) [67]. La Figure 27 muestra una señal temporal junto con su espectrograma.

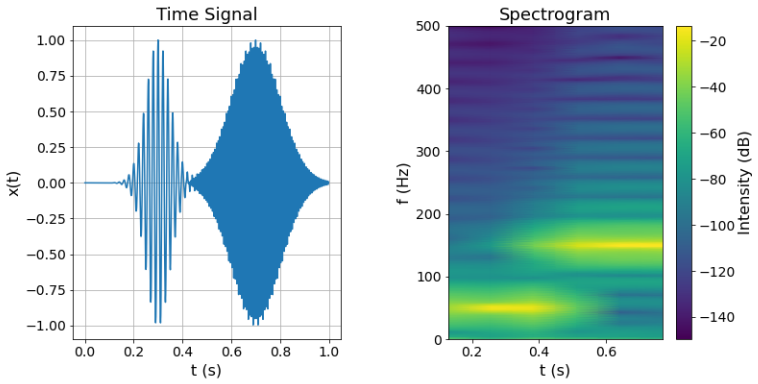


Fig. 27. Izquierda: una señal temporal compuesta por dos pulsos gaussianos modulados. Derecha: representación de intensidad de su espectrograma.

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 [68]. Conviene señalar que, además del espectrograma, existen otras representaciones alternativas para describir la distribución tiempo-frecuencia de la energía de una señal [69, 70].

Vea también: clasificación, red profunda

esperanza Consideremos un vector de atributos numérico $\mathbf{x} \in \mathbb{R}^d$ que interpretamos como la realización de una variable aleatoria 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, 6, 71]. Nótese que la esperanza solo está definida si esta integral existe, es decir, si la variable aleatoria es integrable. [2], [6], [71]. Fig. 28 ilustra la esperanza de una variable

aleatoria discreta x que toma valores solo de un conjunto finito.

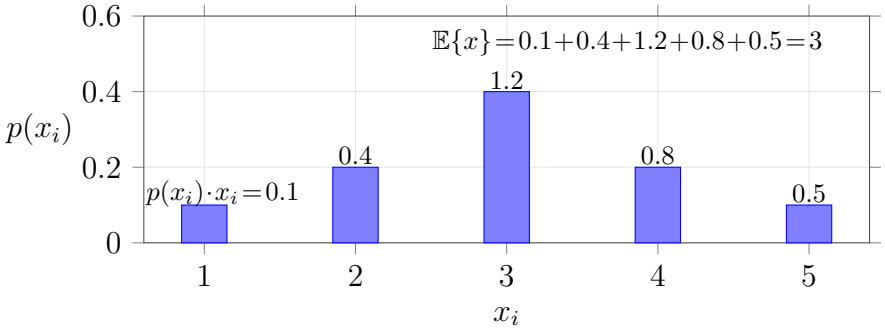


Fig. 28. La esperanza de una variable aleatoria discreta x se obtiene sumando sus posibles valores x_i,ponderados por la correspondiente probabilidad $p(x_i) = \mathbb{P}(x = x_i)$.

Vea también: vector de atributos, realización, RV, distribución de probabilidad, probabilidad.

esperanza-maximización (EM) Considera un modelo de probabilidad $\mathbb{P}(\mathbf{z}; \mathbf{w})$ para los puntos de datos $\mathcal{D}$ generados en alguna aplicación de aprendizaje Automático. El estimador de maxima verosimilitud para los parametros del modelo $\mathbf{w}$ se obtienen maximizando $\mathbb{P}(\mathcal{D}; \mathbf{w})$. Sin embargo, el problema de optimización resultante puede ser computacionalmente desafiante. El algoritmo de expectativa-maximización (EM) aproxima el estimador de maxima verosimilitud introduciendo una variable aleatoria latente $\mathbf{z}$ de modo que maximizar $\mathbb{P}(\mathcal{D}, \mathbf{z}; \mathbf{w})$ sea más fácil [41, 72, 73]. 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 $\max_{\mathbf{w}} \mathbb{P}(\mathcal{D}, \widehat{\mathbf{z}}; \mathbf{w})$. El punto clave es que la esperanza condicional $\widehat{\mathbf{z}}$ depende de los parametros 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 parametros del modelo. En la práctica, repetimos el cálculo de la esperanza condicional (es decir, el E-step) y la actualización de los parametros del modelo (es decir, the M-step) hasta que se cumpla algún criterio de parada.

Vea también: modelo probabilístico, punto de datos, ML, máxima verosimilitud, parámetros del modelo, RV, conjunto de datos, esperanza, criterio de parada.

estabilidad La estabilidad es una propiedad deseable de un método de aprendizaje automático $\mathcal{A}$ que mapea un conjunto de datos$\mathcal{D}$ (por ejemplo, un conjunto de entrenamiento) a una salida $\mathcal{A}(\mathcal{D})$, como parametros del modelo aprendidos o la predicción para un punto de datos específico. De forma intuitiva, $\mathcal{A}$ es estable si pequeños cambios en el conjunto de datos de entrada $\mathcal{D}$ provocan pequeños cambios en la salida $\mathcal{A}(\mathcal{D})$. Existen varias nociones formales de estabilidad que permiten establecer cotas sobre el error de generalización o el riesgo del método; véase [53, Cap. 13]. Para desarrollar la intuición, considérese los tres conjuntos de datos representados en la Fig. 29, cada uno de los cuales es igualmente probable bajo los mismos datos generada por una distribucion de probabilidad. Dado que los parametros del modelo óptimos están determinados por esta distribución de probabilidad subyacente, un método preciso de aprendizaje automático $\mathcal{A}$ debería devolver la

misma (o muy similar) salida $\mathcal{A}(\mathcal{D})$ para los tres conjuntos de datos. En otras palabras, cualquier $\mathcal{A}$ útil debe ser robusto frente a la variabilidad en las realizaciónes de muestras provenientes de la misma distribución de probabilidad, es decir, debe ser estable.

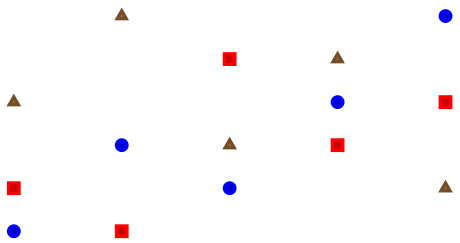


Fig. 29. Tres conjuntos de datos diferentes $\mathcal{D}^{(*)}$, $\mathcal{D}^{(\square)}$, y $\mathcal{D}^{(\triangle)}$, cada uno muestreado independientemente del mismo distribución de datos generador de datos. Un método estable de aprendizaje automático debería generar resultados similares al entrenarse con cualquiera de estos conjuntos de entrenamiento.

.

Vea también: ML, conjunto de datos, conjunto de entrenamiento, parámetros del modelo, predicción, punto de datos, generalización, riesgo, datos, distribución de probabilidad,

estimador de Bayes Considera un modelo de probabilidad 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 funcion de perdida 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 minimo [62]. 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 funcion de perdida $L(\cdot, \cdot)$.

Vea también: modelo probabilístico, distribución de probabilidad, atributo, etiqueta, punto de datos, función de pérdida, hipótesis, riesgo, mínimo.

estocástico Un proceso o método se denomina estocástico si implica un componente aleatorio o está regido por leyes probabilísticas. En aprendizaje automático, los métodos estocásticos a menudo incorporan aleatoriedad por razones como la optimización (por ejemplo, descenso por gradiento estocástico) o la modelización de la incertidumbre (por ejemplo, mediante un modelo de probabilidad). Un proceso estocástico es una colección de variables aleatorias indexadas por tiempo o espacio, que se utilizan para modelar fenómenos aleatorios que evolucionan en el tiempo (p. ej., ruido en sensores o series temporales financieras).

Véa también: ML, SGD, incertidumbre, modelo probabilístico, RV, SBM.

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" [37], [38], [39].

Vea también: punto de datos

evento Considera una variable aleatoria $\mathbf{x}$, definida en algún espacio de probabilidad $\mathcal{P}$, que toma valores en un espacio medible $\mathcal{X}$. Un evento

$\mathcal{A} \subseteq \mathcal{X}$ es un subconjunto de $\mathcal{X}$ tal que la probabilidad $\mathbb{P}(\mathbf{x} \in \mathcal{A})$ está bien definida. En otras palabras, la preimagen $\mathbf{x}^{-1}(\mathcal{A})$ de un evento pertenece a la σ-álgebra de $\mathcal{P}$.

Véa también: punto de datos, suposición de independencia e idéntica distribución (suposición i.i.d.), RV, modelo probabilístico.

experimento aleatorio Un experimento aleatorio es un proceso físico (o abstracto) que produce un resultado ω a partir de un conjunto de posibilidades Ω. Este conjunto de todos los posibles resultados se denomina espacio muestral del experimento. La característica clave de un experimento aleatorio es que su resultado es impredecible (o incierto). Cualquier medición u observación del resultado es una variable aleatoria, es decir, una función del resultado $\omega \in \Omega$. La teoría de la probabilidad utiliza un espacio de probabilidad como estructura matemática para el estudio de los experimentos aleatorios. Una propiedad conceptual clave de un experimento aleatorio es que puede repetirse bajo condiciones idénticas. Estrictamente hablando, repetir un experimento aleatorio un número dado m de veces define un nuevo experimento aleatorio. Los resultados de este nuevo experimento son secuencias de longitud m de resultados del experimento original. Aunque el resultado de un único experimento es incierto, el comportamiento a largo plazo de los resultados de experimentos repetidos tiende a volverse cada vez más predecible. Esta afirmación informal puede formalizarse mediante resultados fundamentales de la teoría de la probabilidad, tales como la ley de los grandes numeros y el teorema central del limite.

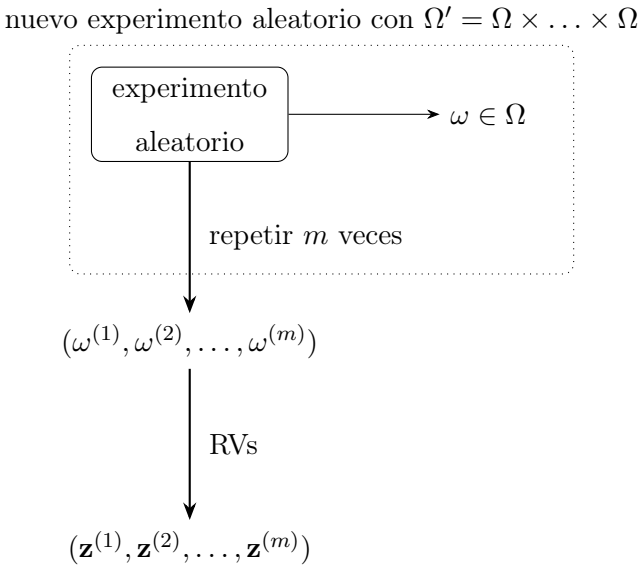


Fig. 30. Un experimento aleatorio produce un resultado $\omega \in \Omega$ a partir de un conjunto de posibilidades (o espacio muestral) Ω. Repetir el experimento m veces da lugar a otro experimento aleatorio, cuyos resultados son secuencias $(\omega^{(1)}, \omega^{(2)}, \ldots, \omega^{(m)}) \in \Omega \times \ldots \times \Omega$. Un ejemplo de experimento aleatorio que aparece en muchas aplicaciones de aprendizaje automatico es la recopilación de un conjunto de entrenamiento $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}$.

Ejemplos de experimentos aleatorios que surgen en aplicaciones de aprendizaje automático incluyen:

- Recopilación de datos: los puntos de datos recogidos en métodos basados en ERM pueden interpretarse como variables aleatorias, es decir, como funciones del resultado $\omega \in \Omega$ de un experimento aleatorio.

- El descenso por gradiente estocástico utiliza un experimento aleatorio en cada iteración para seleccionar un subconjunto del conjunto de entrenamiento.
- Los métodos de protección de la privacidad emplean experimentos aleatorios para generar ruido que se añade a las salidas de un método de aprendizaje automático para garantizar la privacidad diferencial.

experto El aprendizaje automático 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 prediccion utilizando una funcion de perdida. Idealmente, buscamos una hipótesis que incurra en la perdida mínima para cualquier punto de datos. Podemos hacer este objetivo más preciso mediante la suposición de independencia e idéntica distribución y utilizando el riesgo de bayes como referencia (baseline) para la perdida promedio de una hipótesis. Una manera alternativa de obtener una referencia es utilizar la hipótesis h' aprendida por un método de aprendizaje automático existente. A esta hipótesis h' la denominamos experto [28]. Los métodos de minimización de arrepentimiento aprenden una hipótesis que incurre en una perdida comparable a la del mejor experto [27, 28].

Vea también: ML, hipótesis, etiqueta, glsdatapoint, atributo, predicción, función de pérdida, pérdida, punto de datos, suposición i.i.d., riesgo de Bayes, referencia (baseline), Arrepentimiento (regret)

explicabilidad Definimos la (subjetiva) explicabilidad de un método de

aprendizaje automático como el nivel de simulabilidad [74] de las predicciones entregadas por un sistema de aprendizaje automático a un usuario humano. Se pueden construir medidas cuantitativas para la explicabilidad (subjetiva) de un modelo entrenado comparando sus predicciones con las predicciones proporcionadas por un usuario en un conjunto de prueba [74,75]. Alternativamente, podemos usar modelos probabilísticos para los datos y medir la explicabilidad de un modelo de aprendizaje automático entrenado mediante la entropía condicional (diferencial) de sus predicciones, dadas las predicciones del usuario [76, 77].

Vea también: ML, predicción, modelo, conjunto de prueba, modelo probabilístico, datos

explicacion Un enfoque para mejorar la transparencia de un método de aprendizaje automático para su usuario humano es proporcionar una explicación junto con las predicciones entregadas por el método. Las explicaciones pueden adoptar diferentes formas. Por ejemplo, pueden consistir en texto legible para humanos o indicadores cuantitativos, como puntuaciones de importancia de los atributos para los atributos individuales de un punto de datos dado [78]. Alternativamente, las explicaciones pueden ser visuales, por ejemplo, mapas de intensidad que resaltan las regiones de una imagen que impulsan la predicción [79]. La Fig. 31 ilustra dos tipos de explicaciones. La primera es una aproximación lineal local $g(\mathbf{x})$ de un modelo no lineal entrenado $\hat{h}(\mathbf{x})$ alrededor de un vector de atributos específico $\mathbf{x}'$, como se utiliza en el método lime. La segunda forma de explicación representada en la figura es un conjunto disperso de predicciones $\hat{h}(\mathbf{x}^{(1)}), \hat{h}(\mathbf{x}^{(2)}), \hat{h}(\mathbf{x}^{(3)})$

en vectores de atributos seleccionados, que ofrecen puntos de referencia concretos para el usuario.

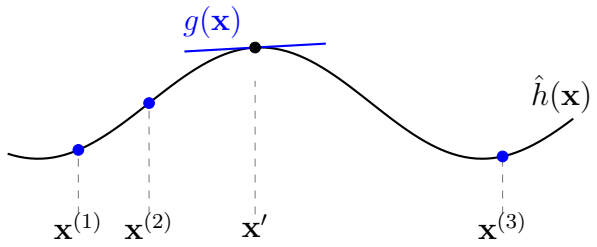


Fig. 31. Un modelo entrenado $\hat{h}(\mathbf{x})$ puede explicarse localmente en algún punto $\mathbf{x}'$ mediante una aproximación lineal $g(\mathbf{x})$. Para un $\hat{h}(\mathbf{x})$ diferenciable, esta aproximación está determinada por el gradiente $\nabla\hat{h}(\mathbf{x}')$. Otra forma de explicación podrían ser los valores de la funcion $\hat{h}\big(\mathbf{x}^{(r)}\big)$ para $r = 1, 2, 3$.

Vea también: explicabilidad, trustworthy AI.

Explicaciones Locales Interpretables e Independientes del Modelo (LIME)

Consideremos un modelo entrenado (o una hipótesis aprendida) $\widehat{h} \in \mathcal{H}$, que asigna el 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 $\mathbf{x}^{(0)}$ [80]. La explicación se da en forma de una aproximación local $g \in \mathcal{H}'$ de $\widehat{h}$ (véa Fig. 32). Esta aproximación puede obtenerse mediante una instancia de ERM con un conjunto de entrenamiento diseñado cuidadosamente. En particular, el conjunto de entrenamiento consiste en puntos de datos 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 arbol de decision para aproximar (localmente) una red profunda. Otra elección ampliamente utilizada para $\mathcal{H}'$ es el modelo lineal.

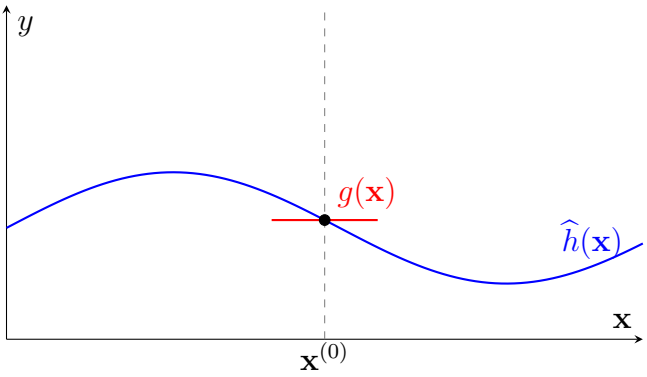


Fig. 32. Para explicar un modelo $\widehat{h} \in \mathcal{H}$ entrenado, alrededor de un vector de atributos $\mathbf{x}^{(0)}$, podemos usar una aproximación local $g \in \mathcal{H}'$.

Vea también: modelo, hipótesis, vector de atributos, punto de datos, predicción, ERM, conjunto de entrenamiento, árbol de decisión, red profunda, modelo lineal

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 parametros del modelo están acoplados a través de la estructura de la red al requerir que tengan una pequeña variación total generalizada (GTV) [81].

Vea también: red FL, GTV

FedAvg FedAvg se refiere a una familia de algoritmos iterativos de aprendizaje federado. Utiliza una configuración cliente-servidor y alterna entre el reentrenamiento de modelos locales en los clientes, seguido de la agregación de parametros del modelo actualizados en el servidor [82]. La actualización local en el cliente $i = 1, \ldots, n$ en el tiempo k comienza con los parametros del modelo actuales $\mathbf{w}^{(k)}$ proporcionados por el servidor y típicamente consiste en ejecutar unas pocas iteraciones de descenso por gradiente estocastico. Tras completar las actualizaciones locales, estas se agregan en el servidor (por ejemplo, promediándolas). La Fig. 33 ilustra la ejecución de una sola iteración de FedAvg.

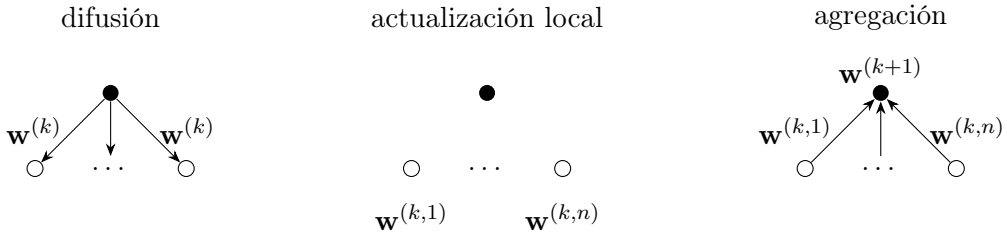


Fig. 33. Ilustración de una sola iteración de FedAvg, que consiste en la difusión de parametros del modelo por parte del servidor, la realización de actualizaciones locales en los clientes y la agregación de las actualizaciones por parte del servidor.

Véa también: FL, algoritmo, modelo local, SGD.

FedGD Un algoritmo distribuido de aprendizaje federado que puede implementarse como paso de mensajes a través de una red de aprendizaje federado.

Véa también: FL, algoritmo distribuido, red FL, paso de gradiente, métodos de gradiente.

FedProx FedProx se refiere a un algoritmo iterativo de aprendizaje federado que alterna entre entrenar modelos locales por separado y combinar los parametros del modelo locales actualizados. A diferencia del promedio federado, que utiliza descenso de gradiente estocástico para entrenar los modelos locales, FedProx usa un operador proximal para el entrenamiento [83].

Vea también: algoritmo, modelo local, parámetros del modelo, FedAvg, SGD, operador proximal

FedRelax Un algoritmo distribuido de aprendizaje federado.

Véa también: FL, algoritmo distribuido.

FedSGD Un algoritmo distribuido de aprendizaje federado que puede implementarse como paso de mensajes a través de una red de aprendizaje federado.

Véa también: FL, algoritmo distribuido, red FL, paso de gradiente, métodos de gradiente, SGD.

filtración de privacidad Consideremos una aplicacion de aprendizaje automático que procesa un conjunto de datos $\mathcal{D}$ y produce una salida, como las predicciones obtenidas para nuevos puntos de datos. 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 el modelo de probabilidad para la generacion de los datos, podemos medir la filtración de privacidad

usando la información mutua (MI) entre la salida y el 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 [84]).

Vea también: ML, conjunto de datos, predicción, punto de datos, atributo, modelo probabilístico, datos, MI, privacidad diferencial (DP)

frontera de decisión Consideremos un mapa de hipótesis h que recibe un vector de atributos $\mathbf{x} \in \mathbb{R}^d$ 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 regiones de decision. 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.

Vea tambien: hipótesis, atributo, región de decisión, entorno.

fuertemente convexa Una función real diferenciable de valor continuo $f(\mathbf{x})$es fuertemente convexa con coeficiente σ si cumple: $f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x}) + (\sigma/2)\|\mathbf{y}-\mathbf{x}\|_2^2$ [15], [85, Sec. B.1.1].

Vea también: diferenciable, convexo

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 red neuronal artificial (RNA) 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 $w_1, \ldots, w_d$.

Vea también:ANN, pesos

función de densidad de probabilidad (pdf) La probabilidad densidad de probabilidad $p(x)$ de una variable aleatoria 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 $\mathbb{P}(x \in \mathcal{B}) = \int_{\mathcal{B}} p(x')dx'$ [7, Ch. 3]. La función de densidad de probabilidad de una variable aleatoria 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 $\mathbb{P}(\mathbf{x} \in \mathcal{R}) = \int_{\mathcal{R}} p(\mathbf{x}')dx'_1 \ldots dx'_d$ [7, Ch. 3].

Vea también: RV, distribución de probabilidad, probabilidad.

función de pérdida Una función de perdida 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 perdida) $L\left((\mathbf{x}, y), h\right)$ 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 34 muestra una

función de perdida para un punto de datos dado, con atributos $\mathbf{x}$ y etiqueta y, como función de la hipótesis $h \in \mathcal{H}$.

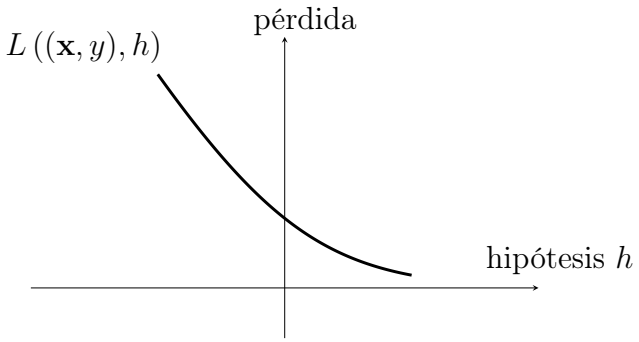


Fig. 34. Alguna funcion de perdida $L\left((\mathbf{x}, y), h\right)$ para un punto de datos fijado, con vector de atributos $\mathbf{x}$ y etiqueta y, y una hipótesis variable h. Los métodos de aprendizaje automático intentan encontrar (o aprender) una hipótesis que incurra en una perdida mínima.

Vea también: pérdida, punto de datos, atributo, etiqueta, hipótesis, predicción, vector de atributos, ML.

función objetivo Una función objetivo es un mapa que asigna un valor numérico objetivo $f(\mathbf{w})$ a cada elección $\mathbf{w}$ de una variable que deseamos optimizar (véa Fig. 35). En el contexto del aprendizaje automático, la variable de optimización puede ser el conjunto de parametros del modelo de una hipótesis $h^{(\mathbf{w})}$. Entre las funciones objetivo comunes se encuentran el riesgo (es decir, la perdida esperada) y el riesgo empírico (es decir, la perdida promedio sobre el conjunto de entrenamiento). Los métodos de aprendizaje automático aplican técnicas de optimización, como los metodos por gradiente, para encontrar la elección $\mathbf{w}$ que

produzca el valor óptimo (por ejemplo, el minimo o el maximo) de la función objetivo.

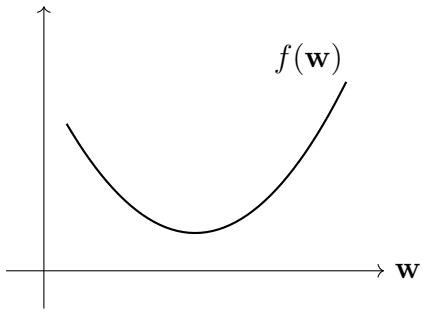


Fig. 35. Una función objetivo asigna a cada valor posible $\mathbf{w}$ de una variable de optimización, como los parametros del modelo de un modelo de aprendizaje automático, un valor que mide la utilidad de $\mathbf{w}$.

Véa también: función, aplicación, ML, parámetros del modelo, hipótesis, riesgo, pérdida, riesgo empírico, conjunto de entrenamiento, métodos de gradiente, mínimo, máximo, modelo, función de pérdida.

generalización Muchos de los sistemas actuales de aprenizaje automático (y inteligencia artificial) se basan en ERM: En esencia, entrenan un modelo (es decir, aprenden una hipótesis $\hat{h} \in \mathcal{H}$) minimizando la perdida promedia (o riesgo empírico) en algunos putnos de datos $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}$, que sirven como un conjunto de entrenamiento $\mathcal{D}^{(\text{train})}$. La generalización se refiere a la capacidad de un método de aprendizaje automático 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 de probabilidad como la suposición iid para la generacion de datos: los puntos de datos en el conjunto de entrenamiento son realizaciónes iid de alguna distribución de probabilidad subyacente $p(\mathbf{z})$. Un modelo de probabilidad nos permite explorar el exterior del conjunto de entrenamiento generando adicionales realizaciónes iid 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 expectativa de la perdida $\bar{L}\big(\hat{h}\big)$. Además, podemos utilizar límites de concentración o resultados de convergencia para secuencias de variables aleatorias iid 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 [53]. También es posible estudiar la generalización sin utilizar modelos de probabilidad. Por ejemplo, podríamos utilizar perturbaciones (determinísticas) de los puntos 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 [86].

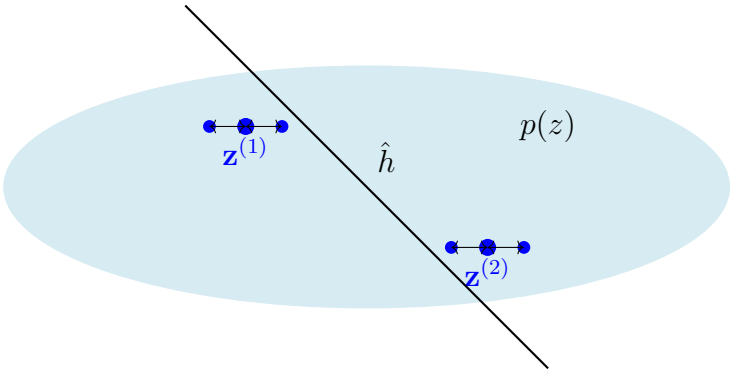


Fig. 36. Dos puntos de datos $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}$ que se utilizan como un conjunto de entrenamiento para aprender una hipótesis $\hat{h}$ mediante ERM. Podemos evaluar $\hat{h}$ fuera de $\mathcal{D}^{(\text{train})}$ ya sea mediante una suposición iid con alguna distribución de probabilidad subyacente $p(\mathbf{z})$ o mediante la perturbación de los puntos 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, $d^{(i)} := \left|\mathcal{N}^{(i)}\right|$.

Vea también: grafo, vecinos

grado de pertenencia El grado de pertenencia es un número que indica en qué medida un punto de datos pertenece a un clúster [8, 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.

Vea también: punto de datos, cluster, Agrupamiento suave, Agrupamiento rígido

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 [87]. 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 $e \in \mathcal{E}$.

Vea también: pesos

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 $\mathcal{V} \setminus \mathcal{V}'$.

Vea también: grafo

grafo de Erdős-Rényi (grafo ER) Un grafo ER es un modelo de probabilidad para grafos definido sobre un conjunto dado de nodos $i = 1, \ldots, n$. Una forma de definir el grafo ER es mediante una colección de variables aleatorias iid binarias $b^{(\{i,i'\})} \in \{0, 1\}$, para cada par de nodos distintos i, i'. Una realización específica del grafo ER contiene una arista $\{i, i'\}$ si y solo si $b^{(\{i,i'\})} = 1$. El grafo ER está parametrizado por el número n de nodos y la probabilidad $\mathbb{P}(b^{(\{i,i'\})} = 1)$.

Véa también: grafo, modelo probabilístico, i.i.d., RV, realización, probabilidad.

grafo de similitud Algunas aplicaciones de aprendizaje automático generan puntos de datos 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 puntos de datos correspondientes son similares.

Vea también: ML, punto de datos, grafo.

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 prediccion $\hat{y} = h(\mathbf{x})$. El aprendizaje automático 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).

Vea también: espacio de etiquetas, punto de datos, atributo, etiqueta, predicción, ML

histograma Considere un conjunto de datos $\mathcal{D}$ que consiste en m puntos de datos $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}$, cada uno perteneciente a una celda $[-U, U] \times \ldots \times [-U, U] \subseteq \mathbb{R}^d$ con longitud lateral U. Particionamos esta celda de manera uniforme en celdas elementales más pequeñas con longitud lateral Δ. El histograma de $\mathcal{D}$ asigna a cada celda elemental la fracción correspondiente de puntos de datos en $\mathcal{D}$ que caen dentro de dicha celda elemental. Un ejemplo visual de tal histograma se muestra en la Fig. 37.

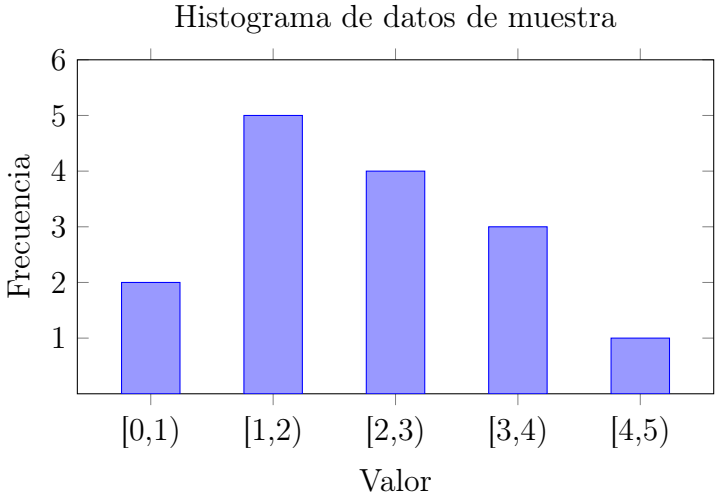


Fig. 37. Un histograma que representa la frecuencia de puntos de datos que caen dentro de rangos de valores discretos (es decir, intervalos). La altura de cada barra muestra la cantidad de muestras en el intervalo correspondiente.

Véa también: conjunto de datos, punto de datos, muestra.

incertidumbre En el contexto del aprendizaje automático, la incertidumbre se refiere a la existencia de múltiples resultados o explicaciones plausibles con base en los datos disponibles. Por ejemplo, la prediccion $\hat{h}(\mathbf{x})$ producida por un modelo de aprendizaje automático entrenado $\hat{h}$ suele reflejar un rango de posibles valores para la verdadera etiqueta de un punto de datos dado. Cuanto más amplio es este rango, mayor es la incertidumbre asociada. La teoría de la probabilidad nos permite representar, cuantificar y razonar sobre la incertidumbre de manera mathemáticamente rigurosa.

Véa también: modelo probabilístico, riesgo, entropía, varianza.

independientes e idénticamente distribuidos (i.i.d.) Es útil interpretar los puntos de datos $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}$ como realizaciónes de variables aleatorias i.i.d., es decir, independientes e idénticamente distribuidos, con una distribución de probabilidad común. Si estas variables aleatorias son de valor continuo, su función de densidad de probabilidad 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 función de densidad de probabilidad marginal común de las variables aleatorias subyacentes.

Vea también: punto de datos, realización, RV, distribución de probabilidad, pdf.

información mutua (MI) La información mutua $I(\mathbf{x}; y)$ entre dos variables aleatorias $\mathbf{x}$, y definidas en el mismo espacio de probabilidad está

dada por [57]

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

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 aprendizaje automático basado en ERM.

Vea también: RV, espacio de probabilidad, predicción, hipótesis, ERM, ML.

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

Vea también: datos.

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 aprendizaje automático 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 funcion de perdida distingue las aplicaciones de IA de las aplicaciones de aprendizaje automático más básicas. Los sistemas de IA rara vez tienen acceso a un conjunto de entrenamiento etiquetado que permita medir la perdida promedio para cualquier posible elección de parametros del modelo. En su lugar, los sistemas de IA utilizan señales de recompensa observadas para obtener una estimación puntual de la

perdida incurrida por la elección actual de parametros del modelo.
Vea también: recompensa, ML, modelo, función de pérdida, conjunto de entrenamiento, pérdida, 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 aprendizaje automático es su confiabilidad [88]. La Unión Europea ha propuesto siete requisitos clave (KRs) para una inteligencia artificial confiable (que típicamente se basa en métodos de aprendizaje automático) [89]:

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.

.

Vea también: aspectos computacionales, aspectos estadísticos, ML, AI

Interfaz de programación de aplicaciones (API) Una API es un mecanismo formal para permitir que componentes de software interactúen de manera estructurada [90]. En el contexto de aprendizaje Automático, las APIs se utilizan frecuentemente para hacer accesible un modelo de

aprendizaje automático 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 modelo de aprendizaje automático permanece oculta para el usuario. Por ejemplo, considera un modelo de aprendizaje automático 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 modelo de aprendizaje automático o su entrenamiento. En la práctica, el modelo de aprendizaje automático suele estar alojado en un ordenador (es decir, un servidor) conectado a internet. Otro ordenador (es decir, un cliente) envía los 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 aprendizaje automático 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.

Vea también: ML, modelo, vector de atributos, punto de datos, predicción, atributo.

interpretabilidad Un método de aprendizaje automático es interpretable para una persona usuaria si esta puede comprender su proceso de decisión. Una forma de desarrollar una definición precisa de interpretabilidad es mediante el concepto de simulabilidad, es decir, la capacidad de una persona de simular mentalmente el comportamiento del mod-

elo [74], [77], [91], [92], [93]. La idea es la siguiente: si una persona entiende un método de aprendizaje automático, entonces debería poder anticipar sus predicciones en un conjunto de prueba. Ilustramos este conjunto de prueba en la Fig. 38, que también muestra dos hipótesis aprendidas: $\hat{h}$ y $\hat{h}'$. El método de aprendizaje automático que produce la hipótesis $\hat{h}$ es interpretable para una persona familiarizada con el concepto de mapa lineal. Dado que $\hat{h}$ corresponde a un mapa lineal, la persona puede anticipar sus predicciones sobre el conjunto de prueba. En cambio, el método que produce $\hat{h}'$ no es interpretable, porque su comportamiento ya no coincide con las expectativas de la persona.

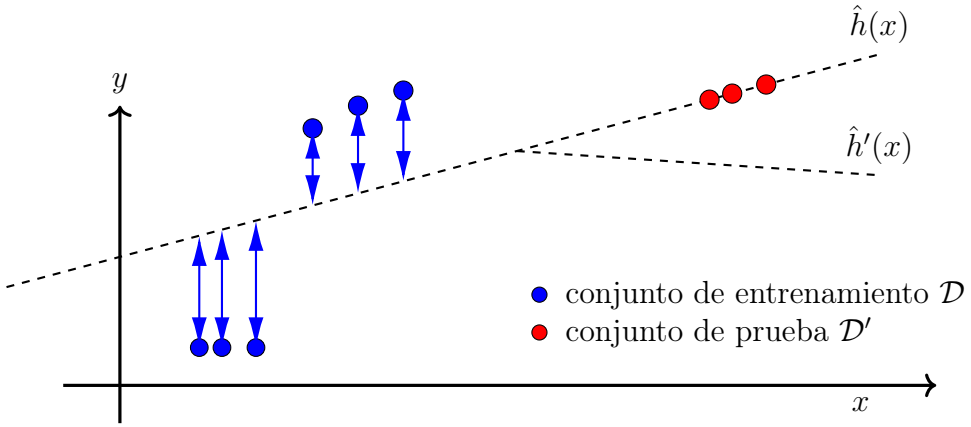


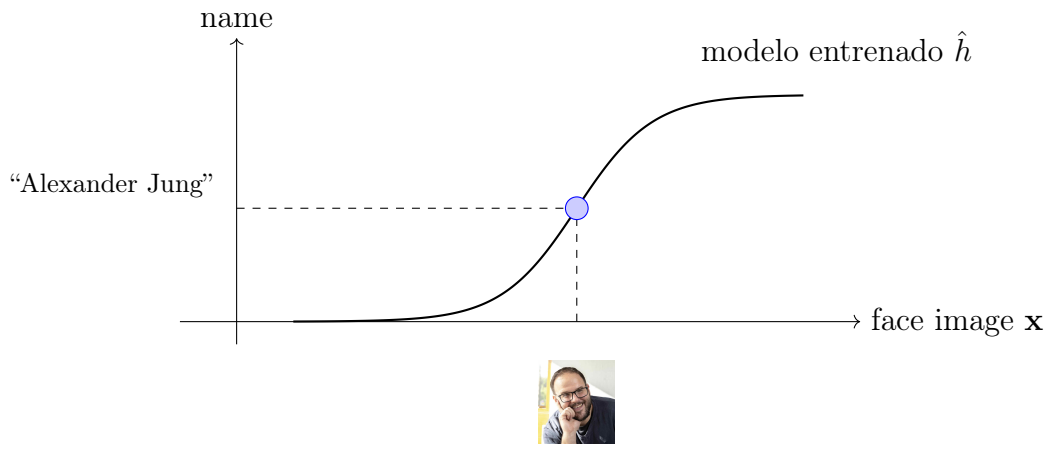
Fig. 38. Podemos evaluar la interpretabilidad de los modelos de aprendizaje automático entrenados $\hat{h}$ y $\hat{h}'$ comparando sus predicciones con pseudo-etiquetas generadas por una persona para $\mathcal{D}'$.

La noción de interpretabilidad está estrechamente relacionada con la de explicabilidad, ya que ambas buscan que los métodos de aprendizaje

automático sean más comprensibles para las personas. En el contexto de la Fig. 38, la interpretabilidad de un método $\hat{h}$ requiere que la persona pueda anticipar sus predicciones sobre cualquier conjunto de prueba. Esto contrasta con la explicabilidad, donde se apoya al usuario mediante explicaciones externas —como mapas de saliencia o ejemplos de referencia del conjunto de entrenamiento— para entender las predicciones de $\hat{h}$ sobre un conjunto de prueba $\mathcal{D}'$.

Vea tambien: explicabilidad, trustworthy AI, regularización, LIME.

inversión de modelo Una inversión de modelo es una forma de ataque privado dirigida a un sistema de aprendizaje automático. Un adversario intenta inferir un atributo sensible de puntos de datos individuales aprovechando el acceso parcial a un modelo entrenado $\hat{h} \in \mathcal{H}$. Este acceso suele consistir en consultar el modelo con predicciónes $\hat{h}(\mathbf{x})$ sobre entradas cuidadosamente seleccionadas. Se han demostrado técnicas básicas de inversión de modelo en el contexto de clasificación de imágenes faciales, donde se reconstruyen imágenes utilizando la (gradiente de la) salida del modelo junto con información auxiliar como el nombre de una persona [94]



Vea tambien: modelo, ataque a la privacidad, ML, atributo sensible, punto de datos, predicción, clasificación, gradiente, trustworthy AI, protección de la privacidad.

Jacobi method El método de Jacobi es un algoritmo para resolver sistemas de ecuaciones lineales de la forma $\mathbf{A}\mathbf{x} = \mathbf{b}$. Aquí, $\mathbf{A} \in \mathbb{R}^{d \times d}$ es una matriz cuadrada con elementos diagonales principales no nulos. El método construye una secuencia $\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \ldots$ actualizando cada componente de $\mathbf{x}^{(k)}$ de acuerdo con la fórmula

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j \neq i} a_{ij} x_j^{(k)} \right).$$

Es importante notar que todos los elementos $x_1^{(k)}, \ldots, x_d^{(k)}$ se actualizan de manera simultánea La iteración anterior converge a una solución, es decir, $\lim_{k \to \infty} \mathbf{x}^{(k)} = \mathbf{x}$, bajo ciertas condiciones sobre la matriz $\mathbf{A}$, por ejemplo, si es estrictamente diagonal dominante o simétrica

definida positiva [3], [95], [13]. Los métodos tipo Jacobi son atractivos para sistemas lineales grandes debido a su estructura paralelizable [26]. Podemos interpretar el método de Jacobi como una iteración de punto fijo. En efecto, al descomponer $\mathbf{A} = \mathbf{D}+\mathbf{R}$, donde $\mathbf{D}$ es la parte diagonal de $\mathbf{A}$, podemos reescribir la ecuación $\mathbf{Ax} = \mathbf{b}$ como una ecuación de punto fijo:

$$\mathbf{x} = \underbrace{\mathbf{D}^{-1}(\mathbf{b} - \mathbf{Rx})}_{\mathcal{F}_{\mathbf{x}}},$$

lo que conduce a la iteración $\mathbf{x}^{(k+1)} = \mathbf{D}^{-1}(\mathbf{b} - \mathbf{Rx}^{(k)})$.

Por ejemplo, para el sistema lineal

$$\mathbf{Ax} = \mathbf{b}, \quad \text{where} \quad \mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix},$$

el método de Jacobi actualiza cada componente de $\mathbf{x}$ como sigue:

$$\begin{aligned} x_1^{(k+1)} &= \frac{1}{a_{11}} \left(b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)} \right), \\ x_2^{(k+1)} &= \frac{1}{a_{22}} \left(b_2 - a_{21}x_1^{(k)} - a_{23}x_3^{(k)} \right), \\ x_3^{(k+1)} &= \frac{1}{a_{33}} \left(b_3 - a_{31}x_1^{(k)} - a_{32}x_2^{(k)} \right). \end{aligned}$$

Vea tambien: algoritmo, método de optimización.

kernel Considere un conjunto de puntos de datos, cada uno representado por un vector de atributos $\mathbf{x} \in \mathcal{X}$, donde $\mathcal{X}$ denota el espacio de atributos. Un kernel (de valores reales) es una función $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ que asigna a cada par de vector de atributos $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ un número real $K\big(\mathbf{x}, \mathbf{x}'\big)$. Este valor se interpreta típicamente como una medida de similitud entre

$\mathbf{x}$ y $\mathbf{x}'$. La propiedad definitoria de un kernel es que es simétrico, es decir, $K\big(\mathbf{x}, \mathbf{x}'\big) = K\big(\mathbf{x}', \mathbf{x}\big)$, y que para cualquier conjunto finito de vectores de atributos $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$, la matriz

$$\mathbf{K} = \begin{pmatrix} K\big(\mathbf{x}_1, \mathbf{x}_1\big) & K\big(\mathbf{x}_1, \mathbf{x}_2\big) & \ldots & K\big(\mathbf{x}_1, \mathbf{x}_n\big) \\ K\big(\mathbf{x}_2, \mathbf{x}_1\big) & K\big(\mathbf{x}_2, \mathbf{x}_2\big) & \ldots & K\big(\mathbf{x}_2, \mathbf{x}_n\big) \\ \vdots & \vdots & \ddots & \vdots \\ K\big(\mathbf{x}_n, \mathbf{x}_1\big) & K\big(\mathbf{x}_n, \mathbf{x}_2\big) & \ldots & K\big(\mathbf{x}_n, \mathbf{x}_n\big) \end{pmatrix} \in \mathbb{R}^{n \times n}$$

es semi-positiva definida. Un kernel define naturalmente una transformación de un vector de atributos $\mathbf{x}$ en una funcion $\mathbf{z} = K\big(\mathbf{x}, \cdot\big)$. La función $\mathbf{z}$ asigna a una entrada $\mathbf{x}' \in \mathcal{X}$ el valor $K\big(\mathbf{x}, \mathbf{x}'\big)$. Podemos ver la funcion $\mathbf{z}$ como un nuevo vector de atributos que pertenece a un espacio de atributos $\mathcal{X}'$, que típicamente es diferente de $\mathcal{X}$. Este nuevo espacio de atributos $\mathcal{X}'$ posee una estructura matemática particular: es un espacio de Hilbert con núcleo reproducible (RKHS, por sus siglas en inglés) [96], [97]. Dado que $\mathbf{z}$ pertenece a un RKHS, que es un espacio de vectores, podemos interpretarlo como un vector de atributos generalizado. Nótese que un vector de atributos de longitud finita $\mathbf{x} = \big(x_1, \ldots, x_d\big)^T \in \mathbb{R}^d$ puede verse como una funcion $\mathbf{x} : \{1, \ldots, d\} \to \mathbb{R}$ que asigna un valor real a cada índice $j \in \{1, \ldots, d\}$. Véa también: vector de atributos, espacio de atributos, espacio de Hilbert, método de kernel.

ley de los grandes números La ley de los grandes números se refiere a la convergencia del promedio de un número creciente (grande) de variables aleatorias iid 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 [65].

Vea también: i.i.d., RV, media, distribución de probabilidad.

lote En el contexto de descenso de gradiente estocástico, un lote se refiere a un subconjunto elegido al azar del conjunto total de conjunto de entrenamiento. Utilizamos los puntos locales de este subconjunto para estimar el gradiente del error de entrenamiento y, a su vez, actualizar los parametros del modelo.

Vea también: SGD, conjunto de entrenamiento, punto de datos, gradiente, error de entrenamiento, parámetros del modelo

mapa de atributos Un mapa de atributos se refiere a una función

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

que transforma un vector de atributos $\mathbf{x} \in \mathcal{X}$ de un punto de datos en un nuevo vector de atributos $\mathbf{x}' \in \mathcal{X}'$, donde típicamente $\mathcal{X}'$ es diferente de $\mathcal{X}$. La representación transformada $\mathbf{x}'$ suele ser más útil que la original $\mathbf{x}$. Por ejemplo, la geometría de los puntos de datos puede volverse más lineal en $\mathcal{X}'$, lo que permite aplicar un modelo lineal a $\mathbf{x}'$. Esta idea es central en el diseño de metodos de kernel [96]. Otros beneficios de usar un mapa de atributos incluyen la reducción del sobreajuste y la mejora de la interpretabilidad [80]. Un caso de uso común es la visualización de datos, donde un mapa de atributos con dos dimensiones de salida permite representar punto de datos en un diagrama de dispersión 2D. Algunos métodos de aprendizaje automático

emplean mapas de atributos entrenables, cuyos parametros se aprenden a partir de datos. Un ejemplo es el uso de capas ocultas en una red profunda, que actúan como mapa de atributos sucesivos [98]. Una forma fundamentada de entrenar un mapa de atributos es mediante ERM, utilizando una funcion de perdida que mide la calidad de reconstrucción, por ejemplo, $L = \|\mathbf{x} - r(\mathbf{x}')\|^2$, donde $r(\cdot)$ es un mapa entrenable que intenta reconstruir $\mathbf{x}$ a partir del vector de atributos transformado $\mathbf{x}'$. Véa también: atributo, aplicación, método de kernel, aprendizaje de atributos, PCA.

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

matriz de confusión Considere puntos de datos caracterizados por atributos $\mathbf{x}$ y sus correspondientes etiquetas y. Las etiquetas toman valores en un espacio de etiquetas finito $\mathcal{Y} = \{1, \ldots, k\}$. Para una hipótesis dada h, la matriz de confusión es una matriz de tamaño $k \times k$ donde cada fila corresponde a un valor distinto de la etiqueta verdadera $y \in \mathcal{Y}$ y cada columna a un valor distinto de la predicción $h(\mathbf{x}) \in \mathcal{Y}$. La entrada (c, c')-ésima de la matriz representa la fracción de puntos de datos con etiquetas verdadera $y = c$ que son clasificadas como $h(\mathbf{x}) = c'$. La diagonal principal de la matriz contiene las fracciones de puntos de datos

correctamente clasificadas (es decir, aquellas para las que $y = h(\mathbf{x})$). Las entradas fuera de la diagonal contienen las fracciones de puntos de datos que son mal clasificadas por h.

Vea tambien: etiqueta, espacio de etiquetas, hipótesis, clasificación.

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

Vea también: RV.

matriz de covarianza muestral La matriz de matriz de covarianza muestral $\widehat{\mathbf{\Sigma}} \in \mathbb{R}^{d\times d}$ para un conjunto dado vectores de atributos $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)} \in \mathbb{R}^d$ 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 $\widehat{\mathbf{m}}$.

Vea también: muestra, matriz de covarianza, vector de atributos, media muestral.

matriz inversa Una matriz inversa $\mathbf{A}^{-1}$ se define para una matriz cuadrada $\mathbf{A} \in \mathbb{R}^{n\times n}$ que sea de rango completo, es decir, que sus columnas sean linealmente independientes. En este caso, se dice que $\mathbf{A}$ es invertible, y su inversa satisface

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}.$$

Una matriz cuadrada es invertible si y solo si su determinante es distinto de cero. Las matrices inversas son fundamentales para resolver sistemas de ecuaciones lineales y para la solución en forma cerrada de regresión lineal [10,95]. El concepto de matriz inversa puede extenderse a matrices

que no son cuadradas o que no tienen rango completo. Se puede definir una "inversa por la izquierda" $\mathbf{B}$ que satisface $\mathbf{BA} = \mathbf{I}$, o una "inversa por la derecha" $\mathbf{C}$ que satisface $\mathbf{AC} = \mathbf{I}$. Para matrices rectangulares o singulares en general, la pseudoinversa de Moore–Penrose $\mathbf{A}^+$ proporciona un concepto unificado de matriz inversa generalizada [3].

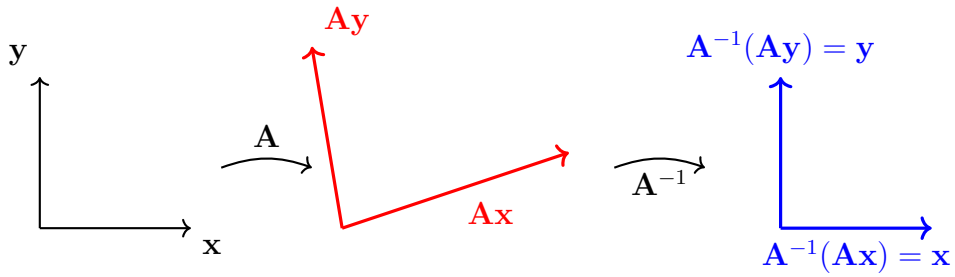


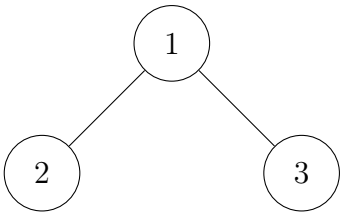
Fig. 39. Una matriz $\mathbf{A}$ representa una transformación lineal de $\mathbb{R}^2$. La matriz inversa $\mathbf{A}^{-1}$ representa la transformación inversa.

Vea tambien: determinante, regresión lineal.

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 [20, 99]. Se define de forma elemento por elemento como (Vea la Figura 40)

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

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



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

Fig. 40. 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 $\mathcal{G}$.

Vea también grafo, peso de arista.

media La media de una variable aleatoria $\mathbf{x}$, que toma valores en un espacio euclidiano $\mathbb{R}^d$, es su esperanza $\mathbb{E}\{\mathbf{x}\}$. Se define como la integral de Lebesgue de $\mathbf{x}$ con respecto a la distribución de probabilidad subyacente P (por ejemplo, ver [2] o [6]), es decir,

$$\mathbb{E}\{\mathbf{x}\} = \int_{\mathbb{R}^d} \mathbf{x} \, \mathrm{d}P(\mathbf{x}).$$

Es útil pensar en la media como la solución del siguiente problema de minimización de riesgo [7]:

$$\mathbb{E}\{\mathbf{x}\} = \operatorname*{arg\,min}_{\mathbf{c}\in\mathbb{R}^d} \mathbb{E}\big\{ \, \|\mathbf{x} - \mathbf{c}\|_2^2 \, \big\}.$$

También usamos el término para referirnos al promedio de una secuencia finita $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)} \in \mathbb{R}^d$. Sin embargo, estas dos definiciones son esencialmente las mismas. De hecho, podemos usar la secuencia $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)} \in \mathbb{R}^d$ para construir una variable aleatoria discreta $\widetilde{\mathbf{x}} = \mathbf{x}^{(I)}$, con el índice I elegido uniformemente al azar del conjunto

$\{1, \ldots, m\}$. La media de $\widetilde{\mathbf{x}}$ es precisamente el promedio $(1/m) \sum_{r=1}^{m} \mathbf{x}^{(r)}$.

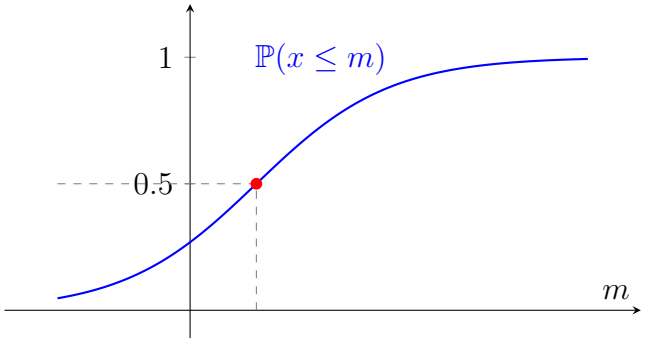
Véa también: RV, esperanza, distribución de probabilidad.

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

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

Vea también: muestra, media, conjunto de datos, vector de atributos.

mediana Una mediana $\text{med}(x)$ de una variable aleatoria real x es cualquier número $m \in \mathbb{R}$ tal que $\mathbb{P}(x \leq m) \geq 1/2$ y $\mathbb{P}(x \geq m) \geq 1/2$ [62].

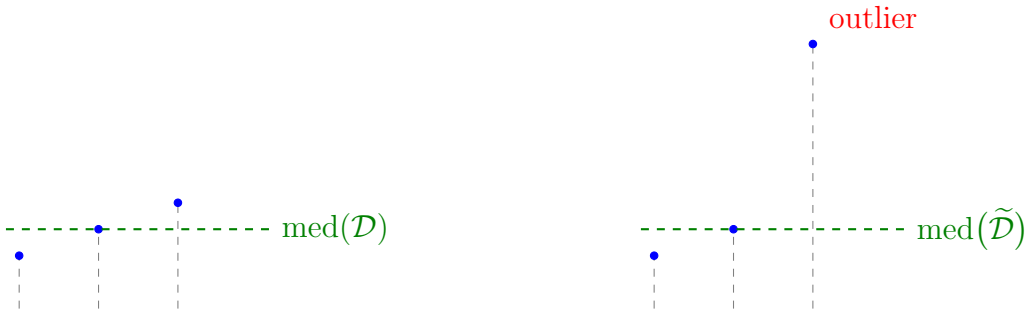


Podemos definir la mediana $\text{med}(\mathcal{D})$ de un conjunto de datos $\mathcal{D} = \{x^{(1)}, \ldots, x^{(m)} \in \mathbb{R}\}$ mediante una variable aleatoria específica $\tilde{x}$ asociada naturalmente con $\mathcal{D}$. En particular, esta variable aleatoria se construye como $\tilde{x} = x^{(I)}$, donde el índice I se elige uniformemente al azar del conjunto $\{1, \ldots, m\}$, es decir, $\mathbb{P}(I = r) = 1/m$ para todo $r = 1, \ldots, m$. Si la variable aleatoria x es integrable, una mediana de x

es la solución del siguiente problema de optimización:

$$\min_{x' \in \mathbb{R}} \mathbb{E}|x - x'|.$$

Al igual que la media, la mediana de un conjunto de datos $\mathcal{D}$ también puede usarse para estimar los parametros del modelo de un modelo de probabilidad subyacente. Comparada con la media, la mediana es más robusta frente a un valor atípico. Por ejemplo, la mediana de un conjunto de datos $\mathcal{D}$ con más de un punto de datos no cambia aunque aumentemos arbitrariamente el valor del mayor elemento de $\mathcal{D}$. En contraste, la media sí lo haría.



(a) conjunto de datos original $\mathcal{D}$. (b) conjunto de datos ruidoso $\widetilde{\mathcal{D}}$ con un valor atípico.

Fig. 41. La mediana es robusta frente a la presencia de punto atípico.

Véa también: media, robustez, valor atípico (outlier).

mediana geométrica (GM) La GM de un conjunto de vectores de entrada $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}$ en $\mathbb{R}^d$ es un punto $\mathbf{z} \in \mathbb{R}^d$ que minimiza la suma de

distancias a dichos vectores [14], es decir:

$$\mathbf{z} \in \operatorname*{argmin}_{\mathbf{y}\in\mathbb{R}^d} \sum_{r=1}^{m} \left\|\mathbf{y} - \mathbf{x}^{(r)}\right\|_2 . \tag{5}$$

La Figura 42 ilustra una propiedad fundamental de la GM: si $\mathbf{z}$ no coincide con ninguno de los vectores de entrada, entonces los vectores unitarios que apuntan desde $\mathbf{z}$ hacia cada $\mathbf{x}^{(r)}$ deben sumar cero; esta es la condición de subgradiente nulo (óptima) de (5). Además, se ha demostrado que la solución de (5) no puede alejarse arbitrariamente de los vectores de entrada confiables mientras estos constituyan la mayoría [100, Th. 2.2].

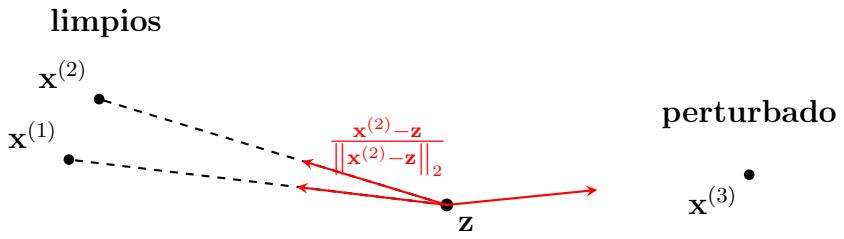


Fig. 42. Sea $\mathbf{z}$ una solución de (5) que no coincide con ninguno de los vectores de entrada. La condición de optimalidad requiere que los vectores unitarios desde $\mathbf{z}$ hacia los vectores de entrada sumen cero.

Véa también: subgradiente.

medible Considera un experimento aleatorio, como registrar la temperatura del aire en una estación meteorológica FMI. El correspondiente espacio muestral Ω consta de todos los resultados posibles ω (por ejemplo, todos los valores posibles de temperatura en grados Celsius). En muchas

aplicaciones de aprendizaje automático, no estamos interesados en el resultado exacto ω, sino solo en si pertenece a un subconjunto $\mathcal{A} \subseteq \Omega$ (por ejemplo, "¿es la temperatura inferior a cero grados?"). Llamamos a dicho subconjunto $\mathcal{A}$ medible si es posible decidir, para cualquier resultado ω, si $\omega \in \mathcal{A}$ o no.

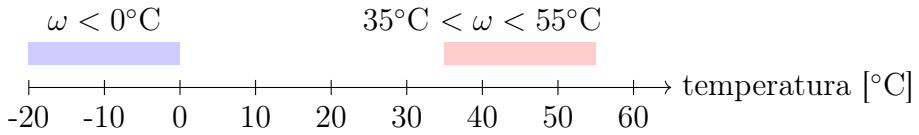


Fig. 43. Un espacio muestral constituido por todos los valores de temperatura posibles ω que pueden experimentarse en una estación FMI. Se destacan dos subconjuntos medibles de valores de temperatura, denotados $\mathcal{A}^{(1)}$ y $\mathcal{A}^{(2)}$. Para cualquier valor de temperatura real ω, es posible determinar si $\omega \in \mathcal{A}^{(1)}$ y si $\omega \in \mathcal{A}^{(2)}$.

En principio, los conjuntos medibles podrían elegirse libremente (por ejemplo, dependiendo de la resolución del equipo de medición). Sin embargo, a menudo es útil imponer ciertos requisitos de completitud en la colección de conjuntos medibles. Por ejemplo, el propio espacio muestral debería ser medible, y la unión de dos conjuntos medibles también debería ser medible. Estos requisitos de completitud pueden formalizarse mediante el concepto de σ-álgebra (o campo σ) [1], [6], [101]. Un espacio medible es un par $\left(\mathcal{X}, \mathcal{F}\right)$ que consta de un conjunto arbitrario $\mathcal{X}$ y una colección $\mathcal{F}$ de subconjuntos medibles de $\mathcal{X}$ que forman una σ-álgebra. Véa también: probabilidad, espacio muestral.

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 minimización del riesgo empírico regularizado que utiliza la variación total generalizada de los parametros del modelo locales como un regularizador [102].

Vea también: RERM, GTV, parámetros del modelo, regularizador

minimización del riesgo empírico explicable (EERM) La minimización del riesgo empírico explicable (EERM) es una instancia de minimización del riesgo estructural que agrega un término de regulación a la perdida promedio en la funcion objetivo de ERM. El término de regulación se elige para favorecer mapas de hipótesis que sean intrínsecamente explicables para un usuario específico. Este usuario se caracteriza por sus predicciones proporcionadas para los puntos de datos en un conjunto de entrenamiento [75].

Vea también: SRM, regularización, pérdida, función objetivo, ERM, hipótesis, predicción, punto de datos, conjunto de entrenamiento.

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 regulación 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),

$$\hat{h} \in \operatorname*{argmin}_{h \in \mathcal{H}} \widehat{L}\big(h|\mathcal{D}\big) + \alpha \mathcal{R}\big\{h\big\}. \tag{6}$$

El parámetro $\alpha \geq 0$ controla la intensidad de la regulación. Para $\alpha = 0$, recuperamos el ERM estándar sin regulación. 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 funcion objetivo de (6) se puede entender intuitivamente como un sustituto para el aumento promedio en la perdida que puede ocurrir al predecir etiquetas para puntos de datos 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 perdida causado por la adición de variables aleatorias gaussianas a los vectores de atributos en el conjunto de entrenamiento [8, 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 minimización del riesgo estructural (SRM) [103, Sec. 7.2].

Vea también: ERM, hipótesis, modelo, riesgo empírico, conjunto de entrenamiento, sobreajuste, regularización, regularizador, punto de datos, pérdida, etiqueta, modelo lineal, vector de atributos,

minimización del riesgo estructural (SRM) La minimización del riesgo estructural (SRM) es una forma de minimización del riesgo empírico regularizado (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 [53, Sec. 7.2].

Vea también: RERM, modelo, ERM, regularizador

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 minima perdida promedio (o riesgo empírico) en un conjunto de datos dado $\mathcal{D}$ (es decir, el conjunto de entrenamiento). Muchos métodos de aprendizaje automático se obtienen a partir de el riesgo empírico mediante elecciones específicas de diseño para el conjunto de datos, el modelo, y la perdida [8, Ch. 3].

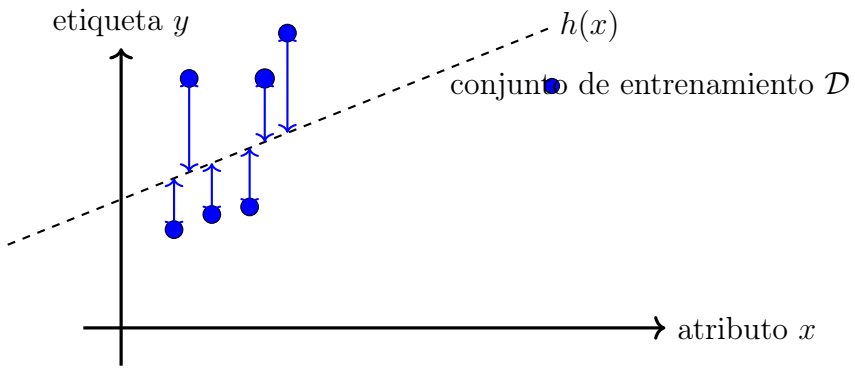


Fig. 44. ERM aprende una hipótesis $h \in \mathcal{H}$, a partir de un modelo $\mathcal{H}$, minimizando la perdida promedio (o riesgo empírico) $\frac{1}{m}\sum_{r=1}^{m} L\left(\left(\mathbf{x}^{(r)}, y^{(r)}\right), h\right)$ incurrida en un conjunto de entrenamiento $\mathcal{D}$.

La Figura 44 ilustra ERM para un modelo lineal y puntos de datos que se caracterizan por un único atributo x y una etiqueta y. La hipótesis h es un mapa lineal que predice la etiqueta de un punto de datos como una función lineal de su atributo x, es decir, $h(x) = w_1 x + w_0$, donde w_1 y w_0 son los parametros del modelo de la hipótesis h. El problema de ERM consiste en encontrar los parametros del modelo w_1 y w_0 que minimicen la perdida promedio incurrida por la hipótesis h en el conjunto de entrenamiento $\mathcal{D}$. Vea también: Riesgo empírico, hipótesis, modelo, mínimo, pérdida, glsemprisk, conjunto de datos, conjunto de entrenamiento, ML

model selection En aprendizaje automático, 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 validacion para cada modelo entrenado; y 3) elegir el modelo con el menor error de validacion [8, Ch. 6].

Vea también: ML, modelo, error de validación

modelo El estudio y diseño de métodos de aprendizaje automático suele basarse en un modelo matemático [104]. Quizás el ejemplo más utilizado de un modelo matemático en aprendizaje automático es un espacio de hipótesis. Un espacio de hipótesis consiste en mapas de hipótesis que son utilizados por un método de aprendizaje automático para predecir etiquetas a partir de los atributos de los puntos de datos. Otro tipo importante de modelo matemático es un modelo de probabilidad, que consiste en distribuciones de probabilidad que describen cómo se generan los puntos de datos. A menos que se indique lo contrario, utilizamos el término modelo para referirnos específicamente al espacio de hipótesis subyacente a un método de aprendizaje automatico.

Véa también: espacio de hipótesis, modelo probabilístico, distribución de probabilidad.

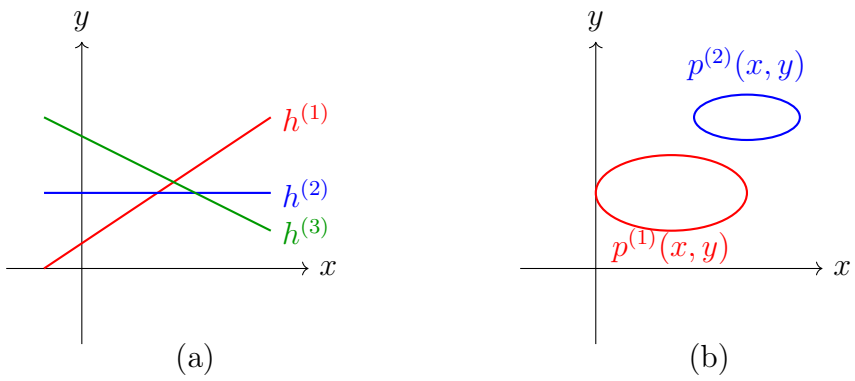


Fig. 45. Dos tipos de modelos matemáticos utilizados en aprendizaje automático: (a) un espacio de hipótesis que consiste en tres mapas lineales; (b) un modelo de probabilidad compuesto por distribuciones de probabilidad sobre el plano definido por los valores de atributos y etiqueta de un punto de datos.

modelo de lenguaje de gran escala (LLM) Los modelos de lenguaje de gran escala son un término genérico para métodos de aprendizaje automático que procesan y generan texto similar al humano. Estos métodos suelen usar redes profundas con miles de millones (o incluso billones) de parametros. Una elección ampliamente utilizada para la arquitectura de red se conoce como Transformers [36]. 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 puntos de datos etiquetados simplemente seleccionando algunas palabras de un texto como etiquetas y las palabras restantes como atributos de puntos de datos. Esta construcción requiere muy poca supervisión humana y permite generar

conjuntos de entrenamiento suficientemente grandes para modelos de lenguaje de gran escala.

Vea también: ML, red profunda, parámetros, punto de dato etiquetado, etiqueta, atributo, punto de datos, conjunto de entrenamiento, modelo.

modelo de mezcla gaussiana (GMM) Un GMM es un tipo particular de modelo de probabilidad para un vector numérico $\mathbf{x}$ (por ejemplo, los 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 variable aleatoria con probabilidades $\mathbb{P}(I = c) = p_c$. Ten en cuenta que un GMM se parametriza por la probabilidad p_c, el vector medio $\boldsymbol{\mu}^{(c)}$, y la matriz de covarianza $\mathbf{\Sigma}^{(\mathbf{c})}$ para cada $c = 1, \ldots, k$. Los GMM se utilizan ampliamente para agrupamiento, estimación de densidad y como un modelo generativo.

Vea también: modelo probabilístico, atributo, punto de datos, distribución normal multivariante, RV, media, matriz de covarianza, agrupamiento, modelo.

modelo en red Un modelo en red sobre un red de aprendizaje federado $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ asigna un modelo local (es decir, un espacio de hipótesis) a cada nodo $i \in \mathcal{V}$ de la red de aprendizaje federado $\mathcal{G}$.

Vea también: red FL, modelo local, espacio de hipótesis

modelo estocástico de bloques (SBM) El modelo estocástico de bloques (SBM) es un modelo generativo probabilístico para un grafo no dirigido $\mathcal{G} = \big(\mathcal{V}, \mathcal{E}\big)$ con conjunto de nodos $\mathcal{V}$ [105]. En su forma básica, asigna

aleatoriamente cada nodo $i \in \mathcal{V}$ a un 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.

Vea también: modelo, grafo, cluster, probabilidad, etiqueta

modelo lineal Considérese una aplicación de aprendizaje automático que involucra puntos de datos, cada uno representado por un vector de atributos numérico $\mathbf{x} \in \mathbb{R}^d$. Un modelo lineal define un espacio de hipótesis que consiste en todas los mapas lineales con valores reales de $\mathbb{R}^d$ a $\mathbb{R}$ tal que

$$\mathcal{H}^{(d)} := \left\{ h : \mathbb{R}^d \to \mathbb{R} \mid h(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} \text{ para algún } \mathbf{w} \in \mathbb{R}^d \right\}. \quad (7)$$

Cada valor de d define un espacio de hipótesis diferente, correspondiente al número de atributos utilizados para calcular la predicción $h(\mathbf{x})$. La elección de d suele guiarse no solo por consideraciones de aspectos computacionales (por ejemplo, menos atributos reducen el costo computacional) y de aspectos estadísticos (por ejemplo, más atributos típicamente reducen el sesgo y el riesgo), sino también por criterios de interpretabilidad. Un modelo lineal que utiliza un pequeño número de atributos bien elegidas suele considerarse más interpretable [30], [80]. El modelo lineal resulta atractivo porque típicamente puede entrenarse usando metodos de optimización convexos escalables [41], [16]. Además, los modelos lineales permiten a menudo un análisis estadístico riguroso, incluyendo límites fundamentales sobre el riesgo minimo alcanzable [52]. También son útiles para analizar modelos más comple-

jos y no lineales, como las redes neuronales artificiales. Por ejemplo, una red profunda puede verse como la composición de un mapa de atributos—implementado por las capas de entrada y ocultas—y un modelo lineal en la capa de salida. De manera similar, un arbol de decision puede interpretarse como la aplicación de un mapa de atributos codificado con one-hot basado en regiones de decision, seguido de un modelo lineal que asigna una predicción a cada región. De forma más general, cualquier modelo entrenado $\hat{h} \in \mathcal{H}$ que sea diferenciable en algún punto $\mathbf{x}'$ puede aproximarse localmente mediante una mapa lineal $g(\mathbf{x})$. La Figura 46 ilustra tal aproximación lineal local, definida por el gradiente $\nabla\hat{h}(\mathbf{x}')$. Nótese que el gradiente solo está definido donde $\hat{h}$ es diferenciable. Para garantizar la robustez en el contexto de AI confiable, puede preferirse modelos cuya mapa asociada $\hat{h}$ sea continua de Lipschitz. Un resultado clásico del análisis matemático—el Teorema de Rademacher—establece que si $\hat{h}$ es continua de Lipschitz con alguna constante L sobre un conjunto abierto $\Omega \subseteq \mathbb{R}^d$, entonces $\hat{h}$ es diferenciable casi en todas partes en Ω [106, Th. 3.1].

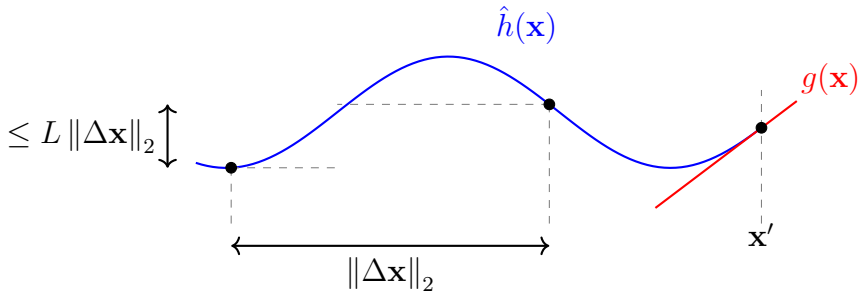


Fig. 46. Un modelo entrenado $\hat{h}(\mathbf{x})$ que es diferenciable en el punto $\mathbf{x}'$ puede ser aproximado localmente por el mapa lineal $g \in \mathcal{H}^{(d)}$. Esta aproximación local es determinada por el gradiente $\nabla\hat{h}(\mathbf{x}')$.

Vea tambien: modelo, espacio de hipótesis, aplicación lineal, interpretabilidad, LIME.

modelo local Considera una colección de conjuntos de datos locales que están asignados a los nodos de una red de aprendizaje federado. 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 espacios de hipótesis, es decir, en general $\mathcal{H}^{(i)} \neq \mathcal{H}^{(i')}$ para diferentes nodos $i, i' \in \mathcal{V}$.

Vea también: dispositivo, red FL, modelo, espacio de hipótesis.

modelo probabilístico Un modelo probabilístico interpreta los puntos de datos como realizaciónes de variables aleatorias con una distribución de probabilidad conjunta. Esta distribución de probabilidad conjunta típicamente incluye parametros que deben seleccionarse manualmente or o aprenderse usando métodos de inferencia estadística como la estimación por máxima verosimilitud [62].

Vea también: modelo, punto de datos, realización, RV, distribución de probabilidad, parámetros, máxima verosimilitud

muestra Una secuencia finita (o lista) de puntos de datos $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}$ que se obtiene o interpreta como la realización de m variables aleatorias iid con una distribución de probabilidad común $p(\mathbf{z})$. La longitud m de la secuencia se denomina tamaño de muestra.

Vea también: punto de datos, realización, i.i.d., RV, distribución de probabilidad, 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 47, su objetivo es separar al máximo los puntos de datos de las dos clases diferentes en el espacio de atributos (es decir, el principio de margen maximo). Maximizar esta separación es equivalente a minimizar una variante regularizada de la perdida de hinge (11) [72, 97, 107].

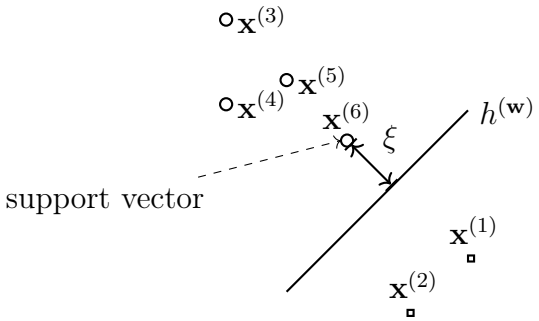


Fig. 47. La SVM aprende una hipótesis (o clasificador) $h^{(\mathbf{w})}$ con pérdida de margen suave promedio mínima perdida de hinge. Minimizar esta perdida es equivalente a maximizar el margen ξ entre la frontera de decision de $h^{(\mathbf{w})}$ y cada clase del conjunto de entrenamiento.

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

Vea también: clasificación, hipótesis, regresión lineal, regresión logística, ERM, modelo lineal, función de pérdida, punto de datos, espacio de atributos, máximo, pérdida de hinge, máquina de vectores de soporte (SVM), clasificador, pérdida, frontera de decisión, conjunto de entrenamiento, ML, kernel.

máxima verosimilitud Considera puntos de datos $\mathcal{D} = \left\{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}\right\}$

que se interpretan como las realizaciónes de variables aleatorias iid con una distribución de probabilidad común $\mathbb{P}(\mathbf{z}; \mathbf{w})$ que depende de los parametros del modelo $\mathbf{w} \in \mathcal{W} \subseteq \mathbb{R}^n$. Los métodos de máxima verosimilitud aprenden los parametros del modelo $\mathbf{w}$ al maximizar la probabilidad (densidad) $\mathbb{P}(\mathcal{D}; \mathbf{w}) = \prod_{r=1}^{m} \mathbb{P}(\mathbf{z}^{(r)}; \mathbf{w})$ del conjunto de datos observado. Por lo tanto, el estimador de máxima verosimilitud es una solución al problema de optimización $\max_{\mathbf{w} \in \mathcal{W}} \mathbb{P}(\mathcal{D}; \mathbf{w})$.

Vea también: punto de datos, realización, i.i.d., RV, distribución de probabilidad, parámetros del modelo, máximo, conjunto de datos.

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 [2, Sec. 1.4]

Vea también: supremo (o mínimo de las cotas superiores).

método de kernel Un método de kernel es un método de aprendizaje Automático 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 $\mathbf{z} = K\big(\mathbf{x}, \cdot\big)$ [96,97]. La motivación para transformar los vectores de atributos es que, al utilizar un kernel adecuado, los puntos de datos presentan una geometría "más conveniente" en el espacio de atributos. Por ejemplo, en un problema de clasificación binaria, el uso de vectores de atributos transformados $\mathbf{z}$ podría permitirnos utilizar modelos lineales, incluso si los puntos de datos no son linealmente separables en el espacio de atributos original (Vea la Figura 48).

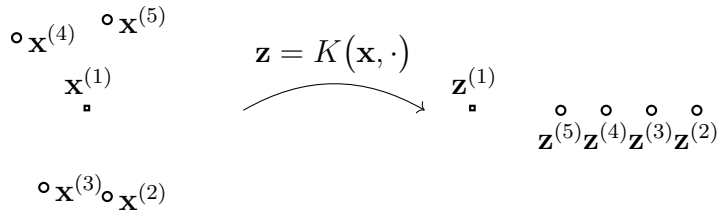


Fig. 48. Cinco puntos de datos caracterizados por vectores de atributos $\mathbf{x}^{(r)}$ y etiquetas $y^{(r)} \in \{\circ, \square\}$, para $r = 1, \ldots, 5$. Con estos vectores de atributos, no hay forma de separar las dos clases mediante una línea recta (rque representa la frontera de decisión de un clasificador lineal). En contraste, los vectores de atributos transformados $\mathbf{z}^{(r)} = K(\mathbf{x}^{(r)}, \cdot)$ permiten separar los puntos de datos usando un clasificador lineal

Vea tambien: kernel, ML, vector de atributos, punto de datos, espacio de atributos, clasificación, modelo lineal, etiqueta, frontera de decisión, clasificador lineal.

método de optimización Un método de optimización es un algoritmo que recibe como entrada una representación de un problema de optimización y entrega como salida una solución (aproximada) [14], [15], [16].
Véa también: algoritmo, problema de optimización.

métodos de gradiente Los métodos de gradiente son técnicas iterativas para encontrar el minimo(o maximo) de una función objetivo diferenciable respecto a los parametros del modelo. Estos métodos construyen una secuencia de aproximaciones hacia una elección óptima de parametros del modelo que resulta en un minimo (o maximo) valor de la funcion

objetivo. Como su nombre indica, los métodos basados en gradiente utilizan los gradientes de la funcion objetivo evaluados en iteraciones previas para construir nuevos parametros del modelo (esperablemente) mejorados. Un ejemplo importante de un método basado en gradiente es el descenso por gradiente.

Vea tambien: gradiente, mínimo, máximo, diferenciable, función objetivo, parámetros del modelo, GD.

métrica En su forma más general, una métrica es una medida cuantitativa utilizada para comparar o evaluar objetos. En matemáticas, una métrica mide la distancia entre dos puntos y debe cumplir ciertas reglas: la distancia es siempre no negativa, igual a cero solo si los puntos son idénticos, simétrica y satisface la desigualdad triangular [2]. En el aprendizaje automático, una métrica es una medida cuantitativa del rendimiento de un modelo. Algunos ejemplos son la precisión (accuracy), la precisión y el promedio de la perdida 0-1 sobre un conjunto de entrenamiento [108], [72]. Una función de perdida se utiliza para entrenar modelos, mientras que una métrica se utiliza para comparar modelos ya entrenados.

Véa también: ML, modelo, precisión, 0/1 loss, conjunto de prueba, función de pérdida, pérdida, selección de modelo.

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 [15].

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 [11].

norma dual Toda norma $\|\cdot\|$ definida en un espacio euclidiano $\mathbb{R}^d$ tiene una norma dual asociada, denotada por $\|\cdot\|_*$ y definida como $\|\mathbf{y}\|_* := \sup_{\|\mathbf{x}\|\leq 1} \mathbf{y}^T\mathbf{x}$. La nroma dual mide el mayor producto interno posible entre $\mathbf{y}$ y cualquier vector en la bola unitaria de la norma original. Para más detalles, véa [14, Sec. A.1.6].

Véa también: norma, espacio euclidiano.

normalización de datos La normalización de datos se refiere a transformaciones aplicadas a los vectores de atributos de puntos de datos para mejorar los aspectos estadísticos o aspectos computacionales del método de aprendizaje automático. Por ejemplo, en regresión lineal con metodos de gradiente que utilizan una tasa de aprendizaje fija, la convergencia depende de controlar la norma de los vectores de atributos en el conjunto de entrenamiento. Un enfoque común es normalizar los vectores de atributos de modo que su norma no exceda de uno [8, Ch. 5].

Vea también: datos, vector de atributos, punto de datos, ML, aspectos estadísticos, aspectos computacionales, regresión lineal, métodos de gradiente, tasa de aprendizaje, norma, conjunto de entrenamiento.

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 de $\mathbf{Q}$. El número de condición es útil para el análisis de

métodos de aprendizaje automático. 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 puntos de datos que hagan que $\mathbf{Q}$ tenga un número de condición cercano a 1.

Vea también: ML, métodos de gradiente, regresión lineal, matriz de atributos, conjunto de entrenamiento, atributo, punto de datos

operador de reducción y selección absoluta mínima (Lasso) El Lasso es una implementación de minimización del riesgo estructural (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.

Vea también: SRM, pesos, conjunto de entrenamiento, regresión lineal, norma, pérdida de error cuadrático.

operador proximal Dada una funcion convexa $f(\mathbf{w}')$, definimos su operador proximal como [109, 110]

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

Como se ilustra en la Figura 49, 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 convexa $f(\mathbf{w}')$. 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 proximal de la función $\tilde{f}(\mathbf{w}') = \big(\nabla f(\mathbf{w})\big)^T(\mathbf{w}' - \mathbf{w})$ y usar $\rho = 1/\eta$.

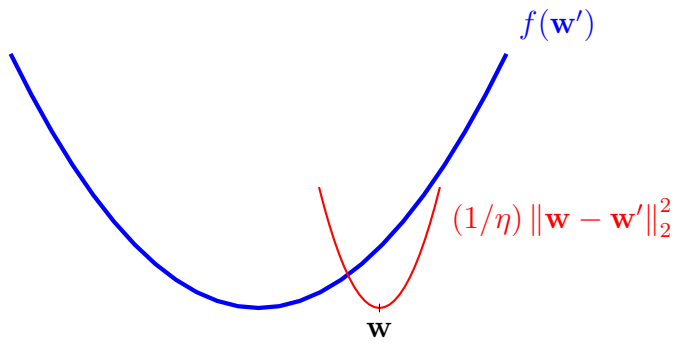


Fig. 49. Un paso de gradiente generalizado actualiza un vector $\mathbf{w}$ minimizando una versión penalizada de la función $f(\cdot)$. El término de penalización es la distancia euclidiana cuadrada escalada entre la variable de optimización $\mathbf{w}'$ y el vector dado $\mathbf{w}$.

optimismo ante la incertidumbre Los metodos de aprendizaje automático aprenden parametros 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 perdida promedio en un conjunto de datos (por ejemplo, el conjunto de entrenamiento) como estimación del riesgo de una hipótesis. Usando un modelo de probabilidad, se puede construir un intervalo de confianza.

$\left[l^{(\mathbf{w})}, u^{(\mathbf{w})}\right]$ para cada elección $\mathbf{w}$ de los parametros 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 parametros 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 aprendizaje automático que usan una construcción optimista de una funcion objetivo son métodos de minimización del riesgo estructural (SRM) [53, Ch. 11] y cota superior de confianza (UCB) para decisiones secuenciales [40, Sec. 2.2].

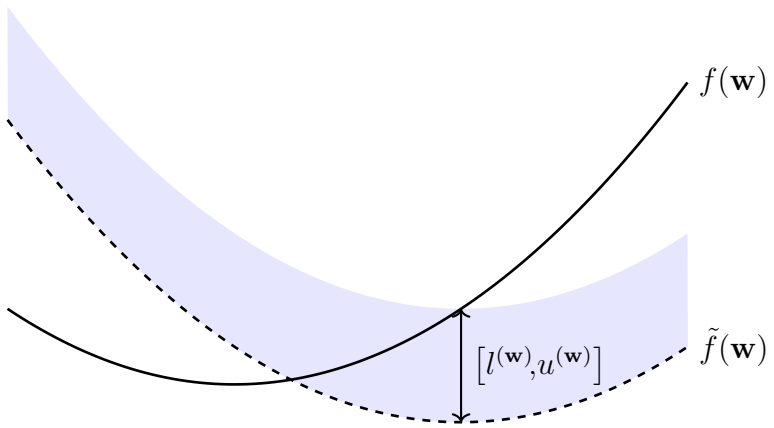


Fig. 50. Los métodos de aprendizaje automático aprenden parametros del modelo $\mathbf{w}$ usando una estimación de $f(\mathbf{w})$ como aproximación del criterio de desempeño $\bar{f}(\mathbf{w})$. Usando un modelo de probabilidad, se pueden construir intervalos de confianza $\left[l^{(\mathbf{w})}, u^{(\mathbf{w})}\right]$ que contienen $\bar{f}(\mathbf{w})$ con alta probabilidad. La mejor medida plausible del desempeño para una elección especifica $\mathbf{w}$ es $\tilde{f}(\mathbf{w}) := l^{(\mathbf{w})}$.

.

Vea también: ML, parámetros del modelo, ERM, pérdida, conjunto de datos, conjunto de entrenamiento, riesgo, hipótesis, modelo probabilístico, cota superior de confianza (UCB),

parámetro Un parámetro de un modelo de aprendizaje automático es una cantidad ajustable (es decir, que puede aprenderse o modificarse) que nos permite 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 mapas de hipótesis de la forma $h^{(\mathbf{w})}(x) = w_1 x + w_2$ con una elección

particular de los parámetros $\mathbf{w} = \big(w_1, w_2\big)^T \in \mathbb{R}^2$. Otro ejemplo de un parámetro de modelo es el conjunto de pesos asignados a una conexión entre dos neuronas en una RNA.

Véa también: ML, modelo, hipótesis, aplicación, modelo lineal, pesos, ANN.

parámetros Los parámetros de un modelo de aprendizaje automático 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 $\mathbf{w} = \big(w_1, w_2\big)^T \in \mathbb{R}^2$. Otro ejemplo de parámetros son los pesos asignados a las conexiones entre neuronas de una red neuronal artificial.

Vea también: ML, modelo, hipótesis, modelo lineal, pesos, ANN.

parámetros del modelo Los parametros 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 parametros 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.

Vea también: modelo, parámetros, hipótesis.

paso de gradiente Dada una función 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

(vea la Figura 51)

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

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

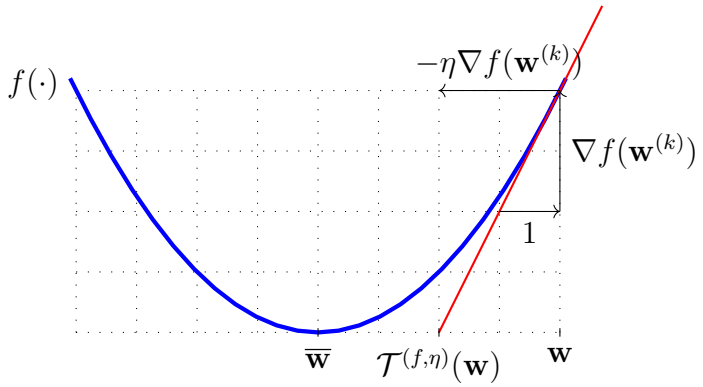


Fig. 51. El paso básico de gradiente (8) mapea un vector $\mathbf{w}$ al vector actualizado $\mathbf{w}'$. Define un operador $\mathcal{T}^{(f,\eta)}(\cdot) : \mathbb{R}^d \to \mathbb{R}^d : \mathbf{w} \mapsto \widehat{\mathbf{w}}$.

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

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

Intencionalmente usamos el mismo símbolo η para el parámetro en (9) que en el tamaño de paso de (8). Mientras mayor sea el valor de η

en (9), más progreso hará la actualización en la reducción del valor de la función $f(\widehat{\mathbf{w}})$. Nótese que, al igual que el paso de gradiente (8), la actualización (9) también define un operador (típicamente no lineal) parametrizado por la función $f(\cdot)$ y el parámetro η. Para una función convexa $f(\cdot)$, este operador es conocido como el operador proximal de $f(\cdot)$ [109].

Vea también: diferenciable, gradiente, tamaño de paso, entorno, generalización, convexo, operador proximal

peso de arista Cada arista $\{i, i'\}$ de una red de aprendizaje federado 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 $i, i' \in \mathcal{V}$.

Vea también: red FL.

pesos Considera un espacio de hipótesis parametrizado $\mathcal{H}$. Usamos el término pesos para los parametros 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 red neuronales artificiales para formar combinaciones lineales de los atributos o de las salidas de las neuronas en capas ocultas.

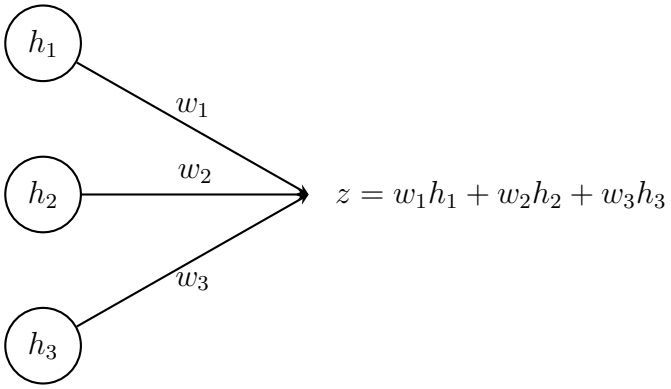


Fig. 52. Una sección de una red neuronal artifical que contiene una capa oculta con salidas (o activaciones) h_1,h_2, and h_3. Estas salidas se combinan linealmente para calcular z que puede usarse como salida de la red neuronal artificial o como entrada de otra capa.

Vea también: espacio de hipótesis, parámetros del modelo, atributo, modelo lineal, ANN.

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

predicción Una predicción es una estimación o aproximación de una cantidad de interés. El Aprendizaje Automático se centra en aprender o encontrar

un mapa de hipótesis h que recibe los atributos $\mathbf{x}$ de un punto datos and y produce una predicción $\widehat{y} := h(\mathbf{x})$ para su etiqueta y.

Vea también: ML, hipótesis, atributo, punto de datos

predictor Un predictor es un mapa de hipótesis con valores reales. Dado un punto 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.

Vea también: hipótesis, punto de datos, atributo, predicción, etiqueta

preimagen Considére una función $f\colon \mathcal{U} \to \mathcal{V}$ entre dos conjuntos. La preimagen $f^{-1}(\mathcal{B})$ de un subconjunto $\mathcal{B} \subseteq \mathcal{V}$ es el conjunto de todas las entradas $u \in \mathcal{U}$ que son mapeadas en $\mathcal{B}$ por f, es decir,

$$f^{-1}(\mathcal{B}) := \{u \in \mathcal{U} \mid f(u) \in \mathcal{B}\}.$$

La preimagen está bien definida incluso si la función f no es invertible [2].

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 [111, Article 5(1)(c)], [112].

Vea también: datos.

privacidad diferencial (DP) Consideremos un método de aprendizaje automático $\mathcal{A}$ 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 predicciones para ciertos puntos de datos. DP es una medida precisa de la filtración de privacidad ocasionada al revelar dicha salida. Aproximadamente, un método de aprendizaje automático 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 de probabilidad para un método de aprendizaje automático, es decir, interpretamos su salida $\mathcal{A}(\mathcal{D})$ como la realización de una variable aleatoria. La aleatoriedad en la salida puede asegurarse añadiendo intencionalmente la realización de una variable aleatoria auxiliar (ruido) a la salida del método de aprendizaje automático.

Vea también: ML, conjunto de datos, conjunto de entrenamiento, ERM, parámetros del modelo, predicción, punto de datos, filtración de privacidad, distribución de probabilidad, atributo sensible, modelo probabilístico, realización, RV

probabilidad Asignamos un valor de probabilidad, típicamente elegido en el intervalo $[0, 1]$, a cada evento que pueda ocurrir en un experimento aleatorio [6, 7, 71, 113].

proceso de decisión de Markov (MDP) Un MDP es una estructura matemática que puede utilizarse para estudiar aplicaciones de aprendizaje por refuerzo. Un MDP formaliza cómo las señales de recompensa dependen de las predicciones (y las acciones correspondientes) realizadas por un método de aprendizaje por refuerzo. Formalmente, un MDP es un tipo

específico de proceso estocástico definido por:

- un espacio de estados $\mathcal{S}$,
- un espacio de acciones $\mathcal{A}$ (cada acción $a \in \mathcal{A}$ corresponde a una prediccion específica realizada por el método de aprendizaje por refuerzo),
- una función de transición $\mathbb{P}(s' \mid s, a)$ que especifica la distribución de probabilidad sobre el siguiente estado $s' \in \mathcal{S}$, dado el estado actual $s \in \mathcal{S}$ y la acción $a \in \mathcal{A}$,
- una función de recompensa $r(s, a) \in \mathbb{R}$ que asigna una recompensa numérica a cada par estado-acción.

La propiedad definitoria de un MDP es la propiedad de Markov: el siguiente estado s' y la recompensa dependen únicamente del estado actual s y de la acción a, y no de todo el historial de interacción.

proceso gaussiano (GP) Un GP es una colección de variables aleatorias $\{f(\mathbf{x})\}_{\mathbf{x}\in\mathcal{X}}$ indexadas por valores de entrada $\mathbf{x}$ de un cierto espacio de entrada $\mathcal{X}$, tal que, para cualquier subconjunto finito $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)} \in \mathcal{X}$, las variables aleatorias correspondientes $f(\mathbf{x}^{(1)}), \ldots, f(\mathbf{x}^{(m)})$ tienen una distribución conjunta gaussiana multivariante:

$$\big(f(\mathbf{x}^{(1)}), \ldots, f(\mathbf{x}^{(m)})\big) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K}).$$

Para un espacio de entrada $\mathcal{X}$ fijo, un GP queda completamente especificado (o parametrizado) por:

- una función media $\mu(\mathbf{x}) = \mathbb{E}\{f(\mathbf{x})\}$

- y una función de covarianza $K\big(\mathbf{x}, \mathbf{x}'\big) = \mathbb{E}\{\big(f(\mathbf{x}) - \mu(\mathbf{x})\big)\big(f(\mathbf{x}') - \mu(\mathbf{x}')\big)\}$.

Ejemplo: Podemos interpretar la distribución de temperatura en Finlandia (en un instante específico) como la realización de un GP $f(\mathbf{x})$, donde cada entrada $\mathbf{x} = (\text{lat}, \text{lon})$ representa una ubicación geográfica. Las observaciones de temperatura de las estaciones meteorológicas de FMI constituyen muestras de $f(\mathbf{x})$ en ubicaciones específicas (vea la Figura 53). Un GP nos permite predecir la temperatura en las cercanías de las estaciones meteorológicas y cuantificar la incertidumbre de dichas predicciones.

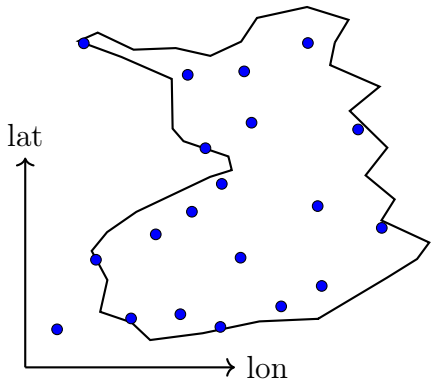


Fig. 53. Podemos interpretar la distribución de temperatura sobre Finlandia como una realización de un GP indexado por coordenadas geográficas y muestreado en estaciones meteorológicas de FMI (indicadas por puntos azules).

Véa también: RV, media, función, realización, FMI, muestra, incer-

tidumbre.

producto de Kronecker El producto de Kronecker de dos matrices $\mathbf{A} \in \mathbb{R}^{m \times n}$ y $\mathbf{B} \in \mathbb{R}^{p \times q}$ es una matriz por bloques denotada como $\mathbf{A} \otimes \mathbf{B}$ y definida como [3], [11]

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix} \in \mathbb{R}^{mp \times nq}.$$

El producto de Kronecker es un caso particular del producto tensorial para matrices y se utiliza ampliamente en estadística multivariada, álgebra lineal y modelos estructurados de aprendizaje automático. Satisface la identidad $(\mathbf{A} \otimes \mathbf{B})(\mathbf{x} \otimes \mathbf{y}) = (\mathbf{A}\mathbf{x}) \otimes (\mathbf{B}\mathbf{y})$ para vectores $\mathbf{x}$ y $\mathbf{y}$ de dimensiones compatibles.

Véa también: ML, modelo.

protección de la privacidad Consideremos un método de aprendizaje automático $\mathcal{A}$ que recibe como entrada un conjunto de datos $\mathcal{D}$ y entega una salida $\mathcal{A}(\mathcal{D})$. La salida puede ser los parametros 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 aprendizaje automático involucran puntos de datos 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 sensible de los puntos de datos 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.

Vea también: ML, conjunto de datos, parámetros del modelo, predicción, punto de datos, atributo, etiqueta, atributo sensible

proximable Una funcion convexa para la cual el operador proximal puede calcularse de manera eficiente se denomina a veces proximable o simple [114].

Vea también: convexo, operador proximal

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

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

En otras palabras, $P_{\mathcal{W}}\big(\mathbf{w}\big)$ es el vector en $\mathcal{W}$ más cercano a $\mathbf{w}$. La proyección está bien definida solo para aquellos subconjuntos $\mathcal{W}$ para los cuales existe el minimo anterior [14].

Vea también: mínimo

pseudoinversa La pseudoinversa de Moore–Penrose $\mathbf{A}^+$ de una matriz $\mathbf{A} \in \mathbb{R}^{m \times d}$ generaliza la noción de una inversa [3]. La pseudoinversa surge de forma natural en el contexto de la regresión ridge cuando se aplica a un conjunto de datos con etiqueta $\mathbf{y}$ arbitrarios y una matriz de atributos $\mathbf{X} = \mathbf{A}$ [41, Cap. 3]. Los parametros del modelo aprendidos mediante regresión Ridge están dados por

$$\widehat{\mathbf{w}}^{(\alpha)} = \big(\mathbf{A}^T\mathbf{A} + \alpha\mathbf{I}\big)^{-1}\mathbf{A}^\top\mathbf{y}, \quad \alpha > 0.$$

Podemos entonces definir la pseudoinversa $\mathbf{A}^{+} \in \mathbb{R}^{d \times m}$ vía el límite [115, Cap. 3]

$$\lim_{\alpha \to 0^{+}} \widehat{\mathbf{w}}^{(\alpha)} = \mathbf{A}^{+}\mathbf{y}.$$

Vea también: matriz inversa, regresión ridge, conjunto de datos, etiquetas, matriz de atributos, parámetros del modelo

puerta trasera (backdoor) Un ataque de puerta trasera (backdoor) se refiere a la manipulación intencional del proceso de entrenamiento de un método de aprendizaje automático. 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. El objetivo de un ataque de puerta trasera es inclinar la hipótesis aprendida $\hat{h}$ hacia predicciones específicas para un rango determinado de valores de atributos. Este rango de atributos actúa como una clave (o desencadenante) que desbloquea una puerta trasera en el sentido de generar predicciones anómalas. La clave $\mathbf{x}$ y la predicción anómala correspondiente $\hat{h}(\mathbf{x})$ son conocidas únicamente por el atacante.
Vea también: ML, conjunto de entrenamiento, datos, algoritmo, ERM, hipótesis, predicción, atributo,

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.
Vea también: punto de datos, etiqueta

punto de datos Un punto de datos es cualquier objeto que transmite infor-

mación [57]. Ejemplos incluyen estudiantes, señales de radio, árboles, imágenes, variables aleatorias, números reales o proteínas. Describimos puntos de datos del mismo tipo mediante dos categorías de propiedades:

1) Atributos son propiedades medibles o computables de un punto de datos. Estos atributos pueden extraerse o calcularse automáticamente mediante sensores, ordenadores u otros sistemas de recolección de datos. Para un punto de datos que representa a un paciente, una atributos podría ser el peso corporal.

2) Etiquetas son hechos de alto nivel (o cantidades de interés) asociados con el punto de datos. Determinar las etiquetas de un punto de datos generalmente requiere experiencia humana o conocimiento del dominio. Para un punto de datos que representa a un paciente, un diagnóstico de cáncer proporcionado por un médico serviría como etiqueta.

La Fig. 54 muestra una imagen como ejemplo de punto de datos junto con sus atributos y etiquetas. Es importante destacar que lo que constituye una atributo o una etiqueta no es inherente al propio punto de datos, sino que es una decisión de diseño que depende de la aplicación específica de aprendizaje automático.

Un único punto de datos.

Atributos:

- $x_1, \ldots, x_{d_1}$: Intensidades de color de todos los píxeles de la imagen.
- x_{d_1+1}: Marca temporal (timestamp) de la captura de la imagen.
- x_{d_1+2}: Ubicación espacial de la captura de la imagen.

Etiquetas:

- y_1: Número de vacas representadas.
- y_2: Número de lobos representados.
- y_3: Estado del pasto (por ejemplo, saludable, sobrepastoreado).

Fig. 54. Ilustración de un punto de datoos que consiste en una imagen. Podemos usar propiedades distintas de la imagen como atributos, y hechos de alto nivel sobre la imagen como etiquetas.

La distinción entre atributos y etiquetas no siempre es nítida. Una propiedad que se considera una etiqueta en un contexto (por ejemplo, un diagnóstico de cáncer) puede tratarse como una atributo en otro—en particular, si la automatización confiable (por ejemplo, mediante análisis de imágenes) permite calcularla sin intervención humana. El aprendizaje automático tiene como objetivo general predecir la etiqueta de un punto de datos en función de sus atributos.

Véa también: datos, atributo, etiqueta, conjunto de datos.

pérdida Los métodos de aprendizaje automático usan una funcion de perdida $L\left(\mathbf{z}, h\right)$ 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 funcion de perdida L en sí como para el valor específico $L\left(\mathbf{z}, h\right)$, para un punto de datos $\mathbf{z}$ y una hipótesis h.

Vea también: ML, función de pérdida, hipótesis, punto de datos.

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

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

.

Vea también: pérdida, predicción, hipótesis, etiqueta, atributo, punto de datos

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

perdida de hinge incurrida por un mapa de hipótesis $h(\mathbf{x})$ se define como

$$L\left((\mathbf{x}, y), h\right) := \max\{0, 1 - yh(\mathbf{x})\}. \tag{11}$$

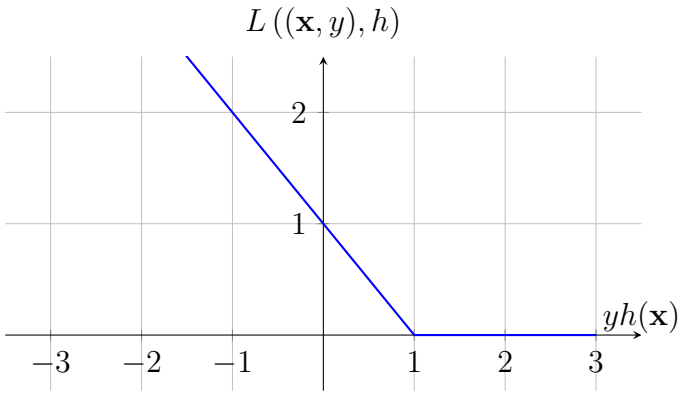


Fig. 55. La perdida de Hinge incurrida por la predicción $h(\mathbf{x}) \in \mathbb{R}$ para un punto de datos con etiqueta $y \in \{-1, 1\}$. Una variante regularizada de la perdida de hinge es usada por la maquina de vectores de soporte [97].

Vea también: punto de datos, vector de atributos, etiqueta, pérdida, hipótesis, SVM.

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

Vea también: pérdida, pérdida de error cuadrático, pérdida por error absoluto.

pérdida logística Considérese un punto de datos caracterizado por los atributos $\mathbf{x}$ y una etiqueta binaria $y \in \{-1, 1\}$. Usamos una hipótesis

con valores reales h para predecir la etiqueta y a partir de los atributos $\mathbf{x}$. La perdida logística incurrida por esta predicción se define como

$$L\left((\mathbf{x}, y), h\right) := \log\left(1 + \exp\left(-yh(\mathbf{x})\right)\right). \tag{12}$$

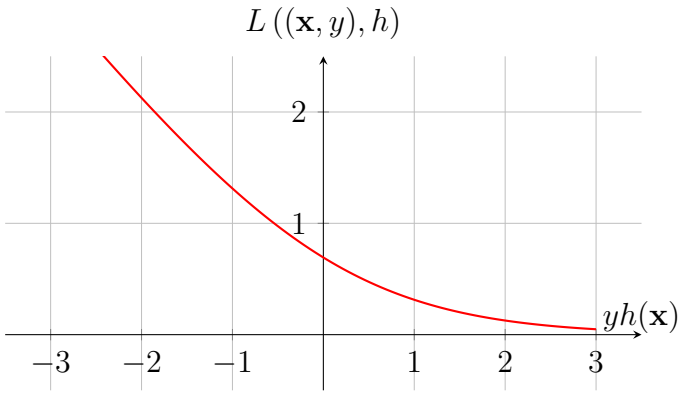


Fig. 56. La perdida logística incurrida por la predicción $h(\mathbf{x}) \in \mathbb{R}$ para un punto de datos con etiqueta $y \in \{-1, 1\}$.

Nótese que la expresión (12) para la perdida logística aplica únicamente para el espacio de etiquetas $\mathcal{Y} = \{-1, 1\}$ y cuando se utiliza la regla de umbral dada en (1).

Véa también: punto de datos, atributo, etiqueta, hipótesis, pérdida, predicción, espacio de etiquetas.

pérdida por error absoluto Considérese un punto de datos con atributos $\mathbf{x} \in \mathcal{X}$ y una etiqueta numérica $y \in \mathbb{R}$. Como su nombre lo sugiere, la pérdida por error absoluto incurrida por una hipótesis $h : \mathcal{X} \to \mathbb{R}$ se define como

$$L\left((\mathbf{x}, y), h\right) = |y - h(\mathbf{x})|.$$

La Figura 57 muestra la pérdida por error absoluto para un punto de datos fijo con vector de atributos $\mathbf{x}$ y etiqueta y. También indica los valores de perdida incurridos por dos diferentes hipótesis h' y h''. Al igual que la perdida por error cuadrático, la perdida por error absoluto es una función convexa de la predicción $\hat{y} = h(\mathbf{x})$. Sin embargo, en contraste con la perdida por error cuadrático, la perdida por error absoluto no es suave, ya que no es diferenciable en la predicción óptima $\hat{y} = y$. Esta propiedad hace que los métodos basados en ERM que usan la perdida por error absoluto sean computacionalmente más exigentes [15, 116]. Para desarrollar intuición, resulta útil considerar las dos hipótesis representadas en la Figura 57. Solo a partir de la pendiente de L alrededor de $h'(\mathbf{x})$ y $h''(\mathbf{x})$ es imposible determinar si estamos muy cerca del óptimo (h') o aún lejos (h''). Como resultado, cualquier método de optimización basado en aproximaciones locales de la función de perdida (como descenso de gradiente estocástico) debe usar una tasa de aprendizaje decreciente para evitar sobrepasar el óptimo al acercarse a él. Esta disminución requerida de la tasa de aprendizaje tiende a ralentizar la convergencia del método de optimización. Además del aumento en la complejidad computacional, usar la perdida por error absoluto en ERM puede ser beneficioso en presencia de valores atípicos en el conjunto de entrenamiento. A diferencia de la perdida por error cuadrático, la pendiente de la perdida por error absoluto no aumenta con el incremento del error de predicción $y - h(\mathbf{x})$. Como resultado, el efecto de introducir valores atípicos con gran error de predicción en la solución $\hat{h}$ de ERM con perdida por error absoluto es mucho menor

en comparación con el efecto sobre la solución de ERM con la perdida de error cuadrático. Véa también: punto de datos, atributo, etiqueta,

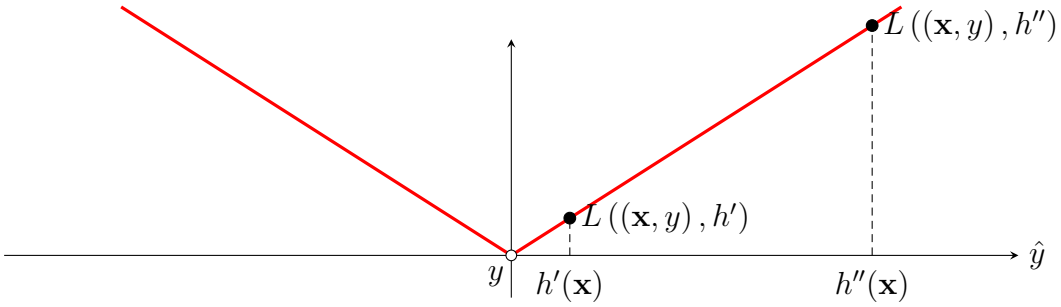


Fig. 57. Para un punto de datos con etiqueta numérica $y \in \mathbb{R}$, el error absoluto $|y - h(\mathbf{x})|$ puede usarse como funcion de perdida para guiar el aprendizaje de una hipótesis h.

pérdida, descenso por subgradiente, ERM.

realización Consideremos una variable aleatoria x que asigna cada elemento (es decir, resultado o evento elemental) $\omega \in \mathcal{P}$ de un espacio de probabilidad $\mathcal{P}$ a un elemento a de un espacio medible $\mathcal{N}$ [2, 6, 71]. Una realización de x es cualquier elemento $a' \in \mathcal{N}$ tal que existe un elemento $\omega' \in \mathcal{P}$ con $x(\omega') = a'$.

Vea también: RV, espacio de probabilidad

recompensa Una recompensa se refiere a alguna cantidad observada (o medida) que nos permite estimar la perdida incurrida por la predicción (o decisión) de una hipótesis $h(\mathbf{x})$. Por ejemplo, en una aplicación de aprendizaje automático 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.

Vea también: pérdida, predicción, hipótesis, ML.

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 aprendizaje federado aprenden una hipótesis local $h^{(i)}$ para cada nodo $i \in \mathcal{V}$, de manera que incurra en una perdida baja sobre su conjnto de datos local.

Vea también: grafo, datos, modelo, dispositivo, conjunto de datos local, modelo local, FL, hipótesis

red neuronal artificial (RNA) Una RNA es una representación gráfica (de flujo de señales) de una función que mapea los 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.

Vea también: atributo, punto de datos, predicción, etiqueta, función de activación.

red profunda Una red profunda es una red neuronal artificial (RNA) con un número (relativamente) grande de capas ocultas. El aprendizaje profundo (deep learning) es un término general para los métodos de aprendizaje automático que utilizan una red profunda como modelo [108].

Vea también: ANN, ML, modelo

reducción de dimensionalidad La reducción de dimensionalidad se refiere a métodos que aprenden una transformación $h : \mathbb{R}^d \to \mathbb{R}^{d'}$ de un conjunto (típicamente grande) de atributos originales $x_1, \ldots, x_d$ a un conjunto más pequeño de atributos informativas $z_1, \ldots, z_{d'}$. Usar un conjunto más pequeño de atributos ofrece múltiples ventajas:

- **Ventaja estadística:** Disminuir la cantidad de atributos reduce usualmente la dimensión efectiva de un modelo, lo cual a su vez disminuye el riesgo de sobreajuste.
- **Ventaja computacional:** Menos atributos implican menos carga computacional durante el entrenamiento de los modelos. Por ejemplo, los métodos de regresión lineal requieren invertir una matriz cuyo tamaño depende del número de atributos.
- **Visualización:** La reducción de dimensionalidad es clave para la visualización de datos. Por ejemplo, se puede aprender una transformación que genere dos atributos z_1, z_2 y emplearlas como coordenadas en un diagrama de dispersión. La Figura 58 ilustra dicho diagrama de dispersión con dígitos manuscritos transformados a partir de imágenes de alta dimensión representadas por valores

de escala de grises.

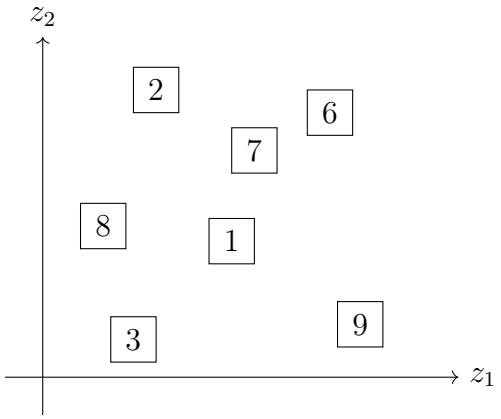


Fig. 58. Ejemplo de reducción de dimensionalidad: Datos de imágenes de alta dimensión (p. ej., dígitos manuscritos) incrustados en 2D mediante atributos aprendidas (z_1, z_2) y visualizados en un diagrama de dispersión.

Véa también: sobreajuste, dimensión efectiva, modelo, diagrama de dispersión.

referencia (baseline) Consideremos un método de aprendizaje automático 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 perdida 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 entrenado. Este valor puede provenir del rendimiento humano, como la tasa de error de dermatólogos

que diagnostican cáncer mediante inspección visual de la piel [117]. Otra fuente de referencia puede ser un método de aprendizaje automático 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 de probabilidad. En muchos casos, dado un modelo de probabilidad $p(\mathbf{x}, y)$, podemos determinar con precisión el minimo riesgo alcanzable entre todas las hipótesis (incluso aquellas que no pertenecen al espacio de hipótesis $\mathcal{H}$) [62]. Este minimo alcanzable se conoce como riesgo de Bayes y corresponde al riesgo para la etiqueta y de un punto de datos, dados sus atributos $\mathbf{x}$. Dado una funcion de perdida específica, el estimador de Bayes (si existe) está completamente determinado por la distribución de probabilidad $p(\mathbf{x}, y)$ [62, Cap. 4]. Calcular el estimador de bayes y el rieso de bayes presenta dos desafíos principales:

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

2) Incluso si se conoce $p(\mathbf{x}, y)$ calcular el riesgo de Bayes puede ser computacionalmente muy costoso [118].

Un modelo de probabilidad ampliamente utilizado es la distribución normal multivariante $(\mathbf{x}, y) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ para puntos de datos caracterizados por atributos y etiquetas numéricas. En este caso, bajo la pérdida de error cuadrático, el estimador de bayes corresponde a la media posterior $\mu_{y|\mathbf{x}}$ de la etiqueta y, dado atributos $\mathbf{x}$ [54,62]. El riesgo

de bayes asociado es la varianza posterior $\sigma^2_{y|\mathbf{x}}$ (see Figure 59).

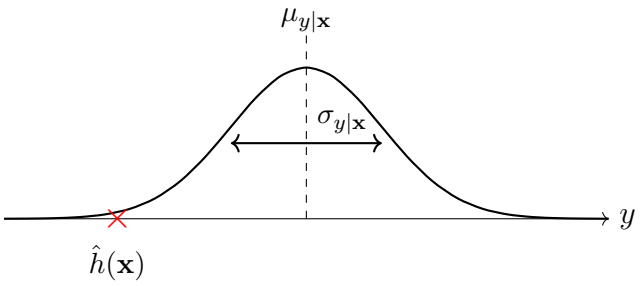


Fig. 59. Si los atributos y la etiqueta de un punto de datos siguen una distribución normal multivariante, podemos alcanzar el minimo riesgo (bajo pérdida de error cuadrático) usando el estimador de bayes $\mu_{y|\mathbf{x}}$ para predecir la etiqueta y de un punto datos con atributos $\mathbf{x}$. El minimo riesgo es dada por la varianza posterior $\sigma^2_{y|\mathbf{x}}$. Podemos usar esta cantidad como baseline para evaluar la perdida promedia del modelo $\hat{h}$ entrenado.

.

Vea también: ML, hipótesis, modelo, pérdida, conjunto de prueba, datos, modelo probabilístico, mínimo, riesgo, espacio de hipótesis, riesgo de Bayes, Bayes estimator, distribución de probabilidad, distribución normal multivariante

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 atributos $\mathbf{x} \in \mathcal{X}$ que resultan en el mismo resultado $h(\mathbf{x}) = a$. Nos referimos a este subconjunto como una región de decisión de la hipótesis h.

Vea también: hipótesis, etiqueta, atributo.

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 [111]. 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 aprendizaje automático. Las disposiciones clave incluyen:

- Principio de minimización de datos: los sistemas de aprendizaje automático deben utilizar únicamente la cantidad necesaria de datos personal para su propósito.
- Transparencia y explicabilidad: los sistemas de aprendizaje automático deben permitir a sus usuarios comprender cómo se toman las decisiones que los afectan.
- Derechos del titular de los 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 los atributos de un punto de datos [8, Ch. 2].

Vea también: etiqueta, atributo, punto de datos

regresión de Huber La regresión de Huber se refiere a métodos basados en ERM que utilizan la perdida 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 perdida de Huber permite al usuario balancear la robustez del pérdida por error absoluto frente a los beneficios computacionales de la error cuadrática media suave.

Vea también: regresión, ERM, pérdida de Huber, predicción, regresión, regresión por desviación absoluta mínima, regresión lineal, pérdida por error absoluto, 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 error cuadrado medio incurrido en un conjunto de puntos de datos etiquetados, al que nos referimos como el conjunto de datos.

Vea también: regresión, hipótesis, etiqueta, atributo, punto de datos, pérdida de error cuadrático, punto de dato etiquetado, 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 puntos de etiquetados (es decir, el conjunto de

entrenamiento).

Vea también: regresión, hipótesis, clasificador, etiqueta, vector de atributos, punto de datos, pérdida logística, punto de dato etiquetado, conjunto de entrenamiento.

regresión polinómica La regresión polinómica es una instancia de ERM que aprende un mapa de hipótesis polinómica para predecir una etiqueta numérica a partir de los atributos numéricos de un punto de datos. Para puntos de datos caracterizados por un único atributo numérico, la regresión polinómica utiliza el espacio de hipótesis

$$\mathcal{H}_d^{(\text{poly})} := \left\{ h(x) = \sum_{j=0}^{d-1} x^j w_j \right\}.$$

La calidad de una hipótesis polinómica se mide mediante la perdida de error cuadrático promedio incurrida en un conjunto de puntos de datos etiquetados (al que nos referimos como el conjunto de entrenamiento).

Vea también: regresión, hipótesis, etiqueta, atributo, punto de datos, espacio de hipótesis, pérdida de error cuadrático, punto de dato etiquetado, conjunto de entrenamiento.

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

Vea también: ERM, pérdida por error absoluto, 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 parametros del modelo $\mathbf{w}$ se mide por la suma de dos componentes. El primer componente es el promedio de error cuadrado medio incurrido por $h^{(\mathbf{w})}$ en un conjunto de puntos de datos 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 regulación $\alpha > 0$. Agregar $\alpha\|\mathbf{w}\|_2^2$ al promedio de medio cuadrado medio es equivalente a reemplazar cada punto de datos original por la realización de (infinitos) variables aleatorias iid centrados alrededor de estos puntos de datos (vea regulación).

Vea también: regresión, pesos, hipótesis, parámetros del modelo, pérdida de error cuadrático, punto de dato etiquetado, conjunto de entrenamiento, norma, regularización, punto de datos, realización, i.i.d., RV.

regularización Un desafío clave de las aplicaciones modernas de aprendizaje automático es que a menudo usan modelos grandes, que tienen un dimension efectiva del orden de miles de millones. Entrenar un modelo de alta dimensión utilizando métodos basados en ERM es propenso al sobreajuste, es decir, la hipótesis aprendida tiene un buen rendimiento en el conjunto de entrenamiento pero un rendimiento pobre fuera del conjunto de entrenamiento. La regularización se refiere a modificaciones de una instancia dada de ERM para evitar el sobreajuste, es decir, para asegurar que la hipótesis aprendida no tenga un rendimiento mucho peor fuera del conjunto de entrenamiento. Existen tres rutas para implementar la regularización:

1) Modelo podado: Podamos el modelo original $\mathcal{H}$ para obtener un modelo más pequeño $\mathcal{H}'$. Para un modelo paramétrico, el podado puede implementarse mediante restricciones en los parametros del modelo (como $w_1 \in [0.4, 0.6]$ para el peso del atributo x_1 en regresin lineal).

2) Perdida penalización: Modificamos la funcion objetivo de ERM añadiendo un término de penalización al error de entrenamiento. El término de penalización estima cuánto mayor es la perdida esperada (o riesgo) en comparación con la perdida promedio en el conjunto de entrenamiento.

3) Aumentación de Datos: Podemos ampliar el conjunto de entrenamiento $\mathcal{D}$ añadiendo copias perturbadas de los puntos de datos originales en $\mathcal{D}$. Un ejemplo de tal perturbación es añadir la realización de una variable aleatoria al vector de atributos de un punto de datos.

La Fig. 60 ilustra las tres rutas anteriores para la regularización. Estas rutas están estrechamente relacionadas y, a veces, son completamente equivalentes. La aumentación de datos usando variables aleatorias gaussianas para perturbar los vectores de atributos en el conjunto de entrenamiento de regresion lineal tiene el mismo efecto que añadir la 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 más fácil implementar aumentación de datos que el podado de modelo.

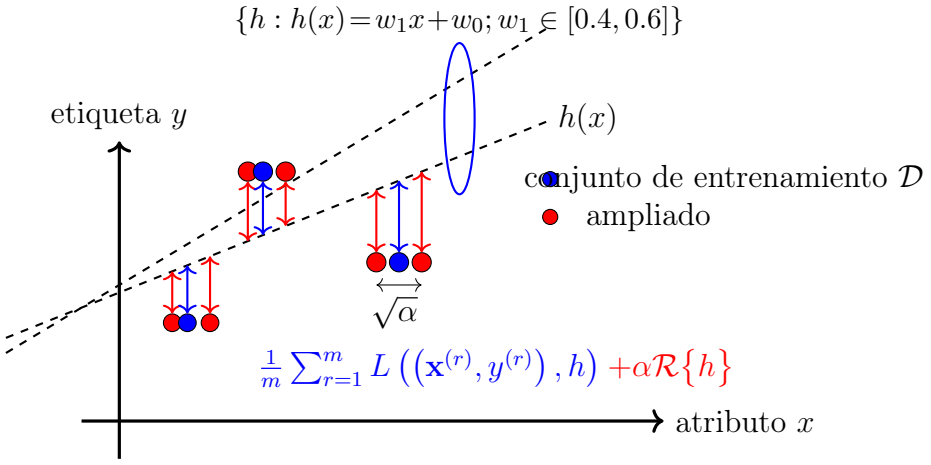


Fig. 60. Tres enfoques para la regularización: 1) Aumentación de datos; 2) penalización de perdida; y 3) modelo podado (mediante restricciones en parametros del modelo).

Véa también: ML, modelo, dimensión efectiva, ERM, sobreajuste, hipótesis, conjunto de entrenamiento, parámetros del modelo, atributo, regresión lineal, pérdida, función objetivo, error de entrenamiento, riesgo, aumentación de datos, punto de datos, realización, RV, vector de atributos, VA gaussiana, regresión ridge, etiqueta.

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 prediccion en un conjunto de entrenamiento de sus errores de predicción en puntos de datos fuera del conjnto de entrenamiento. La regresión ridge utiliza el regularizador $\mathcal{R}\{h\} := \|\mathbf{w}\|_2^2$ para mapas de hipótesis lineales $h^{(\mathbf{w})}(\mathbf{x}) := \mathbf{w}^T\mathbf{x}$ [8, Ch. 3]. El operador de reducción y selección absoluta mínima (Lasso) utiliza el regularizador

$\mathcal{R}\{h\} := \|\mathbf{w}\|_1$ para mapas de hipótesis lineales $h^{(\mathbf{w})}(\mathbf{x}) := \mathbf{w}^T\mathbf{x}$ [8, Ch. 3].

Vea también: hipótesis, espacio de hipótesis, predicción, conjunto de entrenamiento, punto de datos, conjunto de entrenamiento, Regresión ridge, Lasso

divergencia de Rényi La divergencia de Rényi mide la (dis)similitud entre dos distribuciones de probabilidad [119].

Retropropagación Backpropagation es un algoritmo para calcular el gradiente $\nabla_{\mathbf{w}} f(\mathbf{w})$ de una funcion objetivo $f(\mathbf{w})$ que depende de los parametros del modelo $\mathbf{w}$ de una red artificial neuronal. Un ejemplo de tal función de objetivo es la perdida promedio incurrida por la red artificial neuronal en un lote de puntos de datos. Este algoritmo es una aplicación directa de la regla de la cadena del cálculo diferencial para calcular de manera eficiente las derivadas parciales de la función de perdida con respecto a los parametros del modelo.

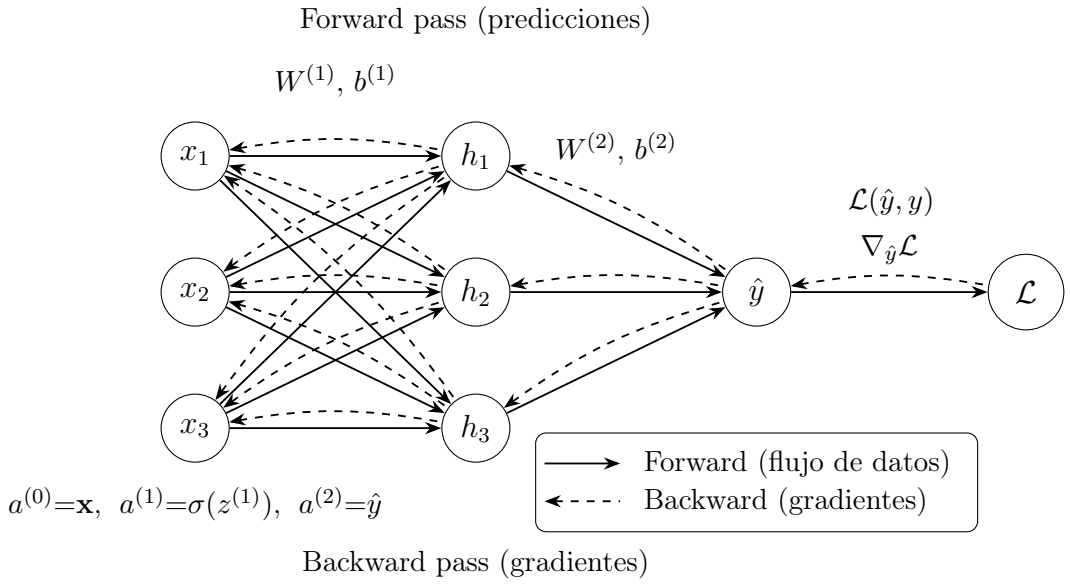


Fig. 61. Las flechas sólidas muestran la pasada hacia adelante (flujo de datos y cálculo de la pérdida), las flechas discontinuas muestran el flujo de corrección de gradientes durante la pasada hacia atrás para actualizar los parámetros $W^{(x)}$, $b^{(x)}$.

Backpropagation consta de dos fases consecutivas: 1) Forward pass: un lote de puntos de datos se introduce en la red artificial neuronal, la cual procesa la entrada a través de sus capas usando sus pesos actuales, produciendo finalmente una predicción en su salida. La predicción del lote se compara con la etiqueta verdadera mediante una función de perdida, que cuantifica el error de la predicción. 2) Backward pass (backpropagation): el error se retropropaga a través de las capas de la ANN. Las derivadas parciales obtenidas con respecto a los parametros $w_1, \ldots, w_d$ constituyen el gradiente $\nabla f(\mathbf{w})$, el cual puede emplearse, a

su vez, para implementar un paso de gradiente

Véa también: GD, función de pérdida, ANN, método de optimización.

riesgo Consideremos una hipótesis$h$ que se utiliza para predecir la etiqueta y de un punto de datgos a partir de sus atributos $\mathbf{x}$. Evaluamos la calidad de una predicción específica usando una funcion de perdida $L\left((\mathbf{x}, y), h\right)$. Si interpretamos los puntos de datos como realizaciónes de variabels aleatorias iid, entonces $L\left((\mathbf{x}, y), h\right)$ también se convierte en una realización de una variable aleatoria. La suposición de independencia e idéntica distribución nos permite definir el riesgo de una hipótesis como la expectativa de la perdida $\mathbb{E}\big\{L\left((\mathbf{x}, y), h\right)\big\}$. El riesgo de h depende tanto de la funcion de perdida elegida como de la distribución de probabilidad de los puntos de datos.

Vea también: hipótesis, etiqueta, punto de datos, atributo, predicción, función de pérdida, realización, RV, i.i.d., glsrealization, RV, suposición i.i.d., esperanza, pérdida, distribución de probabilidad

riesgo de Bayes Considera un modelo de probabilidad 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 minimo posible que puede alcanzarse por cualquier hipótesis $h : \mathcal{X} \to \mathcal{Y}$. Cualquier hipótesis que alcance el riesgo de Bayes se denomina un estimador de Bayes [62].

Vea también: modelo probabilístico, distribución de probabilidad, atributo, etiqueta, punto de datos, riesgo, mínimo, hipótesis, Bayes estimator.

riesgo empírico El rieso empírico empírico $\widehat{L}\big(h|\mathcal{D}\big)$ de una hipótesis sobre

un conjunto de datos $\mathcal{D}$ es la perdida promedio incurrida por h al aplicarse a los puntos de datos en el $\mathcal{D}$.

Vea también: riesgo, hipótesis, conjunto de datos, pérdida, punto de datos

robustez La robustez es un requisito clave para la IA confiable. Se refiere a la propiedad de un sistema de aprendizaje automático de mantener un rendimiento aceptable incluso cuando se somete a diferentes formas de perturbaciones. Estas perturbaciones pueden afectar los atributos de un punto de datos con el fin de manipular la prediccion producida por un modelo de aprendizaje automático ya entrenado. La robustez también abarca la estabilidad de los métodos basados en ERM frente a perturbaciones en el conjunto de entrenamiento. Estas perturbaciones pueden ocurrir como parte de ataque de envenenamiento de datos.

Véa también: trustworthy AI, ML, atributo, punto de datos, predicción, modelo, stability, ERM, conjunto de entrenamiento, envenenamiento de datos, ataque.

régimen de alta dimensión El régimen de alta dimensión de minimización empírica del riesgo se caracteriza por que la dimensión efectiva del modelo es mayor que el tamaño de la muestra, es decir, el número de puntos de datos 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 puntos de datos excede el número de puntos de datos en el conjunto de entrenamiento. Otro ejemplo de métodos de aprendizaje automático

que operan en el régimen de alta dimensión son las red neuronal artifical grandes, que tienen muchos más pesos ajustables (y términos de sesgo) que el número total de puntos de datos 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 aprendizaje automático en el régimen de alta dimensión [52, 120].

Vea también: ERM, dimensión efectiva, modelo, tamaño de la muestra, punto de datos, conjunto de entrenamiento, regresión lineal, atributo, ML, ANN, pesos, probabilidad.

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 vectores de atributos $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}$, 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 [96].

Vea también: kernel, vector de atributos

sesgo Consideremos un método de aprendizaje automático que utiliza un espacio de hipótesis $\mathcal{H}$ parametrizado. Este aprende los parametros del modelo $\mathbf{w} \in \mathbb{R}^d$ utilizando el conjunto de datos

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

Para analizar las propiedades del método de aprendizaje automático, típicamente interpretamos los puntos de datos como realizaciónes de

iid. variables aleatorias,

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

Entonces podemos interpetar el método de aprendizaje automático 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 $B^2 := \big\|\mathbb{E}\{\widehat{\mathbf{w}}\} - \overline{\mathbf{w}}\big\|_2^2$.
Vea también: ML, espacio de hipótesis, parámetros del modelo, punto de datos, realización, i.i.d., RV, ERM

sobreajuste Consideremos un método de aprendizaje automático que utiliza ERM para aprender una hipótesis con el minimo riesgo empírico en un conjunto de entrenamiento dado. Dicho método presenta sobreajuste del conjunto de entrenamiento si aprende una hipótesis con un rieso empírico pequeño sobre el conjunto de entrenamiento pero una perdida significativamente mayor fuera de él conjunto de entrenamiento.
Vea también: ML, ERM, hipótesis, mínimo, riesgo empírico, conjunto de entrenamiento, pérdida

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 $\mathbf{w} \in \mathbb{R}^d$ [15], [121]. 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 metodos

de descenso de gradiente. De hecho, los metodos de descenso de gradiente aproximan la funcion 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. 62).

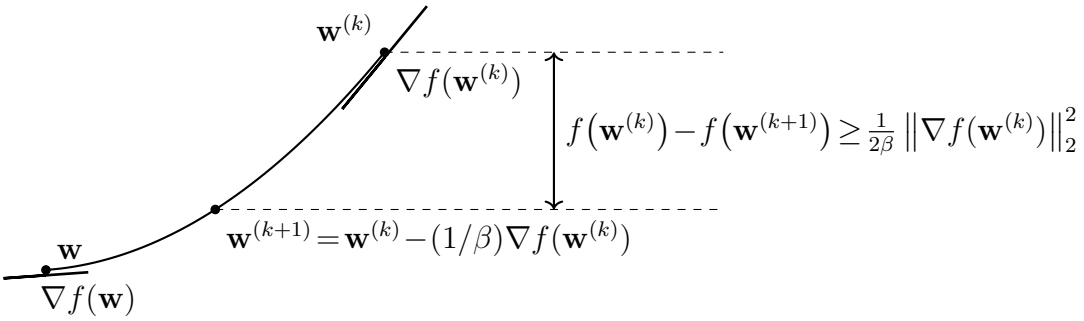


Fig. 62. Considere un funcion de objetivo $f(\mathbf{w})$ que es β-suave. Tomar un paso de gradiente, con tamaño de paso $\eta = 1/\beta$, disminuye el objetivo por al menos $\frac{1}{2\beta} \left\|\nabla f(\mathbf{w}^{(k)})\right\|_2^2$ [15], [121], [85]. Nótese que el tamaño de paso $\eta = 1/\beta$ se hace más grande para un β más pequeño. Por lo tanto, para funciones de objetivo más suaves (es decir, aquellas con un β más pequeño), podemos tomar pasos más grandes.

Vea tambien: diferenciable, gradiente, función objetivo, métodos de gradiente, paso de gradiente, tamaño de paso.

subajuste Consideremos un método de aprendizaje automatico que utiliza ERM para aprender una hipótesis con el minimo 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 rieso empírico suficientemente pequeño sobre el conjunto de entrenamiento. Si un método sufre de subajuste, típicamente tampoco podrá aprender una hipótesis con un riesgo pequeño.

Vea también: ML, ERM, hipótesis, mínimo, riesgo empírico, conjunto de entrenamiento, riesgo

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 $\mathbf{w}'$ [16, 61].

suposición de agrupamiento La suposición de agrupamiento postula que los puntos de datos en un conjunto de datos forman un (pequeño) número de grupos o clústers. Los puntos de datos en el mismo clúster son más similares entre sí que aquellos que están fuera del clúster [34]. Obtenemos diferentes métodos de agrupamiento utilizando diferentes nociones de similitud entre puntos de datos.

Vea también:punto de datos, conjunto de datos, cluster, agrupamiento

suposición de independencia e idéntica distribución (suposición i.i.d.) La suposición iid interpreta los puntos de datos de un conjunto de datos como las realizaciónes de variables aleatorias iid.

Vea también: i.i.d., punto de datos, conjunto de datos, realización, RV.

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 puntos de datos individuales contenidos en un conjunto de datos.

Vea también: punto de datos, conjunto de datos.

tamaño de paso Consulte tasa de aprendizaje.

tarea de aprendizaje Considere un conjunto de datos $\mathcal{D}$ que consiste en múltiples puntos de datos $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}$. Por ejemplo, $\mathcal{D}$ podría representar una colección de imágenes en una base de datos. Una tarea de aprendizaje se define especificando aquellas propiedades (o atributos) de un punto de datos que se usan como sus atributos y etiquetas. Dada una elección de modelo $\mathcal{H}$ y funcion de perdida, una tarea de aprendizaje conduce a una instancia de ERM y puede representarse mediante la funcion de objetivo asociada $\widehat{L}\big(h|\mathcal{D}\big)$ para $h \in \mathcal{H}$. Es importante destacar que se pueden construir múltiples tareas de aprendizaje distintas a partir del mismo conjunto de datos seleccionando diferentes conjuntos de atributos y etiquetas.

Una imagen que muestra vacas pastando en el campo austriaco.

Tarea 1 (Regresión):

- Atributos: Los valores RGB de todos los píxeles de la imagen.
- Etiqueta: El número de vacas representadas.

Tarea 2 (Clasificación):

- Atributos: La intensidad promedio de verde en la imagen.
- Etiqueta: ¿Deben moverse las vacas a otra ubicación (sí/no)?

Fig. 63. Dos tareas de aprendizaje construidas a partir de un solo conjunto de datos de imágenes. Estas tareas difieren en la selección de atributo y en la elección de etiqueta (es decir, el objetivo), pero ambas se derivan del mismo conjunto de datos.

. Tales tareas están inherentemente relacionadas, y resolverlas conjuntamente, por ejemplo usando métodos de aprendizaje multitarea, suele

ser más eficaz que tratarlas de forma independiente [122], [123], [124]. Vea también: conjunto de datos, modelo, función de pérdida, función objetivo, aprendizaje multitarea, espacio de etiquetas.

tasa de aprendizaje Considere un método iterativo de aprendizaje automático 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 descenso por gradiente y sus variantes, descenso por gradiente estocástico y descenso por gradiente 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 descenso por gradiente [8, Ch. 5].

Vea también: ML, hipótesis, GD, SGD, descenso por gradiente proyectado (GD proyectado), tamaño de paso.

teorema central del límite (CLT) Consideremos una secuencia de variable aleatoria iid $x^{(r)}$, para $r = 1, 2, \ldots$, cada una con media cero y varianza finita $\sigma^2 > 0$. El teorema central del límite (CLT) establece que la suma normalizada

$$s^{(m)} := \frac{1}{\sqrt{m}} \sum_{r=1}^{m} x^{(r)}$$

converge en distribución a una variable aleatoria gaussiana con media cero y varianza σ^2 cuando $m \to \infty$ [125, Proposition 2.17]. Una forma

elegante de derivar el CLT es mediante la funcion característica de la suma normalizada $s^{(m)}$. Sea $\phi(t) = \mathbb{E}\{\exp(jtx)\}$ (donde $j = \sqrt{-1}$ es la unidad imaginaria) la funcion característica común de cada término $x^{(r)}$, y sea $\phi^{(m)}(t)$ la funcion característica de $s^{(m)}$. Definimos un operador $\mathcal{T}$ que actúa sobre funciones de característica tal que

$$\phi^{(m)}(t) = \mathcal{T}(\phi^{(m-1)})(t) := \phi\left(\frac{t}{\sqrt{m}}\right) \cdot \phi^{(m-1)}\left(\frac{\sqrt{m-1}}{\sqrt{m}}t\right).$$

Esta iteracion de punto fijo captura el efecto de añadir recursivamente una variable aleatoria iid $\mathbf{x}^{(m)}$ y reescalar. Al aplicar $\mathcal{T}$ iterativamente, se obtiene la convergencia de $\phi^{(m)}(t)$ hacia el punto fijo

$$\phi^*(t) = \exp(-t^2\sigma^2/2)$$

que es la funcion característica de una variable aleatoria gaussiana con media cero y varianza σ^2. Las generalizaciones del CLT permiten variables aleatorias dependientes o no idénticamente distribuidas [125, Sec. 2.8].

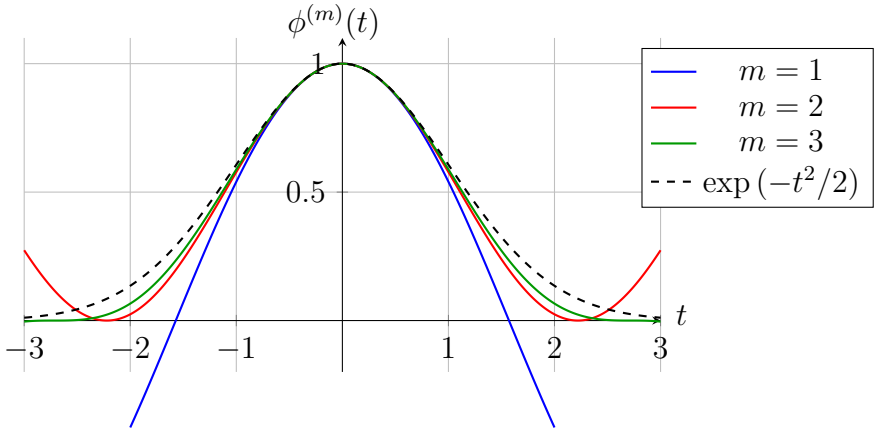


Fig. 64. Funciones característica de sumas normalizadas de variables aleatorias iid $x^{(r)} \in \{-1, 1\}$ para $r = 1, \ldots, m$, comparadas con el límite gaussiano.

Vea tambien: RV, VA gaussiana.

transparencia La transparencia es un requisito fundamental para una inteligencia artificial confiable confiable [126]. En aprendizaje automático, suele utilizarse como sinónimo de explicabilidad [76, 127] pero en el contexto más amplio de sistemas de inteligencia artificial, incluye también información sobre limitaciones, confiabilidad y uso previsto. En sistemas de diagnóstico médico, se requiere informar el nivel de confianza de una prediccion. En aplicaciones financieras como la puntuación crediticia, las decisiones automatizadas basadas en inteligencia artificial 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 aprendizaje automático ofrecen transparencia de manera intrínseca. Por ejemplo, la regresión logística permite interpretar la fiabilidad de una clasificación mediante el valor absoluto $|h(\mathbf{x})|$. Los árboles de decisión, también son consideradas transparentes porque generan reglas comprensibles para los humanos. [30]. La transparencia también requiere que se informe explícitamente cuando una persona está interactuando con un sistema de inteligencia artificial. Por ejemplo, los chatbots impulsados por inteligencia artifical deben indicar claramente que no son humanos. 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 [48] y las tarjetas de sistemas de

inteligencia artifical [128], que ayudan a los desarrolladores y usuarios a entender las limitaciones y aplicaciones adecuadas de un sistema de inteligencia artificial [129].

Vea también: trustworthy AI, ML, AI, predicción, regresión logística, clasificación, árbol de decisión, modelo

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

Vea también: función de activación, ANN

validación La validación se refiere a la práctica de evaluar la perdida incurrida por una hipótesis $\hat{h}$ que ha sido aprendida mediante algún método de aprendizaje automático, por ejemplo, resolviendo ERM en un conjunto de entrenamiento $\mathcal{D}$. La validación implica evaluar el desempeño de la hipótesis en un conjunto de puntos de datos que no están contenidos en el conjunto de entrenamiento $\mathcal{D}$.

Vea también: pérdida, hipótesis, ML, ERM, conjunto de entrenamiento, punto de datos

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 validacion. 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 errores de validación obtenidos desde las k repeticiones.

Vea también: hipótesis, conjunto de datos, modelo, ERM, validación, conjunto de validación, conjunto de entrenamiento, error de validación.

valor atípico (outlier) Muchos métodos de aprendizaje Automático están motivados por la suposición de independencia e idéntica distribución, que interpreta los puntos de datos como realizaciónes de variables aleatorias iid con una distribución de probabilidad común. La suposición de independencia e idéntica distribución es útil en aplicaciones donde las propiedades estadísticas del proceso de generación de datos son estacionarias (o invariantes en el tiempo) [130]. Sin embargo, en algunas aplicaciones, los datos están compuestos por una mayoría de puntos de datos regulares que cumplen con la suposición de independencia e idéntica distribución y un pequeño número de puntos de datos que presentan propiedades estadísticas fundamentalmente diferentes en comparación con los puntos de datos 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 [131, 132].

Vea también: ML, suposición i.i.d., punto de datos, realización, i.i.d.,

RV, distribución de probabilidad, datos.

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 espacio de probabilidad $\mathcal{P}$ a un espacio de valores [6, 54]. El espacio de probabilidad 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 [6].

Vea también: espacio de probabilidad, probabilidad, espacio euclidiano,

variable aleatoria gaussiana (VA gaussiana) Una variable aleatoria gaussiana estándar es una variable aleatoria real x con función de densidad de probabilidad (pdf) [7, 54, 65]

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

Dada una variable aleatoria gaussiana estándar x, podemos construir una variable aleatoria gaussiana general x' con media μ y varianza σ^2 mediante $x' := \sigma(x + \mu)$. La distribución de probabilidad de una variable aleatoria 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 iid gaussianas estándar variables aleatorias $z_1, \ldots, z_d$.

Los procesos aleatorios gaussianos generalizan los vectores aleatorios gaussianos aplicando transformaciones lineales a a secuencias infinitas de variables aleatorias gaussianas estándar [133].

Las variables aleatorias gaussianas se utilizan ampliamente como modelos de probabilidad en el análisis estadístico de métodos de aprendizaje automático. Su importancia se debe, en parte, al teorema del límite central, que establece que el promedio de un número creciente de variables aleatorias independientes (aunque no sean gaussianas) converge a una variable aleatoria gaussiana [64].

The distribución normal multivariante is also distinct in that it represents maximum incertidumbre: Among all vector-valued RVs with a given matriz de covarianza $\mathbf{C}$, the RV $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{C})$ maximizes entropía diferencial [57, Th. 8.6.5]. This makes GPs a natural choice for capturing incertidumbre (or lack of knowledge) in the absence of additional structural information.

Vea también: RV, pdf, media, varianza, distribución de probabilidad,

matriz de covarianza, i.i.d., ML

variación total Vea GTV.

variación total generalizada (GTV) GTV es una medida de la variación de los modelos locales entrenados $h^{(i)}$ (o de sus parametros 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 $\{i, i'\} \in \mathcal{E}$.

Vea también: modelo local, parámetros del modelo, grafo, discrepancia, hipótesis

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

Vea también: RV, esperanza

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 puntos de datos se caracterizan por vector de atributos numéricos, podemos usar la distancia euclidiana como medida. the

metric.

Vea también: hipótesis, vecinos, conjunto de datos, punto de datos, vector de atributos

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

Vea también: red FL

Vector de Activación de Concepto (CAV) Los CAVs [134] son una técnica de explicabilidad para investigar y mejorar las RNA. A diferencia de los métodos tradicionales que analizan la influencia de los atributos de entrada en la prediccion de un modelo, las explicaciones basadas en conceptos buscan lograr explicabilidad en términos de conceptos comprensibles para humanos. Como ejemplo, consideremos el concepto de "rayas" en un problema de clasificación multietiqueta aplicado a imágenes de animales. Para aplicar este enfoque, se reúne un conjunto de datos que ilustra un concepto C definido por el usuario, junto con otro conjuntos de datos de ejemplos "no-concepto" o "aleatorios", que se dividirá en sus respectivos conjuntos de entrenamiento, conjunto de validacion y conjuntos de prueba. Posteriormente, se entrena un clase lineal binario, por ejemplo un SVM, para separar los vectores de atributos obtenidos como las activaciones de la capa latente l de los puntos de datos de entrada correspondientes a ejemplos concepto y no-concepto. De este modo, se obtiene una hipótesis v_C^l (un vector del mismo tamaño que la dimensión de la capa oculta l) ortogonal al

hiperplano separador del clasificador (por convención orientado hacia los ejemplos de concepto), al cual se denomina Vector de Activación de Concepto (CAV) en este contexto. En primer lugar, si este clasificador alcanza una alta acc en alguna capa l, esto indica que la red neuronal ha codificado internamente el concepto, ya que puede separarlo linealmente de los ejemplos no-concepto. Además, esto permite localizar capas o incluso neuronas específicas que han asumido esta tarea. En segundo lugar, esta técnica permite cuantificar la influencia del concepto en la predicción del modelo, es decir, en la probabilidad asignada a una etiqueta concreta. Más precisamente, esto es posible con la técnica Testing with Concept Activation Vectors (TCAV), donde se calculan derivadas direccionales de la predicción en la dirección de v_C^l en la capa l (bajo supuestos suficientes de suave, pueden calcularse como productos internos con los gradientes). Aquí, valores positivos indican que la presencia de un concepto incrementa la probabilidad de la etiqueta (por ejemplo, se espera que el concepto de "rayas" tenga una contribución positiva en la predicción de la etiqueta "cebra"), valores negativos contradicen la predicción, y valores cercanos a cero significan que el concepto es neutro para la clase de interés. Los CAVs resultan interesantes por varias razones. Ayudan a responder preguntas como: ¿utilizan las redes neuronales internamente conceptos comprensibles por humanos? En el caso de modelos altamente precisos, ¿podemos incluso extraer conceptos que eran previamente desconocidos para los expertos humanos? Además, los CAVs son útiles para eliminar sesgos en redes neuronales en los casos en que realizan predicciones basadas

en conceptos indeseables. Existen diversas variantes de este enfoque, y sigue siendo un área activa de investigación.

Véa también: modelo lineal, red profunda, trustworthy AI, interpretabilidad, transparencia.

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

Vea también: ML, espacio euclidiano, ML, método de kernel

vector normal estándar Un vector normal estándar es un vector aleatorio $\mathbf{x} = \left(x_1, \ldots, x_d\right)^T$ cuyas componentes son variables aleatorias gaussianas iid $x_j \sim \mathcal{N}(0, 1)$. Es un caso particular de una distribución normal multivariante, dado que $\mathbf{x} \sim (\mathbf{0}, \mathbf{I})$.

Véa también: i.i.d., VA gaussiana, distribución normal multivariante, RV.

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.

Vea también: valor propio.

0/1 loss La perdida 0/1 $L^{(0/1)}\left((\mathbf{x}, y), h\right)$ mide la calidad de un clasificador

$h(\mathbf{x})$ que genera una prediccion $\hat{y}$ (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 prediccion es correcta, es decir, $L^{(0/1)}\left((\mathbf{x}, y), h\right) = 0$ cuando $\hat{y} = y$. Es igual a 1 si la prediccion es incorrecta, es decir, $L^{(0/1)}\left((\mathbf{x}, y), h\right) = 1$ cuando $\hat{y} \neq y$.

Vea también: pérdida,clasificador, predicción,etiqueta, punto de datos,atributo

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 un punto 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 datos termina en un nodo hoja (que no tiene nodos hijos).

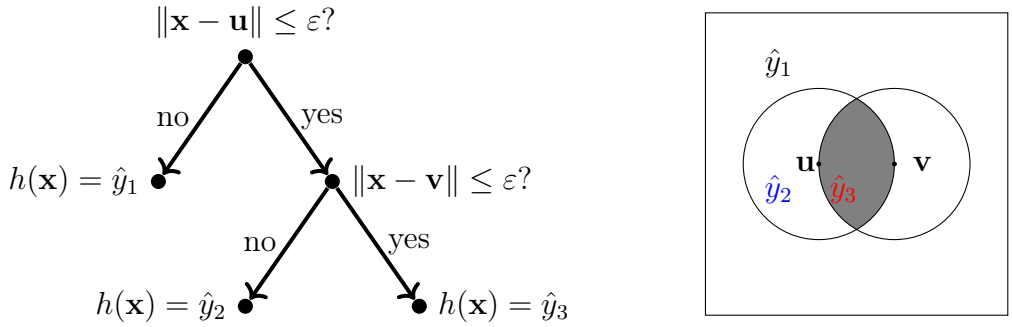


Fig. 65. Izquierda: Un árbol de decisión es una representación similar a un diagrama de flujo de una hipótesis $h : \mathcal{X} \to \mathbb{R}$ constante por partes. Cada parte es una región de decisión $\mathcal{R}_{\hat{y}} := \{\mathbf{x} \in \mathcal{X} : h(\mathbf{x}) = \hat{y}\}$. TEl árbol de decisión mostrado puede aplicarse a vector de atributos 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 region de decisiones. Cada region de decision $\mathcal{R}_{\hat{y}} \subseteq \mathcal{X}$ corresponde a un nodo hoja específico en el árbol de decisión.

Vea también: hipótesis, grafo, vector de atributos, punto de datos, atributo, región de decisión, espacio de atributos.

Index

0/1 loss, 198

k-means, 28

agrupamiento, 28

agrupamiento basado en densidad para aplicaciones espaciales con ruido (DBSCAN), 29

agrupamiento basado en flujo, 29

agrupamiento convexo, 29

agrupamiento en grafos, 30

agrupamiento espectral, 30

agrupamiento rígido, 32

agrupamiento suave, 32

algoritmo, 32

algoritmo distribuido, 33

algoritmo en línea, 34

algoritmo estocástico, 35

análisis de componentes principales (PCA), 35

análisis de componentes principales probabilístico (PPCA), 35

aplicación, 19

aplicación lineal, 36

aprendizaje automático (ML), 37

aprendizaje automático explicable (explainable ML), 37

aprendizaje de atributos, 37

aprendizaje en línea, 38

aprendizaje federado (FL), 42

aprendizaje federado agrupado (CFL), 39

aprendizaje federado en red (NFL), 39

aprendizaje federado horizontal (horizontal FL), 40

aprendizaje federado vertical (VFL), 40

aprendizaje multitarea, 42

aprendizaje por refuerzo (RL), 42

aprendizaje semi-supervisado (SSL), 43

arrepentimiento (regret), 44

aspectos computacionales, 44

aspectos estadísticos, 44

ataque, 44

ataque a la privacidad, 45

ataque de denegación de servicio

(DoS), 45
atención, 46
atributo, 47
atributo sensible, 47
aumentación de datos, 48
autoencoder, 49

backpropagation, 178
bagging, 49
bandido multi-brazo (MAB), 50
boosting, 52
bootstrap, 53
bosque aleatorio, 54
brecha de generalización, 54

caracterización min-max de Courant–Fischer–Weyl, 54
clasificador, 55
clasificador lineal, 56
classification, 55
clúster, 56
condición de gradiente cero, 57
conectividad algebraica, 58
conjunto de datos, 58
conjunto de datos local, 60
conjunto de entrenamiento, 61
conjunto de prueba, 61
conjunto de validación, 61
contraction operator, 62
convergence, 19
convexo, 62
cota superior de confianza (UCB), 62
covarianza, 63
criterio de parada, 64

datos, 64
datos en red, 65
datos faltantes, 65
deep net, 168
descenso de gradiente estocástico (SGD), 66
descenso por gradiente (GD), 67
descenso por gradiente en línea (online GD), 68
descenso por gradiente proyectado (GD proyectado), 70
descenso por subgradiente, 71
descomposición en valores propios (EVD), 72
descomposición en valores singulares (SVD), 72
desigualdad de concentración, 72

determinante, 19
diagrama de dispersión, 72
diferenciable, 73
differential privacy (DP), 153
dimensión de Vapnik–Chervonenkis (dimensión VC), 73
dimensión efectiva, 74
discrepancia, 74
dispositivo, 74
distribución de probabilidad, 74
distribución normal multivariante, 75
divergencia de Kullback-Leibler (divergencia KL), 76
divergencia de Rényi, 178

entorno, 76
entropía diferencial, 77
envenenamiento de datos, 77
epígrafo, 79
error cuadrático medio de estimación (MSEE), 79
error de entrenamiento, 80
error de estimación, 80
error de validación, 81
espacio de atributos, 81
espacio de etiquetas, 82
espacio de Hilbert, 82
espacio de hipótesis, 83
espacio de parámetros, 84
espacio de probabilidad, 85
espacio euclidiano, 86
espacio muestral, 86
espacio vectorial, 20
espectrograma, 88
esperanza-maximización (EM), 90
estabilidad, 91
estimador de Bayes, 92
estocástico, 93
etiqueta, 93
event, 93
expectation, 89
experimento aleatorio, 94
experto, 96
explanation, 97
explicabilidad, 96
Explicaciones Locales Interpretables e Independientes del Modelo (LIME), 98

familias exponenciales en red (nExpFam), 99
FedAvg, 100
FedGD, 100
FedProx, 101
FedRelax, 101
FedSGD, 101
frontera de decisión, 102
función, 22
función característica, 22
función cuadrática, 102
función de activación, 103
función de densidad de probabilidad (pdf), 103
función de pérdida, 103
función objetivo, 104

generalización, 105
gradient step, 149
gradiente, 108
grado de nodo, 108
grado de pertenencia, 108
grafo conexo, 108
grafo de similitud, 109
graph, 108

hipótesis, 109
histograma, 110

incertidumbre, 111
independientes e idénticamente distribuidos (i.i.d.), 111
información mutua (MI), 111
Instituto Meteorológico de Finlandia (FMI), 112
inteligencia artificial (IA), 112
inteligencia artificial confiable (IA confiable), 113
interfaz de programación de aplicaciones (API), 113
interpretabilidad, 114
inversión de modelo, 116

kernel, 118

ley de los grandes números, 119
loperador de reducción y selección absoluta mínima (Lasso), 145
lote, 120

mapa de atributos, 120
matriz, 23
matriz de atributos, 121
matriz de confusión, 121

matriz de covarianza, 122
matriz de covarianza muestral, 122
matriz inversa, 122
matriz laplaciana, 123
measurable, 127
media, 124
media muestral, 125
mediana, 125
mediana geométrica (GM), 126
minimización de la pérdida regularizada (RLM), 129
minimización de variación total generalizada (GTVMin), 129
minimización del riesgo empírico explicable (EERM), 129
minimización del riesgo empírico regularizado (RERM), 130
minimización del riesgo estructural (SRM), 130
minimización empírica del riesgo (ERM), 131
modelo de lenguaje de gran escala (LLM), 134
modelo de mezcla gaussiana (GMM), 135
modelo en red, 135
modelo estocástico de bloques (SBM), 135
modelo lineal, 136
modelo local, 138
modelo probabilístico, 138
muestra, 139
multi-label classification, 55
máquina de vectores de soporte (SVM), 139
máxima verosimilitud, 140
máximo, 141
método de Jacobi, 117
método de kernel, 141
Método de Newton, 24
método de optimización, 142
métodos de gradiente, 142
métrica, 143
mínimo, 143

no suave, 143
norma, 144
normalización de datos, 144
nullspace, 86
número de condición, 144

operador proximal, 145
optimismo ante la incertidumbre, 146
overfitting, 183

parámetro, 148
parámetros, 149
parámetros del modelo, 149
peso de arista, 151
precisión (accuracy), 152
prediction, 152
predictor, 153
preimagen, 153
principio de minimización de datos, 153
privacy funnel, 76
privacy leakage, 101
probabilidad, 154
problema de optimización, 26
proceso de decisión de Markov (MDP), 154
proceso gaussiano (GP), 155
producto de Kronecker, 157
protección de la privacidad, 157
proximable, 158
proyección, 158
pseudoinversa, 158
puerta trasera (backdoor), 159
punto de dato etiquetado, 159
punto de datos, 159
pérdida, 162
pérdida de error cuadrático, 162
pérdida de hinge, 162
pérdida de Huber, 163
pérdida logística, 163
pérdida por error absoluto, 164

realización, 166
recompensa, 166
red de aprendizaje federado (red FL), 167
red neuronal artificial (RNA), 167
reducción de dimensionalidad, 168
referencia (baseline), 169
región de decisión, 171
reglamento general de protección de datos (RGPD), 172
regresión, 172
regresión de Huber, 173
regresión lineal, 173
regresión logística, 173
regresión polinómica, 174

regresión por desviación absoluta mínima, 174
regresión ridge, 174
regularizador, 177
riesgo, 180
riesgo de Bayes, 180
riesgo empírico, 180
robustez, 181
régimen de alta dimensión, 181

selección de modelo, 132
semi-definida positiva (psd), 182
sesgo, 182
suave, 183
subajuste, 184
subgradiente, 185
suposición de agrupamiento, 185
suposición de independencia e idéntica distribución (suposición i.i.d.), 185
supremo (o mínimo de las cotas superiores), 185

tamaño de la muestra, 186
tamaño de paso, 186
tarea de aprendizaje, 186
tasa de aprendizaje, 188
teorema central del límite (CLT), 188
transparency, 190

unidad lineal rectificada (ReLU), 191

validación, 191
validación cruzada de k particiones (k-fold CV), 191
valor atípico (outlier), 192
valor propio (eigenvalue), 193
variable aleatoria (RV), 193
variable aleatoria gaussiana (VA gaussiana), 193
variación total, 195
variación total generalizada (GTV), 195
varianza, 195
vecino más cercano (NN), 195
vecinos, 196
vector, 26
vector de atributos, 198
vector normal estándar, 198
vector propio, 198

weights, 151

árbol de decisión, 199

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] A. Klenke, *Probability Theory: A Comprehensive Course*, 3rd ed. Cham, Switzerland: Springer Nature, 2020.

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

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

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

[9] H. J. Dirschmid, *Tensors and Fields*, (in German). Vienna, Austria: Springer-Verlag, 1996.

[10] G. Strang, *Computational Science and Engineering*. Wellesley, MA, USA: Wellesley-Cambridge Press, 2007.

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

[12] G. B. Folland, *Real Analysis: Modern Techniques and Their Applications*, 2nd ed. New York, NY, USA: Wiley, 1999.

[13] G. Strang, *Introduction to Linear Algebra*, 5th ed. Wellesley-Cambridge Press, MA, 2016.

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

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

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

[17] 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., 2021, pp. 1292–1296, doi: 10.1109/IEEECONF53345.2021.9723162.

[18] 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

[19] 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.

[20] 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.

[21] 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

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

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

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

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

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

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

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

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

[30] 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.

[31] 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.

[32] 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.

[33] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.

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

[35] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Jan. 2003.

[36] 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

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

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

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

[40] 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.

[41] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York, NY, USA: Springer Science+Business Media, 2009.

[42] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001, doi: 10.1214/aos/1013203451.

[43] 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/

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

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

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

[47] 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.

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

[49] 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.

[50] 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.

[51] 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

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

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

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

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

[56] A. Lapidoth, *A Foundation in Digital Communication*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2017.

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

[58] A. Makhdoumi, S. Salamatian, N. Fawaz, and M. Médard, "From the information bottleneck to the privacy funnel," in *2014 IEEE Inf. Theory Workshop*, 2014, pp. 501–505, doi: 10.1109/ITW.2014.6970882.

[59] 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.

[60] J. Zhang, B. Chen, X. Cheng, H. T. T. Binh, and S. Yu, "PoisonGAN: Generative poisoning attacks against federated learning in edge comput-

ing systems," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3310–3322, Mar. 2021, doi: 10.1109/JIOT.2020.3023126.

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

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

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

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

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

[66] R. B. Ash, *Probability and Measure Theory*, 2nd ed. San Diego, CA, USA: Academic, 2000.

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

[68] 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.

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

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

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

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

[73] 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.

[74] 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

[75] 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.

[76] 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.

[77] 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

[78] C. Molnar, *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*, 3rd ed. Ebook, 2025, Accessed: August 1, 2025. [Online]. Available: https://christophm.github.io/interpretable-ml-book/

[79] 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.*, 2017, pp. 618–626, doi: 10.1109/ICCV.2017.74.

[80] 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.

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

[82] 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

[83] 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

[84] 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.

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

[86] 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.

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

[88] 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

[89] 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

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

[91] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," Mar. 2017. [Online]. Available: https://arxiv.org/abs/1702.08608

[92] P. Hase and M. Bansal, "Evaluating explainable AI: Which algorithmic explanations help users predict model behavior?" in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds., Jul. 2020, pp. 5540–5552. [Online]. Available: https://aclanthology.org/2020.acl-main.491

[93] Z. C. Lipton, "The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery," *Queue*, vol. 16, no. 3, pp. 31–57, Jun. 2018, doi: 10.1145/3236386.3241340.

[94] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2015, pp. 1322–1333, doi: 10.1145/2810103.2813677.

[95] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge, UK: Cambridge Univ. Press, 1991.

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

[97] 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.

[98] S. Mallat, "Understanding deep convolutional networks," *Philos. Trans. Roy. Soc. A*, vol. 374, no. 2065, Apr. 2016, Art. no. 20150203, doi: 10.1098/rsta.2015.0203.

[99] 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

[100] H. P. Lopuhaä and P. J. Rousseeuw, "Breakdown points of affine equivariant estimators of multivariate location and covariance matrices," *Ann. Statist.*, vol. 19, no. 1, pp. 229–248, Mar. 1991, doi: 10.1214/aos/1176347978.

[101] R. Durrett, *Probability: Theory and Examples*, 4th ed. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[102] 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.

[103] 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.

[104] E. A. Bender, *An Introduction to Mathematical Modeling.* New York, NY, USA: Wiley, 1978.

[105] 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

[106] J. Heinonen, "Lectures on lipschitz analysis," Dept. Math. Statist., Univ. Jyväskylä, Jyväskylä, Finland, Rep. 100, 2005. [Online]. Available: http://www.math.jyu.fi/research/reports/rep100.pdf

[107] 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.

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

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

[110] 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.

[111] 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

[112] 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

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

[114] 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.

[115] A. Ben-Israel and T. N. E. Greville, *Generalized Inverses: Theory and Applications*, 2nd ed. New York, NY, USA: Springer-Verlag, 2003.

[116] S. Sra, S. Nowozin, and S. J. Wright, Eds., *Optimization for Machine Learning*. MIT Press, 2012.

[117] 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.

[118] 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.

[119] 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.

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

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

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

[123] 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.

[124] 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.

[125] A. van der Vaart, *Asymptotic Statistics*. Cambridge, U.K.: Cambridge Univ. Press, 1998.

[126] 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

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

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

[129] 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.*, 2017, pp. 197–201, doi: 10.1109/IHTC.2017.8058187.

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

[131] 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.

[132] 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.

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

[134] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas *et al.*, "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav)," in *International conference on machine learning*. PMLR, 2018, pp. 2668–2677.