Le Dictionnaire de l'Apprentissage Automatique d'"A"alto

Alexander Jung[1], Konstantina Olioumtsevits[1], Juliette Gronier[2], et Salvatore Rastelli[1]

[1]Aalto University [2]ENS Lyon

July 23, 2025

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

Remerciements

Ce dictionnaire de l'apprentissage automatique a évolué au fil du développement et l'enseignement de plusieurs cours, parmi lesquels 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, and CS-E407507 Human-Centered Machine Learning. Ces cours ont été proposés à Aalto University `https://www.aalto.fi/en`, à des apprenants adultes via le Finnish Institute of Technology (FITech) `https://fitech.io/en/`, et à des étudiants et étudiantes internationaux dans le cadre de l'alliance universitaire européenne Unite! `https://www.aalto.fi/en/unite`.

Nous remercions les étudiants et étudiantes pour leurs retours de qualité qui ont contribué à façonner ce dictionnaire. En particulier, un grand merci à Mikko Seesto pour sa relecture minutieuse.

Cette traduction française s'appuie notamment sur le Glossaire de l'intelligence artificielle (IA) proposé par la CNIL
`https://www.cnil.fr/fr/intelligence-artificielle/glossaire-ia`,
ainsi que sur les ressources du site FranceTerme, géré par le Ministère de la Culture `https://www.culture.fr/franceterme`, *et le site de l'Office québécois de la langue française `https://www.oqlf.gouv.qc.ca`.

Notations et symboles

Ensembles et fonctions

$a \in \mathcal{A}$	L'objet a est un élément de l'ensemble $\mathcal{A}$.
$a := b$	On note a comme abréviation de b.
$\lvert\mathcal{A}\rvert$	Le cardinal (i.e., le nombre d'éléments) d'un ensemble fini $\mathcal{A}$.
$\mathcal{A} \subseteq \mathcal{B}$	$\mathcal{A}$ est un sous-ensemble de $\mathcal{B}$.
$\mathcal{A} \subset \mathcal{B}$	$\mathcal{A}$ est un sous-ensemble strict de $\mathcal{B}$ (i.e., non égal à $\mathcal{B}$).
$\mathbb{N}$	Les entiers naturels $1, 2, \ldots$
$\mathbb{R}$	Les nombres réels x [1].
$\mathbb{R}_+$	Les réels positifs ou nuls $x \geq 0$.
$\mathbb{R}_{++}$	Les réels strictement positifs $x > 0$.
$\{0, 1\}$	L'ensemble composé des deux réels 0 et 1.
$[0, 1]$	L'intervalle fermé des nombres réels x tels que $0 \leq x \leq 1$.

$\operatorname{argmin}_{\mathbf{w}} f(\mathbf{w})$	L'ensemble des points qui minimisent la fonction à valeurs réelles fonction $f(\mathbf{w})$. Voir aussi: fonction.
$\mathbb{S}^{(n)}$	L'ensemble des vecteurs de norme unitaire dans $\mathbb{R}^{n+1}$. Voir aussi: norme, vecteur.
$\exp(a)$	La fonction exponentielle évaluée en un réel $a \in \mathbb{R}$. Voir aussi: fonction.
$\log(a)$	Le logarithme d'un réel strictement positif $a \in \mathbb{R}_{++}$.
$f(\cdot) : \mathcal{A} \to \mathcal{B} : a \mapsto f(a)$	Une fonction (ou application) d'un ensemble $\mathcal{A}$ dans un ensemble $\mathcal{B}$, qui associe à chaque entrée $a \in \mathcal{A}$ une image bien définie $f(a) \in \mathcal{B}$. L'ensemble $\mathcal{A}$ est le domaine de définition de la fonction f et l'ensemble $\mathcal{B}$ est l'ensemble d'arrivée de f. L'apprentissage automatique vise à apprendre une fonction h qui prend en entrée les caractéristiques $\mathbf{x}$ d'un point de données et renvoie une prédiction $h(\mathbf{x})$ pour son étiquette étiquette y. Voir aussi: fonction, application, apprentissage automatique, hypothèse, caractéristique, point de données, prédiction, étiquette.
$\operatorname{epi}(f)$	L' épigraphe d'une fonction à valeurs réelles $f : \mathbb{R}^d \to \mathbb{R}$. Voir aussi: épigraphe, fonction.

$\frac{\partial f(w_1, \ldots, w_d)}{\partial w_j}$	La dérivée partielle (si elle existe) d'une fonction à valeurs réelles $f : \mathbb{R}^d \to \mathbb{R}$ par rapport à w_j [2, Ch. 9]. Voir aussi: fonction.
$\nabla f(\mathbf{w})$	Le gradient d'une fonction à valeurs réelles dérivable $f : \mathbb{R}^d \to \mathbb{R}$ est le vecteur $\nabla f(\mathbf{w}) = \left(\frac{\partial f}{\partial w_1}, \ldots, \frac{\partial f}{\partial w_d}\right)^T \in \mathbb{R}^d$ [2, Ch. 9]. Voir aussi: gradient, dérivable, fonction, vecteur.

Matrices et Vecteurs

$\mathbf{x} = \big(x_1, \ldots, x_d\big)^T$	Un vecteur de taille d, dont la j-ième composante est x_j. Voir aussi: vecteur.
$\mathbb{R}^d$	L'ensemble des vecteurs $\mathbf{x} = \big(x_1, \ldots, x_d\big)^T$ constitués de d composantes réelles $x_1, \ldots, x_d \in \mathbb{R}$. Voir aussi: vecteur.
$\mathbf{I}_{l\times d}$	Une matrice identité généralisée de l lignes et d colonnes. Les composantes de $\mathbf{I}_{l\times d} \in \mathbb{R}^{l\times d}$ valent 1 sur la diagonale principale et 0 ailleurs.
$\mathbf{I}_d, \mathbf{I}$	Une matrice identité carrée de taille $d \times d$. Si la dimension est claire dans le contexte, on peut omettre l'indice.
$\|\mathbf{x}\|_2$	La norme euclidienne (ou ℓ_2) du vecteur $\mathbf{x} = \big(x_1, \ldots, x_d\big)^T \in \mathbb{R}^d$ définie par $\|\mathbf{x}\|_2 := \sqrt{\sum_{j=1}^d x_j^2}$. Voir aussi: norme, vecteur.
$\|\mathbf{x}\|$	Une certaine norme du vecteur $\mathbf{x} \in \mathbb{R}^d$ [3]. Sauf indication contraire, on entend par là la norme euclidienne $\|\mathbf{x}\|_2$. Voir aussi: norme, vecteur
$\mathbf{x}^T$	La transposée d'une matrice ayant pour unique colonne le vecteur $\mathbf{x} \in \mathbb{R}^d$. Voir aussi: vecteur.
$\mathbf{X}^T$	La transposée d'une matrice $\mathbf{X} \in \mathbb{R}^{m\times d}$. Une matrice carrée à valeurs réelles $\mathbf{X} \in \mathbb{R}^{m\times m}$ est dite symétrique si $\mathbf{X} = \mathbf{X}^T$.

$\mathbf{X}^{-1}$	La matrice inverse d'une matrice $\mathbf{X} \in \mathbb{R}^{d \times d}$. Voir aussi: matrice inverse.
$\mathbf{0} = \big(0, \ldots, 0\big)^T$	Le vecteur de $\mathbb{R}^d$ dont toutes les composantes valent 0. Voir aussi: vecteur.
$\mathbf{1} = \big(1, \ldots, 1\big)^T$	Le vecteur de $\mathbb{R}^d$ dont toutes les composantes valent 1. Voir aussi: vecteur.
$\big(\mathbf{v}^T, \mathbf{w}^T\big)^T$	Le vecteur de longueur $d + d'$ obtenu en concaténant les $\mathbf{v} \in \mathbb{R}^d$ avec celles de $\mathbf{w} \in \mathbb{R}^{d'}$. Voir aussi: vecteur.
span$\{\mathbf{B}\}$	Le sous-espace engendré par une matrice $\mathbf{B} \in \mathbb{R}^{a \times b}$, c'est-à-dire l'ensemble de toutes les combinaisons linéaires des colonnes de $\mathbf{B}$: span$\{\mathbf{B}\} = \big\{\mathbf{Ba} : \mathbf{a} \in \mathbb{R}^b\big\} \subseteq \mathbb{R}^a$.
null$\{\mathbf{A}\}$	Le noyau d'une matrice $\mathbf{A} \in \mathbb{R}^{a \times b}$, qui est le sous-espace des vecteurs $\mathbf{a} \in \mathbb{R}^b$ tels que $\mathbf{Aa} = \mathbf{0}$. Voir aussi : matrice.
$\det(\mathbf{C})$	Le déterminant de la matrice $\mathbf{C}$. Voir aussi: déterminant
$\mathbf{A} \otimes \mathbf{B}$	Le produit de Kronecker des matrices $\mathbf{A}$ et $\mathbf{B}$ [4]. Voir aussi: produit de Kronecker

Théorie des probabilités

$\mathbb{P}(\mathcal{A})$	La probabilité de l'événement mesurable $\mathcal{A}$. Voir aussi: probabilité.
$\mathbf{x} \sim p(\mathbf{z})$	La variable aléatoire (VA) $\mathbf{x}$ suit la loi de probabilité $p(\mathbf{z})$ [5, 6]. Voir aussi : VA, loi de probabilité
$\mathbb{E}_p\{f(\mathbf{z})\}$	L'espérance d'une VA $f(\mathbf{z})$ obtenue en appliquant une fonction déterministe f à une VA $\mathbf{z}$ dont la loi de probabilité est $\mathbb{P}(\mathbf{z})$. Si la loi de probabilité est claire dans le contexte, on écrit simplement $\mathbb{E}\{f(\mathbf{z})\}$. Voir aussi : espérance, VA, fonction, loi de probabilité
$\mathrm{cov}\,(x, y)$	La covariance entre deux VA à valeurs réelles définies sur un même espace probabilisé. Voir aussi : covariance, VA, espace probabilisé
$\mathbb{P}(\mathbf{x}, y)$	Une loi de probabilité (conjointe) d'une VA dont les réalisations sont des points de données avec des caractéristiques $\mathbf{x}$ et une étiquette y. Voir aussi : loi de probabilité, VA, réalisation, point de données, caractéristique, étiquette.
$\mathbb{P}(\mathbf{x}\|y)$	Une loi de probabilité conditionnelle d'une VA $\mathbf{x}$ étant donnée la valeur d'une autre VA y [7, Sec. 3.5]. Voir aussi : loi de probabilité, VA.

$\mathbb{P}(\mathbf{x}; \mathbf{w})$	Une loi de probabilité paramétrée d'une VA $\mathbf{x}$. La loi de probabilité dépend d'un vecteur de paramètres $\mathbf{w}$. Par exemple, $\mathbb{P}(\mathbf{x}; \mathbf{w})$ pourrait être une loi normale multivariée avec un vecteur de paramètres $\mathbf{w}$ donné par les composantes du vecteur de moyenne $\mathbb{E}\{\mathbf{x}\}$ et la matrice de covariance $\mathbb{E}\Big\{\big(\mathbf{x} - \mathbb{E}\{\mathbf{x}\}\big)\big(\mathbf{x} - \mathbb{E}\{\mathbf{x}\}\big)^T\Big\}$. Voir aussi : loi de probabilité, VA, paramètre, loi normale multivariée, moyenne, matrice de covariance, vecteur.
$\mathcal{N}(\mu, \sigma^2)$	La loi de probabilité d'une variable aléatoire normale (VA normale) $x \in \mathbb{R}$ ayant comme moyenne (ou espérance) $\mu = \mathbb{E}\{x\}$ et comme variance $\sigma^2 = \mathbb{E}\big\{(x - \mu)^2\big\}$. Voir aussi : VA normale, moyenne, espérance, variance, loi de probabilité
$\mathcal{N}(\boldsymbol{\mu}, \mathbf{C})$	La loi normale multivariée d'une VA normale vectorielle $\mathbf{x} \in \mathbb{R}^d$ ayant comme moyenne (ou espérance) $\boldsymbol{\mu} = \mathbb{E}\{\mathbf{x}\}$ et comme matrice de covariance $\mathbf{C} = \mathbb{E}\big\{\big(\mathbf{x} - \boldsymbol{\mu}\big)\big(\mathbf{x} - \boldsymbol{\mu}\big)^T\big\}$. Voir aussi : loi normale multivariée, VA normale, moyenne, matrice de covariance.
Ω	Un espace échantillon, c'est-à-dire l'ensemble de tous les résultats possibles d'une expérience aléatoire. Voir aussi : événement.

$\mathcal{F}$	Une collection de sous-ensembles mesurables d'un espace échantillon Ω. Voir aussi : espace échantillon, événement.
$\mathcal{P}$	Un espace probabilisé constitué d'un espace échantillon Ω, d'une tribu $\mathcal{F}$ de sous-ensembles mesurables de Ω, et d'une loi de probabilité $\mathbb{P}(\cdot)$. Voir aussi : loi de probabilité, espace échantillon, mesurable.

Apprentissage automatique

r	Un indice $r = 1, 2, \ldots$ qui énumère les points de données. Voir aussi : points de données.
m	Le nombre de points de données dans un jeu de données (c'est-à-dire la taille du jeu de données). Voir aussi : points de données, jeu de données.
$\mathcal{D}$	Un jeu de données $\mathcal{D} = \{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}\}$ est une liste de points de données individuels $\mathbf{z}^{(r)}$, pour $r = 1, \ldots, m$. Voir aussi : points de données, jeu de données.
d	Le nombre de caractéristiques qui constituent un point de données. Voir aussi : caractéristiques, point de données.
x_j	La j-ième caractéristique d'un point de données. La première caractéristique est notée x_1, la deuxième x_2, et ainsi de suite. Voir aussi : caractéristiques, point de données.
$\mathbf{x}$	Le vecteur de caractéristiques $\mathbf{x} = \big(x_1, \ldots, x_d\big)^T$ d'un point de données, dont les composantes sont les différentes caractéristiques du point de données. Voir aussi : vecteur de caractéristiques, caractéristiques, point de données.

$\mathcal{X}$	L'espace des caractéristiques $\mathcal{X}$ est l'ensemble de toutes les valeurs possibles que les caractéristiques $\mathbf{x}$ d'un point de données peuvent prendre. Voir aussi : espace des caractéristiques, caractéristiques, point de données.
$\mathbf{z}$	Au lieu du symbole $\mathbf{x}$, on utilise parfois $\mathbf{z}$ comme un autre symbole pour désigner un vecteur dont les composantes sont les différentes caractéristiques d'un point de données. On a besoin de deux symboles différents pour distinguer les caractéristiques brutes des caractéristiques apprises [8, Ch. 9]. Voir aussi : caractéristiques, point de données, vecteur.
$\mathbf{x}^{(r)}$	Le vecteur de caractéristiques du r-ième point de données dans un jeu de données. Voir aussi : vecteur de caractéristiques, point de données, jeu de données.
$x_j^{(r)}$	La j-ième caractéristique du r-ième point de données dans un jeu de données. Voir aussi : caractéristiques, point de données, jeu de données.
y	L'étiquette (ou quantité d'intérêt) d'un point de données. Voir aussi : étiquette, point de données.
$y^{(r)}$	L'étiquette du r-ième point de données. Voir aussi : étiquette, point de données.

$\left(\mathbf{x}^{(r)}, y^{(r)}\right)$	Les caractéristiques et l'étiquette du r-ième point de données. Voir aussi : caractéristiques, étiquette, point de données.
$\mathcal{Y}$	L'espace des étiquettes $\mathcal{Y}$ d'une méthode d'apprentissage automatique comprend toutes les valeurs d'étiquette qu'un point de données peut porter. L'espace des étiquettes nominal peut être plus grand que l'ensemble des différentes valeurs d'étiquette présentes dans un jeu de données donné (par exemple, un ensemble d'entraînement (ou d'apprentissage)). Les problèmes (ou méthodes) d'apprentissage automatique utilisant un espace des étiquettes numérique, comme $\mathcal{Y} = \mathbb{R}$ ou $\mathcal{Y} = \mathbb{R}^3$, sont appelés problèmes (ou méthodes) de régression. Les problèmes (ou méthodes) d'apprentissage automatique utilisant un espace des étiquettes discret, comme $\mathcal{Y} = \{0, 1\}$ ou $\mathcal{Y} = \{chat, chien, souris\}$, sont appelés problèmes (ou méthodes) de classification. Voir aussi : espace des étiquettes, apprentissage automatique, étiquette, point de données, jeu de données, ensemble d'entraînement, régression, classification.
$\mathcal{B}$	Un mini-lot (ou sous-ensemble) de points de données choisis aléatoirement. Voir aussi : lot, points de données.

B	La taille (c'est-à-dire le nombre de points de données) d'un mini-lot. Voir aussi : lot, points de données.
$h(\cdot)$	Une fonction hypothèse qui lit les caractéristiques $\mathbf{x}$ d'un point de données et produit une prédiction $\hat{y} = h(\mathbf{x})$ pour son étiquette y. Voir aussi : hypothèse, application, caractéristique, point de données, prédiction, étiquette.
$\mathcal{Y}^{\mathcal{X}}$	Étant donnés deux ensembles $\mathcal{X}$ et $\mathcal{Y}$, on note $\mathcal{Y}^{\mathcal{X}}$ l'ensemble de toutes les fonctions hypothèses possibles $h : \mathcal{X} \to \mathcal{Y}$. Voir aussi : hypothèse, application.
$\mathcal{H}$	Un espace des hypothèses ou modèle utilisé par une méthode d'apprentissage automatique. L'espace des hypothèses est constitué des différentes hypothèses $h : \mathcal{X} \to \mathcal{Y}$, parmi lesquelles la méthode d'apprentissage automatique doit choisir. Voir aussi : espace des hypothèses, modèle, apprentissage automatique, hypothèse, application.
$d_{\text{eff}}(\mathcal{H})$	La dimension effective d'un espace des hypothèses $\mathcal{H}$. Voir aussi : dimension effective, espace des hypothèses.

B^2	Le biais au carré d'une hypothèse apprise $\hat{h}$, ou de ses paramètres. Notons que $\hat{h}$ devient une VA lorsqu'elle est apprise à partir de points de données eux-mêmes considérés comme des VA. Voir aussi : biais, hypothèse, paramètre, VA, point de données.
V	La variance d'une hypothèse apprise $\hat{h}$, ou de ses paramètres. Notons que $\hat{h}$ devient une VA lorsqu'elle est apprise à partir de points de données eux-mêmes considérés comme des VA. Voir aussi : variance, hypothèse, paramètre, VA, point de données.
$L\left((\mathbf{x}, y), h\right)$	La perte encourue en prédisant l'étiquette y d'un point de données à l'aide de la prédiction $\hat{y} = h(\mathbf{x})$. La prédiction $\hat{y}$ est obtenue en évaluant la fonction hypothèse $h \in \mathcal{H}$ en $\mathbf{x}$, le vecteur de caractéristiques du point de données. Voir aussi : perte, étiquette, prédiction, hypothèse, vecteur de caractéristiques, point de données.

E_v	L'erreur de validation d'une hypothèse h, c'est-à-dire sa perte moyenne sur un ensemble de validation. Voir aussi : erreur de validation, perte, hypothèse, ensemble de validation.
$\widehat{L}\big(h\|\mathcal{D}\big)$	Le risque empirique, ou perte moyenne, encouru par l'hypothèse h sur un jeu de données $\mathcal{D}$. Voir aussi : risque empirique, perte, hypothèse, jeu de données.
E_t	L'erreur d'entrainement d'une hypothèse h, c'est-à-dire sa perte moyenne sur un ensemble d'entraînement. Voir aussi : erreur d'entrainement, perte, hypothèse, ensemble d'entraînement.
t	Un indice de temps discret $t = 0, 1, \ldots$ utilisé pour énumérer des événements séquentiels (ou des instants temporels).
α	Un paramètre de régularisation qui contrôle la quantité de régularisation. Voir aussi : paramètre, régularisation.
t	Un indice qui énumère les tâches d'apprentissage dans un problème d'apprentissage multitâche. Voir aussi : tâches d'apprentissage, apprentissage multitâche.

η	Le taux d'apprentissage (ou taille de pas) utilisé par les méthodes basées sur le gradient. Voir aussi : taux d'apprentissage, taille de pas, méthodes basées sur le gradient
$\lambda_j(\mathbf{Q})$	La j-ième valeur propre (triée par ordre croissant ou décroissant) d'une matrice semi-définie positive $\mathbf{Q}$. Si la matrice est claire dans le contexte, on écrit simplement λ_j. Voir aussi : valeur propre, semi-définie positive
$\sigma(\cdot)$	La fonction d'activation utilisée par un neurone artificiel dans un réseau de neurones artificiels (RNA). Voir aussi : fonction d'activation, RNA
$\mathcal{R}_{\hat{y}}$	Une région de décision dans un espace des caractéristiques. Voir aussi : région de décision, espace des caractéristiques
$\mathbf{w}$	Un vecteur de paramètres $\mathbf{w} = \big(w_1, \ldots, w_d\big)^T$ d'un modèle, par exemple les poids d'un modèle linéaire ou dans un RNA. Voir aussi : poids, modèle, modèle linéaire, RNA, vecteur

$h^{(\mathbf{w})}(\cdot)$	Une fonction hypothèse qui dépend de paramètres du modèle $w_1, \ldots, w_d$ regroupés dans le vecteur $\mathbf{w} = \big(w_1, \ldots, w_d\big)^T$ et qui peuvent être ajustés. Voir aussi : hypothèse, paramètres du modèle, application, vecteur
$\phi(\cdot)$	Une transformation de caractéristiques $\phi : \mathcal{X} \to \mathcal{X}' : \mathbf{x} \mapsto \mathbf{x}' := \phi\big(\mathbf{x}\big) \in \mathcal{X}'$. Voir aussi : transformation de caractéristiques, espace des caractéristiques
$K\big(\cdot, \cdot\big)$	Étant donné un espace des caractéristiques $\mathcal{X}$, un noyau est une application $K : \mathcal{X} \times \mathcal{X} \to \mathbb{C}$ qui est semi-définie positive. Voir aussi : noyau, espace des caractéristiques, semi-définie positive, application

Apprentissage fédéré

$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	Un graphe non orienté dont les sommets $i \in \mathcal{V}$ représentent des appareils au sein d'un réseau d'apprentissage fédéré. Les arêtes pondérées non orientées $\mathcal{E}$ représentent la connectivité entre les appareils et les similarités statistiques entre leurs jeux de données et tâches d'apprentissage. Voir aussi : graphe, appareil, réseau d'apprentissage fédéré, jeu de données, tâche d'apprentissage
$i \in \mathcal{V}$	Un sommet représentant un appareil dans un réseau d'apprentissage fédéré. L'appareil peut accéder à un jeu de données local et entraîner un modèle local. Voir aussi : appareil, réseau d'apprentissage fédéré, jeu de données local, modèle local
$\mathcal{G}^{(\mathcal{C})}$	Le sous-graphe induit de $\mathcal{G}$ utilisant les sommets de $\mathcal{C} \subseteq \mathcal{V}$. Voir aussi : graphe
$\mathbf{L}^{(\mathcal{G})}$	La matrice laplacienne d'un graphe $\mathcal{G}$. Voir aussi : matrice laplacienne, graphe
$\mathbf{L}^{(\mathcal{C})}$	La matrice laplacienne du graphe induit $\mathcal{G}^{(\mathcal{C})}$. Voir aussi : matrice laplacienne, graphe
$\mathcal{N}^{(i)}$	Le voisinage du sommet i dans un graphe $\mathcal{G}$. Voir aussi : voisinage, graphe

$d^{(i)}$	Le degré pondéré $d^{(i)} := \sum_{i' \in \mathcal{N}^{(i)}} A_{i,i'}$ d'un sommet i dans un graphe $\mathcal{G}$. Voir aussi : graphe.
$d_{\max}^{(\mathcal{G})}$	Le degré pondéré maximal d'un graphe $\mathcal{G}$. Voir aussi : maximum, degré, graphe.
$\mathcal{D}^{(i)}$	Le jeu de données local $\mathcal{D}^{(i)}$ détenu par le sommet $i \in \mathcal{V}$ d'un réseau d'apprentissage fédéré. Voir aussi : jeu de données local, réseau d'apprentissage fédéré
m_i	Le nombre de points de données (i.e., la taille d'échantillon) contenus dans le jeu de données local $\mathcal{D}^{(i)}$ au sommet $i \in \mathcal{V}$. Voir aussi : point de données, taille d'échantillon, jeu de données local
$\mathbf{x}^{(i,r)}$	Les caractéristiques du r-ième point de données dans le jeu de données local $\mathcal{D}^{(i)}$. Voir aussi : caractéristique, point de données, jeu de données local
$y^{(i,r)}$	L'étiquette du r-ième point de données dans le jeu de données local $\mathcal{D}^{(i)}$. Voir aussi : étiquette, point de données, jeu de données local

$\mathbf{w}^{(i)}$	Les paramètres du modèle locaux de l'appareil i au sein d'un réseau d'apprentissage fédéré. Voir aussi : paramètres du modèle, appareil, réseau d'apprentissage fédéré
$L_i\left(\mathbf{w}\right)$	La fonction de perte (ou de coût) locale utilisée par l'appareil i pour évaluer l'utilité d'un certain choix $\mathbf{w}$ pour les paramètres du modèle locaux. Voir aussi : fonction de perte, appareil, paramètres du modèle
$L^{(\mathrm{d})}\left(\mathbf{x}, h\big(\mathbf{x}\big), h'\big(\mathbf{x}\big)\right)$	La perte encourue par une hypothèse h' sur un point de données de caractéristiques $\mathbf{x}$ et d'étiquette $h\big(\mathbf{x}\big)$ obtenue à partir d'une autre hypothèse. Voir aussi : perte, hypothèse, point de données, caractéristique, étiquette
$\mathrm{stack}\big\{\mathbf{w}^{(i)}\big\}_{i=1}^{n}$	Le vecteur $\left(\big(\mathbf{w}^{(1)}\big)^T, \ldots, \big(\mathbf{w}^{(n)}\big)^T\right)^T \in \mathbb{R}^{dn}$ obtenu en empilant verticalement les paramètres du modèle locaux $\mathbf{w}^{(i)} \in \mathbb{R}^d$. Voir aussi : paramètres du modèle, vecteur.

Concepts de l'apprentissage automatique

algorithme Un algorithme est une spécification précise, étape par étape, qui explique comment produire une sortie à partir d'une entrée donnée en un nombre fini d'étapes de calcul [9]. Par exemple, un algorithme pour entraîner un modèle linéaire décrit explicitement comment transformer un ensemble d'entraînement donné en paramètres du modèle via une séquence de pas de gradient. Pour étudier rigoureusement les algorithmes, on peut les représenter (ou les approximer) par différentes structures mathématiques [10]. Une approche consiste à représenter un algorithme comme un ensemble d'exécutions possibles. Chaque exécution individuelle est alors une séquence de la forme

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

Cette séquence commence par une entrée et progresse par des étapes intermédiaires jusqu'à la délivrance d'une sortie. IL est crucial de retenir qu'un algorithme englobe plus qu'une simple fonction de l'entrée vers la sortie ; il inclut aussi les étapes intermédiaires de calcul $s_1, \ldots, s_T$. Voir aussi: modèle linéaire, ensemble d'entraînement, paramètres du modèle, pas, modèle, stochastique.

algorithme incrémental (ou en ligne) Un algorithme incrémental traite les données d'entrée de manière progressive, recevant les points de données de façon séquentielle et prenant des décisions ou produisant des

sorties immédiatement sans avoir accès à l'ensemble des données en avance [11], [12]. Contrairement à un algorithme hors ligne, qui dispose de toutes les données dès le départ, un algorithme incrémental doit gérer l'incertitude liée aux entrées futures et ne peut pas modifier les décisions passées. De manière similaire à un algorithme hors ligne, on représente formellement un algorithme incrémental comme un ensemble d'exécutions possibles. Cependant, la séquence d'exécution d'un algorithme incrémental présente une structure spécifique:

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

Chaque exécution commence par un état initial (c'est-à-dire in_1) et se poursuit par une alternance d'étapes de calcul, de sorties (ou décisions), puis d'entrées. Plus précisément, à l'étape k, l'algorithme effectue une étape de calcul s_k, génère une sortie out_k, puis reçoit l'entrée suivante (le point de données) in_{k+1}. Un exemple notable d'algorithme incrémental en apprentissage automatique est la descente de gradient en ligne, qui met à jour les paramètres du modèle de façon progressive à mesure que de nouveaux points de données arrivent.

Voir aussi: apprentissage incrémental, descente de gradient en ligne, algorithme.

algorithme stochastique Un algorithme stochastique utilise un mécanisme aléatoire lors de son exécution. Par exemple, la descente de gradient stochastique (SGD) utilise un sous-ensemble choisi aléatoirement de points de données pour approximer le gradient d'une fonction objective. On peut représenter un algorithme stochastique par un processus

stochastique, c'est-à-dire que la séquence d'exécution possible correspond aux issues possibles d'une expérience aléatoire [7], [13], [14].

Voir aussi : stochastique, algorithme, SGD, point de données, gradient, fonction objective, processus stochastique, méthode d'optimisation, méthodes basées sur le gradient.

analyse en composantes principales (PCA) L'analyse en composantes principales (PCA) détermine une transformation de caractéristiques linéaire telle que les nouvelles caractéristiques permettent de reconstruire les caractéristiques d'origine avec une erreur de reconstruction minimale [8].

Voir aussi : transformation de caractéristiques, caractéristique, minimum.

appareil Tout système physique qui peut être utilisé pour stocker et traiter des données. Dans le contexte de l'apprentissage automatique, on entend généralement un ordinateur capable de lire des points de données provenant de différentes sources et, en retour, d'entraîner un modèle d'apprentissage automatique en utilisant ces points de données.

Voir aussi : données, apprentissage automatique, point de données, modèle.

application On utilise le terme application comme synonyme pour fonction.

Voir aussi: fonction.

application linéaire Une application linéaire $f : \mathbb{R}^n \to \mathbb{R}^m$ est une fonction qui satisfait l'additivité, c'est-à-dire, $f(\mathbf{x} + \mathbf{y}) = f(\mathbf{x}) + f(\mathbf{y})$, et l'homogénéité, c'est-à-dire, $f(c\mathbf{x}) = cf(\mathbf{x})$, pour tous les vecteurs

$\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ et les scalaires $c \in \mathbb{R}$. En particulier, $f(\mathbf{0}) = \mathbf{0}$. Toute application linéaire peut être représentée comme une multiplication matricielle $f(\mathbf{x}) = \mathbf{A}\mathbf{x}$ pour une certaine matrice $\mathbf{A} \in \mathbb{R}^{m \times n}$. La famille des applications linéaires à valeurs réelles pour une dimension donnée n constitue un modèle linéaire qui est utilisé dans de nombreuses méthodes d'apprentissage automatique.

Voir aussi: application, fonction, modèle linéaire, apprentissage automatique.

apprentissage automatique (ou apprentissage machine) L' apprentissage automatique vise à prédire une étiquette à partir des caractéristiques d'un point de données. Les méthodes d'apprentissage automatique réalisent cela en apprenant une hypothèse (ou modèle) issue d'un espace des hypothèses par la minimisation d'une fonction de perte [8,15]. Une formulation précise de ce principe est donnée par le minimisation du risque empirique (MRE). Les différentes méthodes d'apprentissage automatique sont obtenues par divers choix pour les points de données (leurs caractéristiques et leur étiquette), le modèle et la fonction de perte [8, Ch. 3].

Voir aussi: étiquette, caractéristique, point de données, hypothèse, espace des hypothèses, modèle, fonction de perte, MRE.

apprentissage de caractéristiques Considérons une application d'apprentissage automatique avec des points de données caractérisés par des caractéristiques brutes $\mathbf{x} \in \mathcal{X}$. L'apprentissage de caractéristiques désigne la

tâche consistant à apprendre une application

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

qui lit les caractéristiques $\mathbf{x} \in \mathcal{X}$ d'un point de données et produit de nouvelles caractéristiques $\mathbf{x}' \in \mathcal{X}'$ appartenant à un nouvel espace des caractéristiques $\mathcal{X}'$. Différentes méthodes d'apprentissage de caractéristiques résultent de différents choix de conception pour $\mathcal{X}, \mathcal{X}'$, pour un espace des hypothèses $\mathcal{H}$ d'applications possibles $\mathbf{\Phi}$, et pour une mesure quantitative de l'utilité d'un $\mathbf{\Phi} \in \mathcal{H}$ donné. Par exemple, analyse en composantes principales (PCA) utilise $\mathcal{X} := \mathbb{R}^d$, $\mathcal{X}' := \mathbb{R}^{d'}$ avec $d' < d$, et un espace des hypothèses

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

PCA mesure l'utilité d'une application $\mathbf{\Phi}(\mathbf{x}) = \mathbf{F}\mathbf{x}$ par l'erreur de reconstruction linéaire minimale sur un jeu de données, à savoir :

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

Voir aussi : apprentissage automatique, point de données, caractéristique, application, espace des caractéristiques, espace des hypothèses, PCA, minimum, jeu de données.

apprentissage fédéré L'apprentissage fédéré est un terme générique désignant les méthodes d'apprentissage automatique qui entraînent des modèles de manière collaborative à l'aide de données et de calculs décentralisés.

Voir aussi : apprentissage automatique, modèle, données.

apprentissage incrémental (ou en ligne) Certaines méthodes d'apprentissage automatique sont conçues pour traiter les points de données de manière séquentielle, en mettant à jour les paramètres du modèle au fur et à mesure que de nouveaux points de données deviennent disponibles (un à la fois). Un exemple typique est celui des séries temporelles, comme les températures minimales et maximales journalières enregistrées par une station météorologique du Institut météorologique finlandais (FMI). Ces valeurs forment une séquence chronologique d'observations. À chaque instant t, les méthodes d'apprentissage incrémental mettent à jour (ou affinent) l'hypothèse actuelle $h^{(t)}$ (ou les paramètres du modèle $\mathbf{w}^{(t)}$) à partir du nouveau point de données observé $\mathbf{z}^{(t)}$.

Voir aussi: descente de gradient en ligne, algorithme incrémental.

apprentissage multitâche L'apprentissage multitâche vise à exploiter les relations entre différentes tâches d'apprentissage. Considérons deux tâches d'apprentissage obtenues à partir du même jeu de données d'images de webcam. La première tâche consiste à prédire la présence d'un humain, tandis que la seconde tâche consiste à prédire la présence d'une voiture. Il peut être utile d'utiliser la même structure de réseau de neurones profond pour les deux tâches et de ne permettre qu'aux poids de la couche de sortie finale d'être différents.

Voir aussi: tâche d'apprentissage, jeu de données, réseau de neurones profond, poids.

apprentissage par renforcement L'apprentissage par renforcement désigne un cadre d'apprentissage incrémental dans lequel on ne peut évaluer

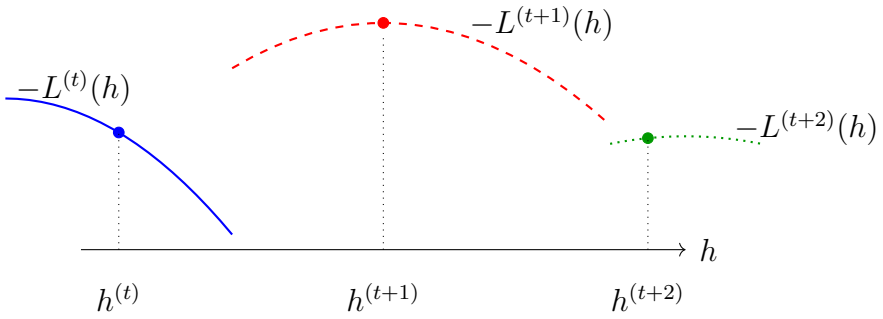


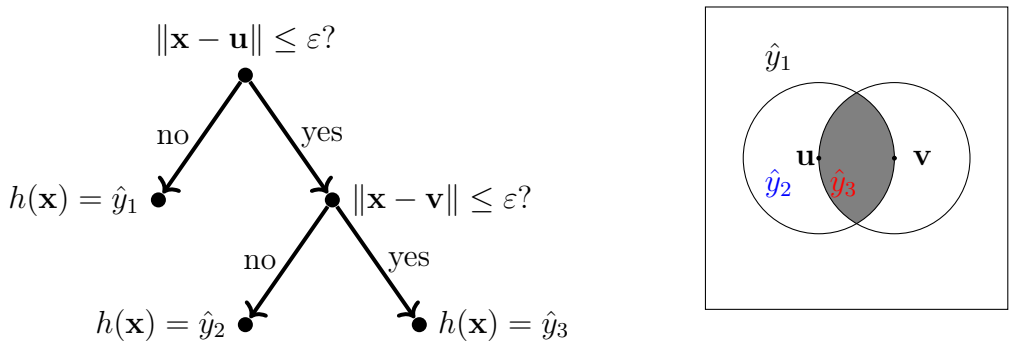
Fig. 1. Trois pas de temps consécutifs $t, t+1, t+2$ avec les fonctions de perte correspondantes $L^{(t)}, L^{(t+1)}, L^{(t+2)}$. Au pas de temps t, une méthode d'apprentissage par renforcement peut évaluer la fonction de perte uniquement pour une hypothèse spécifique $h^{(t)}$, ce qui donne le signal de récompense $r^{(t)} = -L^{(t)}(h^{(t)})$.

l'utilité que d'une seule hypothèse (c'est-à-dire un choix de modèle et de paramètres) à chaque instant t. En particulier, les méthodes d'apprentissage par renforcement appliquent l'hypothèse actuelle $h^{(t)}$ au vecteur de caractéristiques $\mathbf{x}^{(t)}$ du point de données nouvellement reçu. L'utilité de la prédiction obtenue $h^{(t)}(\mathbf{x}^{(t)})$ est quantifiée par un signal de récompense $r^{(t)}$. En général, la récompense dépend également des prédictions précédentes $h^{(t')}\big(\mathbf{x}^{(t')}\big)$ pour $t' < t$. L'objectif de l'apprentissage par renforcement est d'apprendre $h^{(t)}$ à chaque instant t de façon à maximiser la récompense cumulée (éventuellement avec un facteur d'actualisation) [8], [16].

Voir aussi : fonction de perte, récompense, apprentissage automatique.

arbre de décision Un arbre de décision est une représentation en forme

d'organigramme d'une fonction hypothèse h. Plus formellement, un arbre de décision est un graphe orienté composé d'un sommet en racine qui lit le vecteur de caractéristiques $\mathbf{x}$ d'un point de données. La racine transfère ensuite ce point de données à l'un de ses sommets enfants en fonction d'un test élémentaire sur les caractéristiques de $\mathbf{x}$. Si le sommet récepteur n'est pas une feuille (c'est-à-dire qu'il a lui-même des enfants), il représente un nouveau test. Selon le résultat de ce test, le point de données est à nouveau transféré vers l'un des sommets descendants. Ce processus de test et de transfert est répété jusqu'à ce que le point de données atteigne une feuille (un sommet sans enfant).



(a) (b)

Fig. 2. (a) Un arbre de décision est une représentation en organigramme d'une hypothèse $h : \mathcal{X} \to \mathbb{R}$ constante par morceaux. CHaque morceau correspond à une région de décision $\mathcal{R}_{\hat{y}} := \{\mathbf{x} \in \mathcal{X} : h(\mathbf{x}) = \hat{y}\}$. L'arbre de décision illustré s'applique à des vecteurs de caractéristiques, i.e., $\mathcal{X} \subseteq \mathbb{R}^d$. Il est paramétré par un seuil $\varepsilon > 0$ et des vecteurs $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$. (b) Un arbre de décision partitionne l'espace des caractéristiques $\mathcal{X}$ en régions de décision. Chaque région de décision $\mathcal{R}_{\hat{y}} \subseteq \mathcal{X}$ correspond à une feuille particulière de l'arbre.

Voir aussi: région de décision.

aspects computationnels Par aspects computationnels (ou calculatoires) d'une méthode d'apprentissage automatique, on entend principalement les ressources computationnelles nécessaires à sa mise en œuvre. Par exemple, si une méthode d'apprentissage automatique utilise des techniques d'optimisation itérative pour résoudre une MRE, alors ses aspects computationnels incluent: 1) combien d'opérations arithmétiques sont nécessaires pour exécuter une itération unique (pas de gradient) ; et

2) combien d'itérations sont nécessaires pour obtenir des paramètres du modèle utiles. Un exemple important de technique d'optimisation itérative est la descente de gradient.

Voir aussi: apprentissage automatique, MRE, pas, paramètres du modèle, descente de gradient.

aspects statistiques Par aspects statistiques d'une méthode d'apprentissage automatique, on entend (les propriétés de) la loi de probabilité de sa sortie sous un modèle probabiliste pour les données fournies en entrée de la méthode.

Voir aussi: apprentissage automatique, loi de probabilité, modèle probabiliste, données.

attaque Une attaque contre un système d'apprentissage automatique désigne une action intentionnelle — active ou passive — visant à compromettre l'intégrité, la disponibilité ou la confidentialité du système. Les attaques actives consistent à perturber certains composants, tels que les jeux de données (par exemple via des attaques par empoisonnement de données) ou les liens de communication entre les appareils dans une application d'apprentissage automatique. Les attaques passives, comme les atteintes à la vie privée, cherchent à déduire des données sensibles sans modifier le système. Selon leur objectif, on distingue les attaque par déni de services, par porte dérobée, ainsi que les atteintes à la vie privée.

Voir aussi : empoisonnement de données, atteinte à la vie privée, donnée sensible, attaque par déni de service, porte dérobée.

attaque par déni de service Une attaque par déni de service vise (par

exemple via une attaque par empoisonnement de données) à orienter l'entraînement d'un modèle de manière à ce qu'il présente de mauvaises performances sur des points de données typiques.

Voir aussi : attaque, empoisonnement de données, modèle, point de données.

atteinte à la vie privée Une atteinte à la vie privée d'un système d'apprentissage automatique vise à déduire des données sensibles d'individus en exploitant un accès partiel à un modèle d'apprentissage automatique entraîné. Une forme d'atteinte à la vie privée est l'inversion de modèle.

Voir aussi : attaque, donnée sensible, inversion de modèle, intelligence artificielle digne de confiance (IA digne de confiance), Règlement général sur la protection des données (RGPD).

augmentation de données Les méthodes d'augmentation de données ajoutent des points de données synthétiques à un ensemble existant de points de données. Ces points de données synthétiques sont obtenus par perturbation (par exemple, ajout de bruit aux mesures physiques) ou transformation (par exemple, rotations d'images) des points de données originaux. Ces perturbations et transformations sont telles que les points de données synthétiques résultants doivent toujours avoir la même étiquette. À titre d'exemple, une image de chat tournée est toujours une image de chat même si leurs vecteurs de caractéristiques (obtenus en empilant les intensités des pixels) sont très différents (voir Figure 3). L'augmentation de données peut être une forme efficace de régularisation.

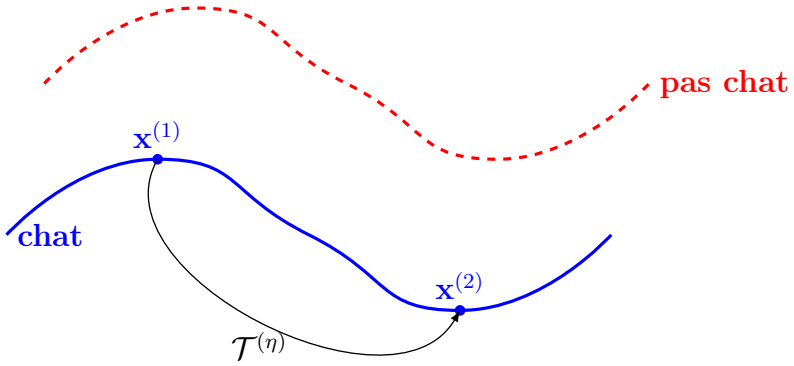


Fig. 3. L'augmentation de données exploite les symétries intrinsèques des points de données dans un certain espace des caractéristiques $\mathcal{X}$. On peut représenter une symétrie par un opérateur $\mathcal{T}^{(\eta)} : \mathcal{X} \to \mathcal{X}$, paramétré par un nombre $\eta \in \mathbb{R}$. Par exemple, $\mathcal{T}^{(\eta)}$ pourrait représenter l'effet de la rotation d'une image de chat de η degrés. Un point de données avec comme vecteur de caractéristiques $\mathbf{x}^{(2)} = \mathcal{T}^{(\eta)}\left(\mathbf{x}^{(1)}\right)$ doit avoir la même étiquette $y^{(2)} = y^{(1)}$ qu'un point de données avec comme vecteur de caractéristiques $\mathbf{x}^{(1)}$.

Voir aussi: données, point de données, étiquette, vecteur de caractéristiques, régularisation, espace des caractéristiques.

bandit manchot Un problème de bandit manchot est une formulation mathématique précise d'une tâche de prise de décision séquentielle sous incertitude. À chaque pas de temps discret k, un apprenant sélectionne l'une des plusieurs actions possibles—appelées bras—à partir d'un ensemble fini $\mathcal{A}$. Tirer le bras a au temps k donne lieu à une récompense $r^{(a,k)}$ tirée selon une loi de probabilité inconnue $\mathbb{P}(r^{(a,k)})$. On obtient différentes classes de problèmes du bandit manchot selon les restrictions

imposées à cette loi de probabilité. Dans le cas le plus simple, la loi de probabilité $\mathbb{P}(r^{(a,k)})$ ne dépend pas de t. Étant donné un problème de bandit manchot, l'objectif est de concevoir des méthodes d'apprentissage automatique qui maximisent la récompense cumulée dans le temps en équilibrant stratégiquement exploration (c.-à-d. collecte d'informations sur les bras incertains) et exploitation (c.-à-d. sélection des bras connus pour bien performer). Les problèmes de de bandit manchot constituent un cas particulier important des problèmes d'apprentissage par renforcement [16, 17].

Voir aussi: récompense, regret.

biais Considérons une méthode d'apprentissage automatique utilisant un espace des hypothèses paramétré $\mathcal{H}$. Celle-ci apprend les paramètres du modèle $\mathbf{w} \in \mathbb{R}^d$ à partir du jeu de données

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

Pour analyser les propriétés de la méthode d'apprentissage automatique, on interprète généralement les points de données comme des réalisations de VA indépendantes et identiquement distribuées (i.i.d.)),

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

On peut alors considérer la méthode d'apprentissage automatique comme un estimateur $\widehat{\mathbf{w}}$ calculé à partir de $\mathcal{D}$ (par exemple, en résolvant une MRE). Le biais (au carré) de l'estimateur $\widehat{\mathbf{w}}$ se définit alors comme $B^2 := \left\|\mathbb{E}\{\widehat{\mathbf{w}}\} - \overline{\mathbf{w}}\right\|_2^2$.

Voir aussi: apprentissage automatique, espace des hypothèses, paramètres

du modèle, jeu de données, point de données, réalisation, i.i.d., plural=i.i.d., VA, MRE.

borne supérieure La borne supérieure d'un ensemble de nombres réels est le plus petit nombre qui est supérieur ou égal à chaque élément de cet ensemble. Plus formellement, un nombre réel a est la borne supérieure d'un ensemble $\mathcal{A} \subseteq \mathbb{R}$ si: 1) a est un majorant de $\mathcal{A}$; et 2) aucun nombre strictement plus petit que a n'est un majorant de $\mathcal{A}$. Tout ensemble non vide de nombres réels qui est majoré possède une borne supérieure, même s'il ne contient pas cette borne supérieure [2, Sec. 1.4].

caractéristique Une caractéristique d'un point de données est l'un de ses attributs pouvant être mesuré ou calculé facilement sans nécessiter de supervision humaine. Par exemple, si un point de données est une image numérique (par ex., stockée sous forme de fichier `.jpeg`), alors on peut utiliser les intensités rouge-vert-bleu de ses pixels comme caractéristiques. Les synonymes spécifiques au domaine pour ce terme incluent « covariable », « variable explicative », « variable indépendante », « variable d'entrée », « variable prédictive » ou « régressseur » [18], [19], [20].

Voir aussi: point de données.

classification La classification est la tâche qui consiste à déterminer une étiquette discrète y pour un point de données fixé, uniquement à partir de ses caractéristiques. L'étiquette y appartient à un ensemble fini, par exemple $y \in \{-1, 1\}$ ou $y \in \{1, \ldots, 19\}$, et représente la catégorie à

laquelle appartient le point de données correspondant.

Voir aussi: étiquette, point de données, caractéristique.

classifieur Un classifieur est une (fonction) hypothèse $h(\mathbf{x})$ utilisée pour prédire une étiquette prenant ses valeurs dans un ensemble fini appelé espace des étiquettes. On peut utiliser directement la valeur $h(\mathbf{x})$ comme prédiction $\hat{y}$ pour l'étiquette, mais il est courant d'utiliser une application $h(\cdot)$ produisant une quantité numérique. La prédiction est alors obtenue par un simple seuillage. Par exemple, dans un problème de classification binaire avec un espace des étiquettes $\mathcal{Y} \in \{-1, 1\}$, on peut utiliser une fonction hypothèse à valeurs réelles $h(\mathbf{x}) \in \mathbb{R}$ comme classifieur. Une prédiction $\hat{y}$ peut alors être obtenue par seuillage:

$$\hat{y} = 1 \text{ si } h(\mathbf{x}) \geq 0 \quad \text{et} \quad \hat{y} = -1 \text{ sinon.} \tag{1}$$

On peut caractériser un classifieur par ses régions de décision $\mathcal{R}_a$ pour chaque valeur possible $a \in \mathcal{Y}$ de l'étiquette.

Voir aussi: hypothèse, application, étiquette, espace des étiquettes, fonction, prédiction, classification, région de décision.

classifieur linéaire Considérons des points de données caractérisés par des caractéristiques numériques $\mathbf{x} \in \mathbb{R}^d$ et une étiquette $y \in \mathcal{Y}$ appartenant à un espace des étiquettes fini $\mathcal{Y}$. Un classifieur linéaire est caractérisé par des régions de décision séparées par des hyperplans dans $\mathbb{R}^d$ [8, Ch. 2].

Voir aussi: point de données, caractéristique, étiquette, espace des étiquettes, classifieur, région de décision.

convexe Un sous-ensemble $\mathcal{C} \subseteq \mathbb{R}^d$ de l'espace euclidien $\mathbb{R}^d$ est dit convexe s'il contient le segment de droite qui relie deux points $\mathbf{x}, \mathbf{y} \in \mathcal{C}$ quelconques de cet ensemble. Une fonction $f : \mathbb{R}^d \to \mathbb{R}$ est convexe si son épigraphe $\left\{ \left(\mathbf{w}^T, t \right)^T \in \mathbb{R}^{d+1} : t \geq f(\mathbf{w}) \right\}$ est un ensemble convexe [21]. On illustre un exemple d'ensemble convexe et de fonction convexe dans la Figure 4.

(a) (b)

Fig. 4. (a) Un ensemble convexe $\mathcal{C} \subseteq \mathbb{R}^d$. (b) Une fonction convexe $f : \mathbb{R}^d \to \mathbb{R}$.

Voir aussi: espace euclidien, fonction, épigraphe.

covariance La covariance entre deux VA réelles x et y, définies sur un même espace probabilisé, mesure leur dépendance linéaire. Elle est définie par

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

Une covariance positive indique que x et y tendent à augmenter ensemble, tandis qu'une covariance négative suggère que l'un tend à augmenter quand l'autre diminue. Si $\mathrm{cov}\,(x, y) = 0$, les VA sont dites non corrélées, bien que non nécessairement indépendantes. Voir la Figure 5 pour des exemples visuels.

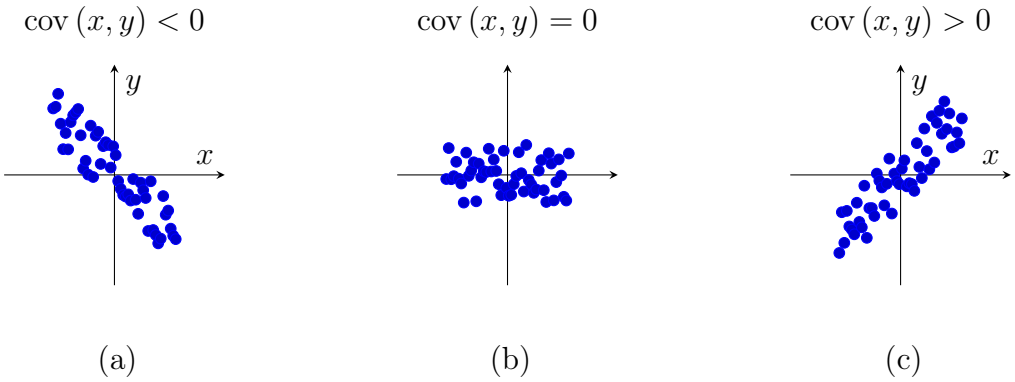


Fig. 5. Un nuage de pointss illustrant des réalisations issues de trois modèles probabilistes différents pour deux VA avec des valeurs de covariance différentes. (a) Négative. (b) Nulle. (c) Positive.

Voir aussi: modèle probabiliste, espérance.

degré d'un sommet Le degré $d^{(i)}$ d'un sommet $i \in \mathcal{V}$ dans un graphe non orienté est le nombre de voisins de ce sommet, c'est-à-dire $d^{(i)} := \left|\mathcal{N}^{(i)}\right|$.

Voir aussi: graphe, voisins.

descente de gradient La descente de gradient est une méthode itérative permettant de trouver le minimum d'une fonction dérivable $f : \mathbb{R}^d \to \mathbb{R}$. Elle génère une suite d'estimations $\mathbf{w}^{(0)}, \mathbf{w}^{(1)}, \mathbf{w}^{(2)}, \ldots$ qui (idéalement) convergent vers un minimum de f. À chaque itération k, la descente de gradient affine l'estimation actuelle $\mathbf{w}^{(k)}$ en effectuant un pas dans la direction de la plus forte pente descendante de l'approximation linéaire locale. Cette direction est donnée par le gradient négatif $\nabla f(\mathbf{w}^{(k)})$ de la fonction f à l'estimation courante $\mathbf{w}^{(k)}$. La règle de mise à jour est

alors donnée par :

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

où $\eta > 0$ est une taille de pas choisie suffisamment petite. Pour une valeur bien choisie de η, la mise à jour réduit généralement la valeur de la fonction, c'est-à-dire $f(\mathbf{w}^{(k+1)}) < f(\mathbf{w}^{(k)})$. La figure 6 illustre un pas de descente de gradient.

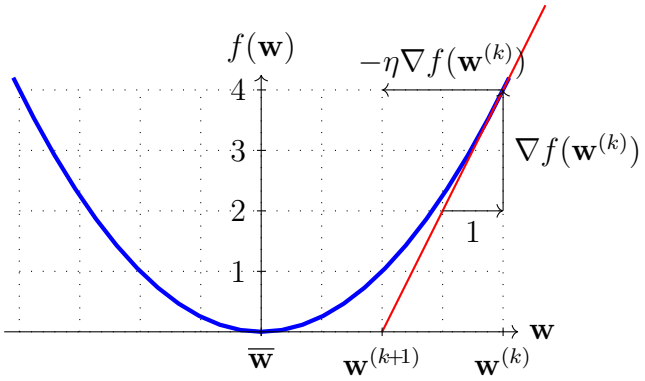


Fig. 6. Un seul pas (2) vers le minimiseur $\overline{\mathbf{w}}$ de $f(\mathbf{w})$.

Voir aussi: minimum, dérivable, pas, taille de pas, gradient.

descente de gradient avec projection Considérons une méthode basée sur la MRE qui utilise un modèle paramétré avec un espace des paramètres $\mathcal{W} \subseteq \mathbb{R}^d$. Même si la fonction objective de la MRE est régulière, nous ne pouvons pas utiliser une descente de gradient classique, car elle ne prend pas en compte les contraintes sur la variable d'optimisation (c'est-à-dire les paramètres du modèle). La descente de gradient avec projection étend la descente de gradient classique pour gérer les contraintes sur

la variable d'optimisation. Une seule itération de descente de gradient avec projection consiste à d'abord effectuer un pas, puis à projeter le résultat sur l'espace des paramètres.

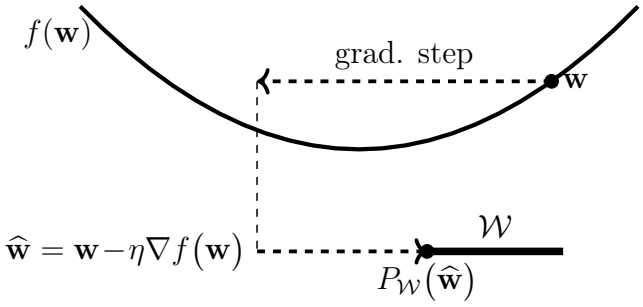


Fig. 7. La descente de gradient avec projection complète un pas de gradient classique avec une projection sur l'ensemble de contraintes $\mathcal{W}$.

Voir aussi: MRE, modèle, espace des paramètres, fonction objective, régulière, descente de gradient, paramètres du modèle, pas, projection.

descente de gradient en ligne (ou incrémentale) Considérons une méthode d'apprentissage automatique qui apprend des paramètres du modèle $\mathbf{w}$ à partir d'un espace des paramètres $\mathcal{W} \subseteq \mathbb{R}^d$. Le processus d'apprentissage utilise des points de données $\mathbf{z}^{(t)}$ arrivant à des instants successifs $t = 1, 2, \ldots$. Interprétons les points de données $\mathbf{z}^{(t)}$ comme des copies i.i.d., plural=i.i.d. d'une VA $\mathbf{z}$. Le risque $\mathbb{E}\{L(\mathbf{z}, \mathbf{w})\}$ d'une hypothèse $h^{(\mathbf{w})}$ peut alors (sous certaines conditions légères) être obtenu comme la limite $\lim_{T\to\infty}(1/T)\sum_{t=1}{}^{T} L\left(\mathbf{z}^{(t)}, \mathbf{w}\right)$. Cette limite peut être utilisée comme fonction objective pour apprendre les paramètres du modèle $\mathbf{w}$. Malheureusement, cette limite ne peut être évaluée que si l'on attend un temps infini afin de collecter tous les points de don-

nées. Certaines applications d'apprentissage automatique nécessitent des méthodes qui apprennent en ligne: dès qu'un nouveau point de données $\mathbf{z}^{(t)}$ arrive à l'instant t, on met à jour les paramètres du modèle actuels $\mathbf{w}^{(t)}$. Notons que le nouveau point de données $\mathbf{z}^{(t)}$ contribue par la composante $L\left(\mathbf{z}^{(t)}, \mathbf{w}\right)$ au risque. Comme son nom l'indique, la descente de gradient en ligne met à jour $\mathbf{w}^{(t)}$ via un pas de gradient (projeté)

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

Notons que (3) est un pas pour la composante actuelle $L\left(\mathbf{z}^{(t)}, \cdot\right)$ du risque. La mise à jour (3) ignore toutes les composantes précédentes $L\left(\mathbf{z}^{(t')}, \cdot\right)$, pour $t' < t$. Il peut donc arriver que, comparé à $\mathbf{w}^{(t)}$, les paramètres du modèle mis à jour $\mathbf{w}^{(t+1)}$ augmentent la moyenne rétrospective de la perte $\sum_{t'=1}^{t-1} L\left(\mathbf{z}^{(t')}, \cdot\right)$. Cependant, pour un taux d'apprentissage η_t judicieusement choisi, la descente de gradient en ligne peut être montrée optimale dans des contextes pertinents d'un point de vue pratique. Par optimale, on entend que les paramètres du modèle $\mathbf{w}^{(T+1)}$ fournis par la descente de gradient en ligne après avoir observé T points de données $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(T)}$ sont au moins aussi bons que ceux fournis par toute autre méthode d'apprentissage [12, 22].

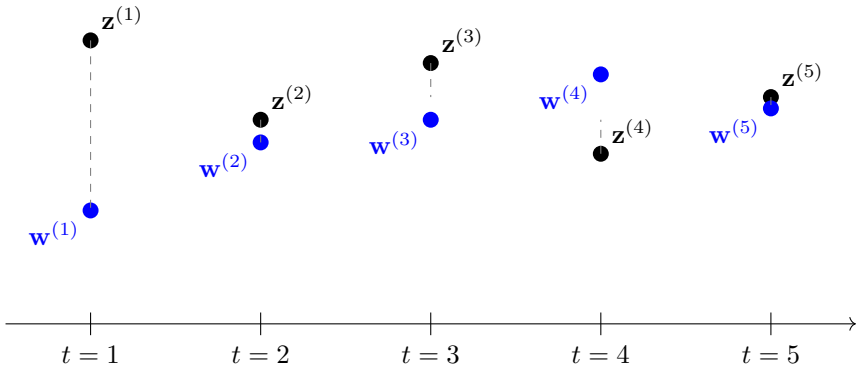


Fig. 8. Un exemple de descente de gradient en ligne qui met à jour les paramètres du modèle $\mathbf{w}^{(t)}$ en utilisant le point de données $\mathbf{z}^{(t)} = x^{(t)}$ arrivant à l'instant t. Cet exemple utilise la perte quadratique $L\left(\mathbf{z}^{(t)}, w\right) = (x^{(t)} - w)^2$.

Voir aussi: apprentissage automatique, paramètres du modèle, espace des paramètres, point de données, i.i.d., plural=i.i.d., VA, risque, hypothèse, fonction objective, descente de gradient, pas, perte, taux d'apprentissage, perte quadratique.

descente de gradient stochastique (SGD) La descente de gradient stochastique s'obtient à partir de la descente de gradient en remplaçant le gradient de la fonction objective par une approximation stochastique. Une application principale de la SGD est d'entraîner un modèle paramétré via la MRE sur un ensemble d'entraînement $\mathcal{D}$ qui est soit très grand, soit difficilement accessible (par exemple, lorsque les points de données sont stockés dans une base de données répartie dans le monde entier). Pour évaluer le gradient du risque empirique (en tant que fonction des paramètres du modèle $\mathbf{w}$), il faut calculer la somme $\sum_{r=1}^{m} \nabla_{\mathbf{w}} L\left(\mathbf{z}^{(r)}, \mathbf{w}\right)$

sur tous les points de données de l'ensemble d'entraînement. On obtient une approximation stochastique du gradient en remplaçant la somme $\sum_{r=1}^{m} \nabla_{\mathbf{w}} L\left(\mathbf{z}^{(r)}, \mathbf{w}\right)$ par une somme $\sum_{r \in \mathcal{B}} \nabla_{\mathbf{w}} L\left(\mathbf{z}^{(r)}, \mathbf{w}\right)$ sur un sous-ensemble $\mathcal{B} \subseteq \{1, \ldots, m\}$ choisi aléatoirement (voir Fig. 9). On appelle souvent ces points de données choisis aléatoirement un lot. La taille du lot $|\mathcal{B}|$ est un paramètre important de la SGD. Une SGD avec $|\mathcal{B}| > 1$ est appelée SGD par mini-lots [23].

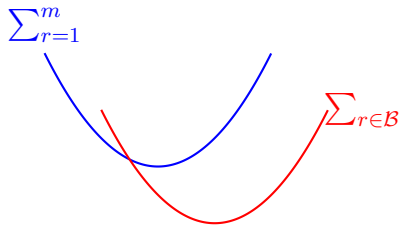


Fig. 9. La descente de gradient stochastique pour la MRE approxime le gradient $\sum_{r=1}^{m} \nabla_{\mathbf{w}} L\left(\mathbf{z}^{(r)}, \mathbf{w}\right)$ en remplaçant la somme sur tous les points de données de l'ensemble d'entraînement (indexés par $r = 1, \ldots, m$) par une somme sur un sous-ensemble aléatoire $\mathcal{B} \subseteq \{1, \ldots, m\}$.

Voir aussi: descente de gradient, gradient, fonction objective, stochastique, modèle, MRE, ensemble d'entraînement, point de données, risque empirique, fonction, paramètres du modèle, lot, paramètre.

dimension effective La dimension effective $d_{\text{eff}}(\mathcal{H})$ d'un espace des hypothèses infini $\mathcal{H}$ est une mesure de sa taille. Grosso modo, la dimension effective correspond au nombre effectif de paramètres du modèle ajustables indépendants. Ces paramètre peuvent être les coefficients utilisés dans une application linéaire ou les poids et termes de biais d'un

RNA.

Voir aussi: espace des hypothèses, paramètres du modèle, RNA, biais.

divergence Considérons une application d'apprentissage fédéré avec des données en réseau représentées par un réseau d'apprentissage fédéré. Les méthodes d'apprentissage fédéré utilisent une mesure de divergence pour comparer des fonctions hypothèse issues de modèles locaux aux sommets i, i' liés par une arête dans le réseau d'apprentissage fédéré.

Voir aussi : apprentissage fédéré, réseau d'apprentissage fédéré, modèle local.

donnée sensible L'apprentissage automatique vise à apprendre une hypothèse permettant de prédire l'étiquette d'un point de données à partir de ses caractéristiques. Dans certains contextes, il est essentiel que la sortie produite par le système d'apprentissage automatique ne permette pas de déduire des données sensibles associées à un point de données. Le caractère sensible d'une donnée dépend du domaine d'application, et peut inclure, par exemple, des informations liées à la santé, à l'origine ethnique ou aux opinions politiques.

Voir aussi : apprentissage automatique, hypothèse, application, étiquette, point de données, caractéristique.

données Les données désignent des objets porteurs d'information. Ces objets peuvent être soit des entités physiques concrètes (comme des personnes ou des animaux), soit des concepts abstraits (comme des nombres). On utilise souvent des représentations (ou approximations) des données originales qui sont plus pratiques pour le traitement. Ces

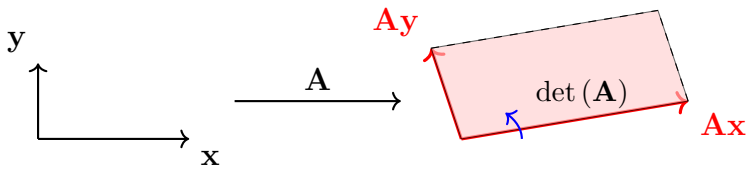
approximations utilisent différentes structures mathématiques, telles que les relations utilisées dans les bases de données relationnelles [24], [25]. Voir aussi: modèle, jeu de données, point de données.

données en réseau Les données en réseau sont constituées de jeux de données locaux liés par une notion de similarité deux à deux. On peut représenter les données en réseau à l'aide d'un graphe dont les sommets portent des jeux de données locaux et dont les arêtes codent les similarités deux à deux. Un exemple typique de données en réseau apparaît dans les applications d'apprentissage fédéré où les jeux de données locaux sont générés par des appareils distribués spatialement. Voir aussi : données, jeu de données local, graphe, apprentissage fédéré, appareil.

dérivable Une fonction à valeurs réelles $f : \mathbb{R}^d \to \mathbb{R}$ est dite dérivable si elle peut, en tout point, être approchée localement par une fonction linéaire. L'approximation linéaire locale au point $\mathbf{x}$ est déterminée par le gradient $\nabla f(\mathbf{x})$ [2].
Voir aussi: fonction, gradient.

déterminant Le déterminant $\det(\mathbf{A})$ d'une matrice carrée $\mathbf{A} \in \mathbb{R}^{n \times n}$ est un scalaire qui caractérise la façon dont les volumes (et leur orientation) dans $\mathbb{R}^n$ sont modifiés par l'application de $\mathbf{A}$ [3], [26]. Notons qu'une matrice $\mathbf{A}$ représente une transformation linéaire sur $\mathbb{R}^n$. En particulier, $\det(\mathbf{A}) > 0$ préserve l'orientation, $\det(\mathbf{A}) < 0$ inverse l'orientation, et $\det(\mathbf{A}) = 0$ annule complètement le volume, indiquant que $\mathbf{A}$ n'est pas inversible. Le déterminant vérifie aussi $\det(\mathbf{AB}) = \det(\mathbf{A}) \cdot \det(\mathbf{B})$,

et si $\mathbf{A}$ est diagonalisable avec pour valeurs propres $\lambda_1, \ldots, \lambda_n$, alors $\det(\mathbf{A}) = \prod_{i=1}^{n} \lambda_i$ [27]. Pour les cas particuliers $n = 2$ (2D) et $n = 3$ (3D), le déterminant peut s'interpréter comme une aire orientée ou un volume engendré par les vecteurs colonnes de $\mathbf{A}$.



Voir aussi: valeur propre, matrice inverse.

empoisonnement de données L'empoisonnement de données désigne la manipulation (ou la fabrication) intentionnelle de points de données afin d'influencer l'apprentissage d'un modèle d'apprentissage automatique [28], [29]. Les attaques par empoisonnement prennent plusieurs formes, notamment :

- Attaque par porte dérobée : Implantation de déclencheurs dans les données d'entraînement, de sorte que le modèle entraîné fonctionne normalement sur des vecteurs de caractéristiques typiques, mais produise une mauvaise classification lorsqu'un motif déclencheur est présent.
- Attaque par déni de service : Dégradation des performances globales du modèle entraîné via l'injection d'exemples mal étiquetés ou adverses pour perturber l'apprentissage.

L'empoisonnement de données est particulièrement préoccupant dans des contextes d'apprentissage automatique décentralisé ou distribué (comme en apprentissage fédéré), où les données d'entraînement ne peuvent pas être vérifiées de manière centralisée.

Voir aussi : attaque, porte dérobée, attaque par déni de service, IA digne de confiance.

ensemble d'entraînement (ou d'apprentissage) Un ensemble d'entraînement est un jeu de données $\mathcal{D}$ composé de certains points de données utilisés dans le cadre d'une MRE pour apprendre une hypothèse $\hat{h}$. La perte moyenne de $\hat{h}$ sur l'ensemble d'entraînement est appelée erreur d'entrainement. La comparaison entre l'erreur d'entrainement et l'erreur de validation de $\hat{h}$ permet de évaluer la qualité de la méthode d'apprentissage automatique utilisée et fournit des indications pour améliorer l'erreur de validation (par exemple, en utilisant un autre espace des hypothèses ou en collectant plus de points de données) [8, Sec. 6.6].

Voir aussi: jeu de données, point de données, MRE, hypothèse, perte, erreur d'entrainement, erreur de validation, apprentissage automatique, espace des hypothèses.

ensemble de test (ou jeu de test) Un ensemble de points de données qui n'ont été utilisés ni pour entraîner un modèle (par exemple via MRE), ni dans un ensemble de validation pour la sélection entre différents modèles.

Voir aussi: point de données, modèle, MRE, ensemble de validation.

ensemble de validation (ou jeu de validation) Un ensemble de points de données utilisé pour estimer le risque d'une hypothèse $\hat{h}$ apprise par une méthode d'apprentissage automatique (par exemple, par résolution d'un problème de MRE). La perte moyenne de $\hat{h}$ sur l'ensemble de validation est appelée erreur de validation et peut servir à évaluer les performances d'une méthode d'apprentissage (voir [8, Sec. 6.6]). La comparaison entre erreur d'entrainement et erreur de validation peut guider des améliorations de la méthode (telles que le choix d'un autre espace des hypothèses).

Voir aussi: point de données, risque, hypothèse, apprentissage automatique, MRE, perte, validation, erreur de validation, erreur d'entrainement, espace des hypothèses.

entropie L'entropie quantifie l'incertitude ou l'imprévisibilité associée à une VA [30]. Pour une VA discrète x prenant ses valeurs dans un ensemble fini $\mathcal{S} = \{x_1, \ldots, x_n\}$ avec une fonction de masse $p_i := \mathbb{P}(x = x_i)$, l'entropie est définie par

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

L'entropie est maximale lorsque toutes les issues sont équiprobables, et minimale (i.e., nulle) lorsque l'issue est déterministe. Une généralisation du concept d'entropie pour les VA continues est l'entropie différentielle.

Voir aussi: incertitude, modèle probabiliste, entropie différentielle.

entropie différentielle Pour une VA à valeurs réelles $\mathbf{x} \in \mathbb{R}^d$ avec une fonction de densité de probabilité $p(x)$, l'entropie différentielle est définie

par [30]:

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

L'entropie différentielle peut être négative et ne possède pas certaines propriétés de l'entropie des VA à valeurs discrètes, notamment l'invariance par changement de variables [30]. Parmi toutes les VA ayant une moyenne $\boldsymbol{\mu}$ et une matrice de covariance $\boldsymbol{\Sigma}$ données, $h(\mathbf{x})$ différentielle $h(\mathbf{x})$ est maximisée par $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

Voir aussi: incertitude, modèle probabiliste.

erreur d'entrainement La perte moyenne d'une hypothèse lors de la prédiction des étiquettes des points de données dans un ensemble d'entraînement. On désigne parfois aussi par erreur d'entraînement la perte moyenne minimale qui est atteinte par une solution de MRE.

Voir aussi: perte, hypothèse, étiquette, point de données, ensemble d'entraînement, MRE.

erreur de validation Considérons une hypothèse $\hat{h}$ obtenue à l'aide d'une méthode d'apprentissage automatique, par exemple en résolvant un problème de MRE sur un ensemble d'entraînement. La perte moyenne de $\hat{h}$ sur un ensemble de validation, distinct de l'ensemble d'entraînement, est appelée erreur de validation.

Voir aussi: hypothèse, apprentissage automatique, MRE, ensemble d'entraînement, perte, ensemble de validation, validation.

espace de Hilbert Un espace de Hilbert est un espace préhilbertien complet [31]. Autrement dit, c'est un espace vectoriel muni d'un produit scalaire défini entre paires de vecteurs, et qui satisfait à la condition

supplémentaire de complétude, c'est-à-dire que toute suite de Cauchy de vecteurs converge vers une limite appartenant à l'espace. Un exemple canonique d'espace de Hilbert est l'espace euclidien $\mathbb{R}^d$, pour une certaine dimension d, constitué de vecteurs $\mathbf{u} = \big(u_1, \ldots, u_d\big)^T$ et muni du produit scalaire standard $\mathbf{u}^T\mathbf{v}$.

Voir aussi : espace vectoriel, espace euclidien.

espace des caractéristiques L'espace des caractéristiques d'une application ou méthode d'apprentissage automatique correspond à l'ensemble de toutes les valeurs possibles que peut prendre le vecteur de caractéristiques d'un point de données. Un choix largement utilisé pour l'espace des caractéristiques est l'espace euclidien $\mathbb{R}^d$, où la dimension d représente le nombre de caractéristiques individuelles d'un point de données.

Voir aussi: apprentissage automatique, vecteur de caractéristiques, point de données, caractéristique, espace euclidien.

espace des hypothèses Toute méthode pratique d'apprentissage automatique utilise un espace des hypothèses (ou modèle) $\mathcal{H}$. L'espace des hypothèses d'une méthode d'apprentissage automatique est un sous-ensemble de l'ensemble des applications allant de l'espace des caractéristiques dans l'espace des étiquettes. Le choix de cet espace doit tenir compte des ressources informatiques disponibles ainsi que des aspects statistiques. Si l'infrastructure permet des opérations matricielles efficaces, et qu'il existe une relation (approximativement) linéaire entre un ensemble de caractéristiques et une étiquette, un choix pertinent pour

l'espace des hypothèses peut être un modèle linéaire.

Voir aussi: apprentissage automatique, hypothèse, modèle, application, espace des caractéristiques, espace des étiquettes, aspects statistiques, caractéristique, étiquette, modèle linéaire.

espace des paramètres L' espace des paramètres $\mathcal{W}$ d'un modèle d'apprentissage automatique $\mathcal{H}$ est l'ensemble de tous les choix possibles pour les paramètres du modèle (voir Figure 10). De nombreuses méthodes importantes en apprentissage automatique utilisent un modèle paramétré par des vecteurs de l'espace euclidien $\mathbb{R}^d$. Deux exemples courants de modèles paramétrés sont les modèles linéaires et les réseaux de neurones profonds. L'espace des paramètres est alors souvent un sous-ensemble $\mathcal{W} \subseteq \mathbb{R}^d$, par exemple tous les vecteurs $\mathbf{w} \in \mathbb{R}^d$ dont la norme est inférieure à un.

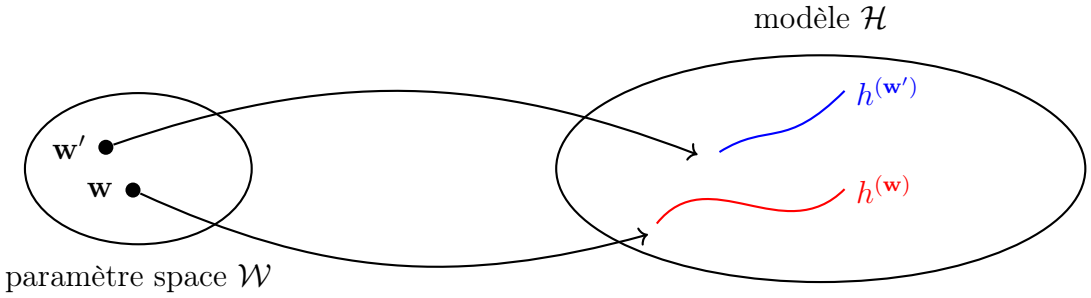


Fig. 10. L'espace des paramètres $\mathcal{W}$ d'un modèle d'apprentissage automatique $\mathcal{H}$ contient tous les choix possibles pour les paramètres du modèle. Chaque choix $\mathbf{w}$ pour les paramètres du modèle sélectionne une hypothèse $h^{(\mathbf{w})} \in \mathcal{H}$.

Voir aussi: paramètre, modèle, paramètres du modèle.

espace des étiquettes Considérons une application d'apprentissage automatique impliquant des points de données caractérisés par des caractéristiques et des étiquettes. L'espace des étiquettes est constitué de toutes les valeurs possibles que l'étiquette d'un point de données peut prendre. Les méthodes de régression, visant à prédire des étiquettes numériques, utilisent souvent l'espace des étiquettes $\mathcal{Y} = \mathbb{R}$. Les méthodes de classification binaire utilisent un espace des étiquettes constitué de deux éléments différents, par exemple $\mathcal{Y} = \{-1, 1\}$, $\mathcal{Y} = \{0, 1\}$, ou .
Voir aussi: apprentissage automatique, point de données, caractéristique, étiquette, régression, classification.

espace euclidien L'espace euclidien $\mathbb{R}^d$ de dimension $d \in \mathbb{N}$ est constitué des vecteurs $\mathbf{x} = \big(x_1, \ldots, x_d\big)$, avec d composantes réelles $x_1, \ldots, x_d \in \mathbb{R}$. Un tel espace euclidien est muni d'une structure géométrique définie par le produit scalaire $\mathbf{x}^T\mathbf{x}' = \sum_{j=1}^{d} x_j x'_j$ entre deux vecteurs $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$ quelconques [2].

espace probabilisé Un espace probabilisé est une structure mathématique qui permet de raisonner sur une expérience aléatoire, par exemple l'observation d'un phénomène physique. Formellement, un espace probabilisé $\mathcal{P}$ est un triplet $(\Omega, \mathcal{F}, \mathbb{P}(\cdot))$ où

- Ω est un espace échantillon contenant tous les résultats possibles d'une expérience aléatoire ;
- $\mathcal{F}$ est une tribu (ou σ-algèbre), c'est-à-dire une collection de sous-ensembles de Ω (appelés événements) qui satisfait certaines propriétés de fermeture par opérations sur les ensembles ;

- $\mathbb{P}(\cdot)$ est une loi de probabilité, c'est-à-dire une fonction qui attribue une probabilité $P(\mathcal{A}) \in [0, 1]$ à chaque événement $\mathcal{A} \in \mathcal{F}$. Cette fonction doit satisfaire $\mathbb{P}(\Omega) = 1$ et $\mathbb{P}(\bigcup_{i=1}^{\infty} \mathcal{A}_i) = \sum_{i=1}^{\infty} \mathbb{P}(\mathcal{A}_i)$ pour toute suite dénombrable d'événements deux à deux disjoints $\mathcal{A}_1, \mathcal{A}_2, \ldots$ dans $\mathcal{F}$.

Les espaces probabilisés sont la base des modèles probabilistes utilisés pour étudier le comportement des méthodes d'apprentissage automatique [6], [32], [33].

Voir aussi : probabilité, VA, incertitude.

espace vectoriel Un espace vectoriel est un famille d'éléments (appelés vecteurs) stable par addition vectorielle et multiplication scalaire, c'est-à-dire:

- Si $\mathbf{x}, \mathbf{y} \in \mathcal{V}$, alors $\mathbf{x} + \mathbf{y} \in \mathcal{V}$.
- Si $\mathbf{x} \in \mathcal{V}$ et $c \in \mathbb{R}$, alors $c\mathbf{x} \in \mathcal{V}$.
- En particulier, le vecteur nul $\mathbf{0} \in \mathcal{V}$.

L'espace euclidien $\mathbb{R}^n$ est un espace vectoriel. Les modèles linéaires et les applications linéaires opèrent dans de tels espaces.

Voir aussi: espace euclidien, modèle linéaire, application linéaire.

espace échantillon Un espace échantillon est l'ensemble de toutes les issues possibles d'une expérience aléatoire [6, 7, 34, 35].

Voir aussi : espace probabilisé.

espérance Considérons un vecteur de caractéristiques numérique $\mathbf{x} \in \mathbb{R}^d$, que l'on interprète comme une réalisation d'une VA suivant une loi

de probabilité $p(\mathbf{x})$. L'espérance de $\mathbf{x}$ est définie comme l'intégrale $\mathbb{E}\{\mathbf{x}\} := \int \mathbf{x} p(\mathbf{x})$. Notons que cette espérance n'est définie que si cette intégrale existe, c'est-à-dire si la VA est intégrable [2], [6], [36]. La figure 11 illustre l'espérance d'une VA discrète scalaire x prenant ses valeurs dans un ensemble fini.

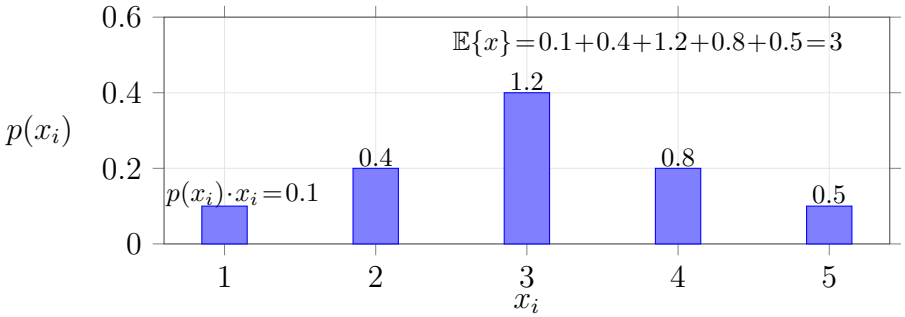


Fig. 11. L'espérance d'une VA discrète x s'obtient en sommant les valeurs possibles x_i, pondérées par leur probabilité correspondante $p(x_i) = \mathbb{P}(x = x_i)$.

Voir aussi: vecteur de caractéristiques, réalisation, VA, loi de probabilité, probabilité.

estimateur bayésien Considérons un modèle probabiliste avec une loi de probabilité conjointe $p(\mathbf{x}, y)$ pour les caractéristiques $\mathbf{x}$ et l'étiquette y d'un point de données. Pour une fonction de perte donnée $L(\cdot, \cdot)$, on appelle une hypothèse h un estimateur bayésien si son risque $\mathbb{E}\{L((\mathbf{x}, y), h)\}$ est le minimum atteignable [37]. Notons que la propriété d'être un estimateur bayésien dépend de la loi de probabilité sous-jacente ainsi que du choix de la fonction de perte $L(\cdot, \cdot)$.

Voir aussi: modèle probabiliste, hypothèse, risque

expert En apprentissage automatique, l'objectif est d'apprendre une hypothèse h capable de prédire avec précision l'étiquette d'un point de données à partir de ses caractéristiques. L'erreur de prédiction est mesurée à l'aide d'une fonction de perte. Idéalement, on cherche à obtenir une hypothèse minimisant la perte sur tout point de données. On peut préciser cet objectif informel avec l'hypothèse d'indépendance et de distribution identique (hypothèse i.i.d.) et le risque bayésien, qui sert de niveau de référence pour la perte moyenne d'une hypothèse. Une autre approche pour définir une niveau de référence consiste à utiliser l'hypothèse h' apprise par une méthode d'apprentissage automatique existante. On appelle alors cette hypothèse h' un expert [11]. Les méthodes de minimisation du regret cherchent à apprendre une hypothèse dont la perte est comparable à celle du meilleur expert [11, 12].
Voir aussi: fonction de perte, niveau de référence, regret.

explicabilité On définit l'explicabilité (subjective) d'une méthode d'apprentissage automatique comme le niveau de simulabilité [38] des prédictions fournies par un système d'apprentissage automatique à un utilisateur humain. Des mesures quantitatives de l'explicabilité (subjective) d'un modèle entraîné peuvent être construites en comparant ses prédictions avec les prédictions fournies par un utilisateur sur un ensemble de test [38, 39]. Alternativement, on peut utiliser des modèles probabilistes pour les données et mesurer l'explicabilité d'un modèle d'apprentissage automatique entraîné via l'entropie différentielle (ou conditionnelle) de ses prédictions, étant donné les prédictions de l'utilisateur [40, 41].
Voir aussi: IA digne de confiance, régularisation.

explication Une manière d'améliorer la transparence d'une méthode d'apprentissage automatique pour un utilisateur humain est de fournir une explication en complément des prédictions renvoyées par la méthode. Les explications peuvent prendre plusieurs formes. Par exemple, elles peuvent consister en un texte lisible par un humain ou en des indicateurs quantitatifs comme les scores d'importance des caractéristiques individuelles d'un point de données donné [42]. Autrement, les explications peuvent être visuelles, comme des cartes d'intensité mettant en évidence les régions d'une image influençant la prédiction [43]. La figure 12 illustre deux types d'explications. Le premier est une approximation linéaire locale $g(\mathbf{x})$ d'un modèle non linéaire entraîné $\hat{h}(\mathbf{x})$ autour d'un certain vecteur de caractéristiques $\mathbf{x}'$, comme le propose la méthode LIME. Le second type d'explication présenté dans la figure est un ensemble clairsemé de prédictions $\hat{h}(\mathbf{x}^{(1)}), \hat{h}(\mathbf{x}^{(2)}), \hat{h}(\mathbf{x}^{(3)})$ évaluées en quelques vecteurs de caractéristiques choisis, servant de points de référence concrets pour l'utilisateur.

Voir aussi : explicabilité, IA digne de confiance.

Explications locales interprétables et agnostiques au modèle (LIME) Considérons un modèle entraîné (ou une hypothèse apprise) $\widehat{h} \in \mathcal{H}$, qui associe le vecteur de caractéristiques d'un point de données à la prédiction $\widehat{y} = \widehat{h}$. Les explications locales interprétables et agnostiques au modèle (LIME) sont une technique permettant d'expliquer le comportement de $\widehat{h}$ localement autour d'un point de données de vecteur de caractéristiques $\mathbf{x}^{(0)}$ [44]. L'explication est donnée sous la forme d'une approximation locale $g \in \mathcal{H}'$ de $\widehat{h}$ (voir Fig. 13). Cette

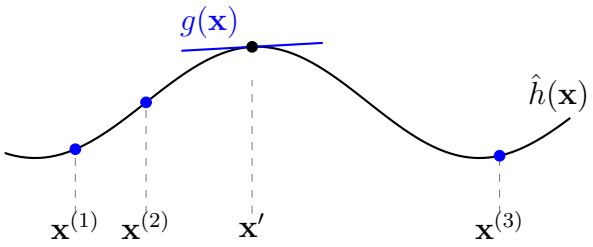


Fig. 12. Un modèle entraîné $\hat{h}(\mathbf{x})$ peut être expliqué localement au voisinage d'un point $\mathbf{x}'$ à l'aide d'une approximation linéaire $g(\mathbf{x})$. Pour un $\hat{h}(\mathbf{x})$ dérivable, cette approximation est déterminée par le gradient $\nabla\hat{h}(\mathbf{x}')$. Une autre forme d'explication consiste à afficher les valeurs de la fonction $\hat{h}\big(\mathbf{x}^{(r)}\big)$ pour $r = 1, 2, 3$.

approximation peut être obtenue par une instance de MRE avec un ensemble d'entraînement soigneusement conçu. En particulier, l'ensemble d'entraînement est composé de points de données ayant un vecteur de caractéristiques $\mathbf{x}$ proche de $\mathbf{x}^{(0)}$ et une (pseudo-)étiquette $\widehat{h}(\mathbf{x})$. Remarquons que l'on peut utiliser un modèle $\mathcal{H}'$ différent du modèle original $\mathcal{H}$ pour l'approximation. Par exemple, on peut utiliser un arbre de décision pour approximer localement un réseau de neurones profond. Un autre choix très courant pour $\mathcal{H}'$ est le modèle linéaire.

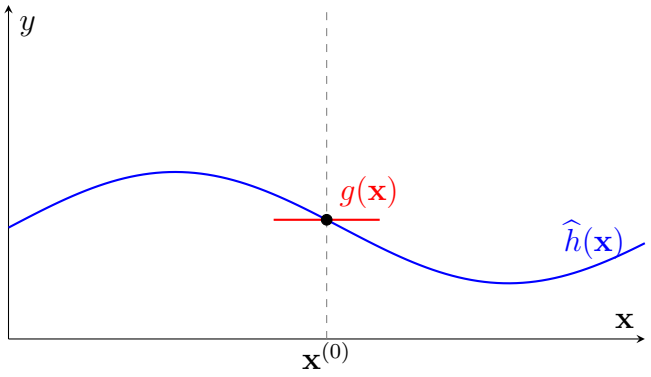


Fig. 13. Pour expliquer (comprendre) un modèle entraîné $\widehat{h} \in \mathcal{H}$, autour d'une vecteur de caractéristiques donnée $\mathbf{x}^{(0)}$, on peut utiliser une approximation locale $g \in \mathcal{H}'$.

Voir aussi: modèle, explication, MRE, ensemble d'entraînement, étiquette, arbre de décision, réseau de neurones profond, modèle linéaire.

fonction Une fonction entre deux ensembles $\mathcal{U}$ et $\mathcal{V}$ associe à chaque élément $u \in \mathcal{U}$ un unique élément $v \in \mathcal{V}$ [2]. On écrit $f : \mathcal{U} \to \mathcal{V}$, où $\mathcal{U}$ est le domaine de définition et $\mathcal{V}$ l'ensemble d'arrivée de f. Autrement dit, une fonction f définit une sortie unique $f(u) \in \mathcal{V}$ pour chaque entrée $u \in \mathcal{U}$.

fonction caractéristique La fonction caractéristique d'une VA réelle x est la fonction [6, Sec. 26]

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

La fonction caractéristique détermine de manière unique la loi de probabilité de x.

Voir aussi: VA, loi de probabilité.

fonction d'activation On associe à chaque neurone artificiel dans un RNA une fonction d'activation $\sigma(\cdot)$ qui prend en entrée une combinaison pondérée des entrées du neurone $x_1, \ldots, x_d$ et produit une sortie unique $a = \sigma\big(w_1 x_1 + \ldots + w_d x_d\big)$. Notons que chaque neurone est paramétré par les poids $w_1, \ldots, w_d$.

Voir aussi: RNA, fonction, poids.

fonction de densité de probabilité La fonction de densité de probabilité $p(x)$ d'une VA réelle $x \in \mathbb{R}$ est une représentation particulière de sa loi de probabilité. Si la fonction de densité de probabilité existe, elle peut être utilisée pour calculer la probabilité que x prenne une valeur dans un ensemble (mesurable) $\mathcal{B} \subseteq \mathbb{R}$ avec $\mathbb{P}(x \in \mathcal{B}) = \int_{\mathcal{B}} p(x')dx'$ [7, Ch. 3]. La fonction de densité de probabilité d'une VA vectorielle $\mathbf{x} \in \mathbb{R}^d$ (si elle existe) permet de calculer la probabilité que $\mathbf{x}$ appartienne à une région (mesurable) $\mathcal{R}$ avec $\mathbb{P}(\mathbf{x} \in \mathcal{R}) = \int_{\mathcal{R}} p(\mathbf{x}')dx'_1 \ldots dx'_d$ [7, Ch. 3].

Voir aussi: VA, loi de probabilité, probabilité.

fonction de perte (ou de coût) Une fonction de perte (ou de coût) est une application

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

Elle associe un réel positif ou nul (i.e., la perte) $L\left((\mathbf{x}, y), h\right)$ à une paire composée d'un point de données, de caractéristiques $\mathbf{x}$ et étiquette y, et d'une hypothèse $h \in \mathcal{H}$. La valeur $L\left((\mathbf{x}, y), h\right)$ mesure l'écart entre l'étiquette réelle y et la prédiction $h(\mathbf{x})$. Des valeurs plus faibles (proches

de zéro) de $L\left((\mathbf{x}, y), h\right)$ indiquent un écart plus faible entre la prédiction $h(\mathbf{x})$ et l'étiquette y. La figure 14 représente une fonction de perte pour un point de données donné, de caractéristiques $\mathbf{x}$ et d'étiquette y, en fonction de l'hypothèse $h \in \mathcal{H}$.

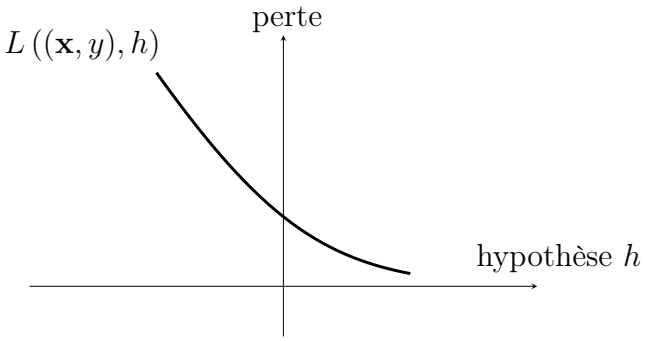


Fig. 14. Une fonction de perte $L\left((\mathbf{x}, y), h\right)$ pour un point de données fixé, de vecteur de caractéristiques $\mathbf{x}$ et d'étiquette y, et une hypothèse variable h. Les méthodes d'apprentissage automatique cherchent à trouver (ou apprendre) une hypothèse minimisant la perte.

Voir aussi: perte, fonction, application, point de données, caractéristique, étiquette, hypothèse, prédiction, vecteur de caractéristiques, apprentissage automatique.

fonction objective Une fonction objective est une application qui associe une valeur numérique $f(\mathbf{w})$ à chaque choix $\mathbf{w}$ d'une variable que l'on souhaite optimiser (voir Fig. 15). Dans le contexte de l'apprentissage automatique, la variable d'optimisation peut être les paramètres du modèle d'une hypothèse $h^{(\mathbf{w})}$. Parmi les fonctions objectives courantes, on trouve le risque (c'est-à-dire la perte espérée) ou le risque em-

pirique (c'est-à-dire la perte moyenne sur un ensemble d'entraînement). Les méthodes d'apprentissage automatique utilisent des techniques d'optimisation, telles que les méthodes basées sur le gradient, pour trouver le choix $\mathbf{w}$ qui minimise ou maximise la valeur de la fonction objective.

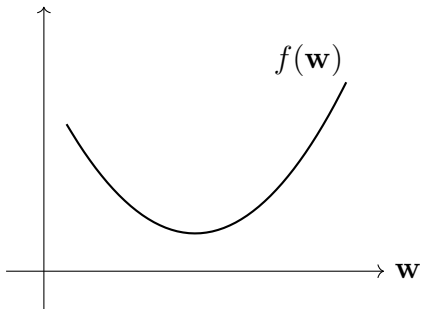


Fig. 15. Une fonction objective associe à chaque valeur possible $\mathbf{w}$ d'une variable d'optimisation, comme les paramètres d'un modèle d'apprentissage automatique, une valeur mesurant l'utilité de $\mathbf{w}$.

Voir aussi : fonction, application, apprentissage automatique, paramètres du modèle, hypothèse, risque, perte, risque empirique, ensemble d'entraînement, méthodes basées sur le gradient, minimum, maximum, modèle, fonction de perte.

frontière de décision Considérons une fonction hypothèse h qui lit un vecteur de caractéristiques $\mathbf{x} \in \mathbb{R}^d$ et renvoie une valeur à partir d'un ensemble fini $\mathcal{Y}$. La frontière de décision de h est l'ensemble des vecteurs $\mathbf{x} \in \mathbb{R}^d$ qui se trouvent entre différentes régions de décision. Plus

précisément, un vecteur $\mathbf{x}$ appartient à la frontière de décision si et seulement si chaque voisinage $\{\mathbf{x}' : \|\mathbf{x} - \mathbf{x}'\| \leq \varepsilon\}$, pour tout $\varepsilon > 0$, contient au moins deux vecteurs avec des images différentes par la fonction.

Voir aussi: hypothèse, application, vecteur de caractéristiques, région de décision, voisinage, fonction.

gradient Pour une fonction à valeurs réelles $f : \mathbb{R}^d \to \mathbb{R} : \mathbf{w} \mapsto f(\mathbf{w})$, s'il existe un vecteur $\mathbf{g}$ tel que $\lim_{\mathbf{w}\to\mathbf{w}'} \frac{f(\mathbf{w}) - \left(f(\mathbf{w}') + \mathbf{g}^T(\mathbf{w}-\mathbf{w}')\right)}{\|\mathbf{w}-\mathbf{w}'\|} = 0$, alors on le nomme le gradient de f en $\mathbf{w}'$. S'il existe, le gradient est unique et est noté $\nabla f(\mathbf{w}')$ ou $\nabla f(\mathbf{w})\big|_{\mathbf{w}'}$ [2].

Voir aussi: fonction, vecteur.

graphe Un graphe $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ est une paire qui consiste en un ensemble de sommets $\mathcal{V}$ et un ensemble d'arêtes $\mathcal{E}$. Dans sa forme la plus générale, un graphe est spécifié par une application qui associe à chaque arête $e \in \mathcal{E}$ une paire de sommets [45]. Une famille importante de graphes est celle des graphes simples non orientés. Un graphe simple non orienté est obtenu en identifiant chaque arête $e \in \mathcal{E}$ à deux sommets différents $\{i, i'\}$. Les graphes pondérés précisent également des poids numériques A_e pour chaque arête $e \in \mathcal{E}$.

Voir aussi : application, poids.

graphe d'Erdős-Rényi (graphe ER) Un graphe ER est un modèle probabiliste pour des graphes défini sur un ensemble de sommets donné $i = 1, \ldots, n$. Une manière de définir le graphe ER est via une collection de VA i.i.d., plural=i.i.d. binaires $b^{(\{i,i'\})} \in \{0, 1\}$, pour chaque

paire de sommets distincts i, i'. Une réalisation spécifique d'un graphe ER contient une arête $\{i, i'\}$ si et seulement si $b^{(\{i,i'\})} = 1$. Le graphe ER est paramétré par le nombre n de sommets et par la probabilité $\mathbb{P}(b^{(\{i,i'\})} = 1)$.

Voir aussi : graphe, modèle probabiliste, i.i.d., plural=i.i.d., VA, réalisation, probabilité.

groupe (ou cluster) Un groupe (parfois aussi appelé amas) est un sous-ensemble de points de données qui sont plus similaires entre eux qu'avec les points de données situés en dehors du groupe. La mesure quantitative de similarité entre points de données est une décision de modélisation. Si les points de données sont caractérisés par des vecteurs de caractéristiques euclidiens $\mathbf{x} \in \mathbb{R}^d$, on peut définir la similarité entre deux points de données à l'aide de la distance euclidienne entre leurs vecteurs de caractéristiques. Un exemple de tels groupes est illustré dans la Fig. 16.

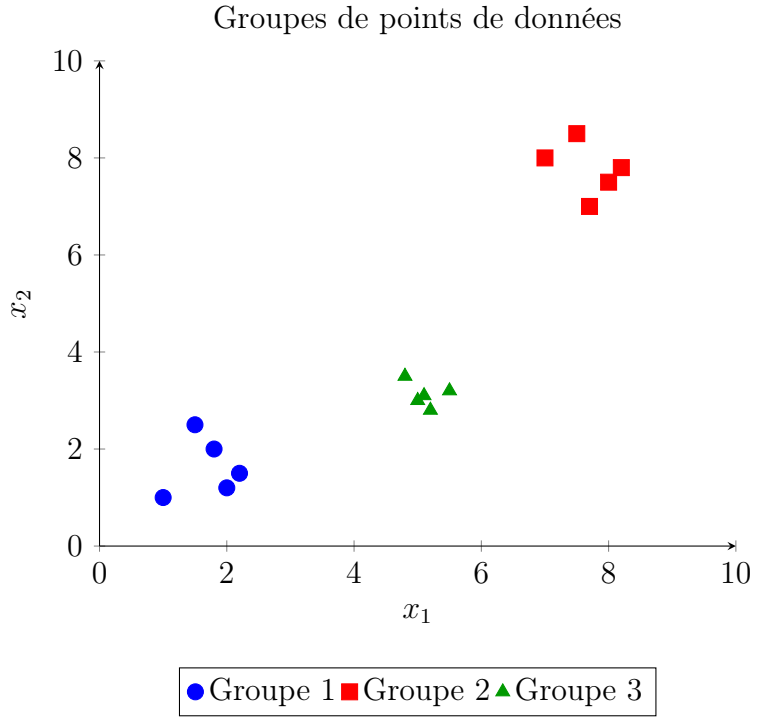


Fig. 16. Illustration de trois groupes dans un espace des caractéristiques bidimensionnel. Chaque groupe rassemble des points de données qui sont plus proches entre eux que de ceux des autres groupes, en termes de distance euclidienne.

Voir aussi : point de données, vecteur de caractéristiques, espace des caractéristiques.

généralisation La généralisation désigne la capacité d'un modèle entraîné sur un ensemble d'entraînement à produire des prédictions précises sur de nouveaux points de données jamais vus. Il s'agit d'un objectif central de l'apprentissage automatique et de l'intelligence artificielle (IA), à savoir apprendre des motifs qui s'étendent au-delà de l'ensemble

d'entraînement. La plupart des systèmes d'apprentissage automatique utilisent le MRE pour apprendre une hypothèse $\hat{h} \in \mathcal{H}$ en minimisant la perte moyenne sur un ensemble d'entraînement de points de données $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}$, noté $\mathcal{D}^{(\text{train})}$. Toutefois, le succès sur l'ensemble d'entraînement ne garantit pas le succès sur des données inconnues — cette différence constitue le défi de la généralisation.

Pour étudier la généralisation de manière mathématique, il est nécessaire de formaliser la notion de données « non vues ». Une approche largement utilisée consiste à supposer un modèle probabiliste pour la génération des données, tel que l'hypothèse i.i.d.. On interprète alors les points de données comme des VA indépendantes suivant une loi de probabilité identique $p(\mathbf{z})$. Cette loi de probabilité, supposée fixe mais inconnue, permet de définir le risque d'un modèle entraîné $\hat{h}$ comme la perte espérée :

$$\bar{L}\big(\hat{h}\big) = \mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\big\{L(\hat{h}, \mathbf{z})\big\}.$$

La différence entre le risque $\bar{L}\big(\hat{h}\big)$ et le risque empirique $\widehat{L}\big(\hat{h}|\mathcal{D}^{(\text{train})}\big)$ est appelée écart de généralisation. Des outils issus de la théorie des probabilités, tels que les inégalités de concentration et la convergence uniforme, permettent de borner cet écart sous certaines conditions [46].

Généralisation sans probabilité : La théorie des probabilités est une manière d'étudier la capacité d'un modèle à généraliser au-delà de l'ensemble d'entraînement, mais ce n'est pas la seule. On peut aussi utiliser des modifications déterministes simples sur les points de données de l'ensemble d'entraînement. L'idée de base est qu'un bon modèle $\hat{h}$ doit être robuste, c'est-à-dire que sa prédiction $\hat{h}(\mathbf{x})$ ne doit pas

beaucoup changer si on modifie légèrement les caractéristiques $\mathbf{x}$ d'un point de données $\mathbf{z}$.

Par exemple, un détecteur d'objets entraîné sur des photos prises avec un smartphone devrait toujours détecter l'objet même si quelques pixels aléatoires sont masqués [47]. De même, il devrait produire le même résultat si on fait pivoter l'objet dans l'image [48].

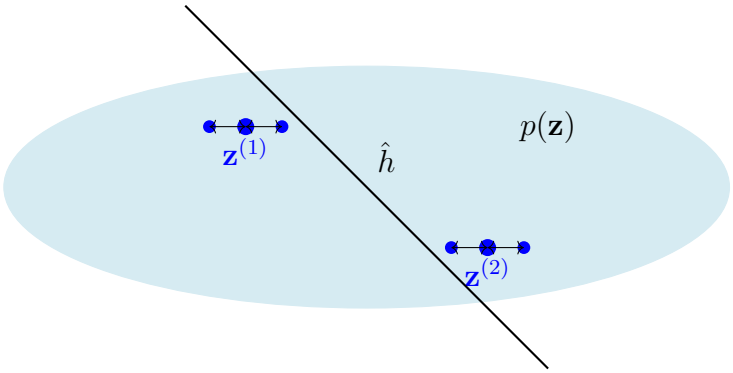


Fig. 17. Deux points de données $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}$ utilisés comme ensemble d'entraînement pour apprendre une hypothèse $\hat{h}$ via la MRE. On peut évaluer $\hat{h}$ en dehors de $\mathcal{D}^{(\text{train})}$ soit par une hypothèse i.i.d. avec une loi de probabilité sous-jacente $p(\mathbf{z})$, soit en perturbant les points de données.

Voir aussi : modèle, ensemble d'entraînement, prédiction, point de données, apprentissage automatique, IA, MRE, hypothèse, perte, données, modèle probabiliste, hypothèse i.i.d., VA, loi de probabilité, risque, risque empirique, écart de généralisation, probabilité, inégalité de concentration, caractéristique.

hypothèse Une hypothèse désigne une application (ou fonction) $h : \mathcal{X} \to \mathcal{Y}$

allant de l'espace des caractéristiques $\mathcal{X}$ vers l'espace des étiquettes $\mathcal{Y}$. Étant donné un point de données avec des caractéristiques $\mathbf{x}$, on utilise une fonction hypothèse h pour estimer (ou approximer) son étiquette y à l'aide de la prédiction $\hat{y} = h(\mathbf{x})$. L'apprentissage automatique consiste à apprendre (ou trouver) une hypothèse h telle que $y \approx h(\mathbf{x})$ pour tout point de données (de caractéristiques $\mathbf{x}$ et étiquette y).

Voir aussi: application, fonction, prédiction, modèle.

hypothèse d'indépendance et de distribution identique (hypothèse i.i.d.) L'hypothèse i.i.d., plural=i.i.d. interprète les points de données d'un jeu de données comme des réalisations de VA i.i.d., plural=i.i.d.s.

Voir aussi: i.i.d., plural=i.i.d., point de données, jeu de données, réalisation, VA.

incertitude Dans le contexte de l'apprentissage automatique, l'incertitude fait référence à la présence de multiples résultats ou explications plausibles à partir des données disponibles. Par exemple, la prédiction $\hat{h}(\mathbf{x})$ produite par un modèle d'apprentissage automatique entraîné, $\hat{h}$, reflète souvent un éventail de valeurs possibles pour la véritable étiquette d'un point de données donné. Plus cet éventail est large, plus l'incertitude associée est grande. La théorie des probabilités permet de représenter, quantifier et raisonner sur l'incertitude de manière rigoureuse d'un point de vue mathématique.

Voir aussi : modèle probabiliste, risque, entropie, variance.

indépendantes et identiquement distribuées (i.i.d.) Une collection de VA $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}$ est dite indépendante et identiquement distribuée

(i.i.d.) si chaque $\mathbf{z}^{(r)}$ suit la même loi de probabilité, et les VA sont mutuellement indépendantes, i.e. : pour toute collection d'événements $\mathcal{A}_1, \ldots, \mathcal{A}_m$, on a

$$\mathbb{P}(\mathbf{z}^{(1)} \in \mathcal{A}_1, \ldots, \mathbf{z}^{(m)} \in \mathcal{A}_m) = \prod_{r=1}^{m} \mathbb{P}(\mathbf{z}^{(r)} \in \mathcal{A}_r).$$

Voir aussi : point de données, hypothèse i.i.d., VA, modèle probabiliste.

Institut météorologique finlandais (FMI) Le FMI est une agence gouvernementale responsable de la collecte et du rapport sur les données météorologiques en Finlande.

Voir aussi: données.

intelligence artificielle (IA) L'intelligence artificielle (IA) fait référence à des systèmes qui se comportent de manière rationnelle au sens de maximiser une récompense à long terme. L'approche de l'apprentissage automatique en matière d'IA consiste à entraîner un modèle pour prédire des actions optimales. Ces prédictions sont calculées à partir d'observations sur l'état de l'environnement. Le choix de la fonction de perte distingue les applications d'IA des applications d'apprentissage automatique plus basiques. Les systèmes d'IA ont rarement accès à un ensemble d'entraînement étiqueté qui permettrait de mesurer la perte moyenne pour tout choix possible des paramètres du modèle. À la place, les systèmes d'IA utilisent des signaux de récompense observés pour obtenir une estimation (ponctuelle) de la perte engendrée par le choix actuel des paramètres du modèle.

Voir aussi: récompense, apprentissage automatique, modèle, fonction de perte, ensemble d'entraînement, perte, paramètres du modèle.

intelligence artificielle digne de confiance (IA digne de confiance)
Outre les aspects computationnels et aspects statistiques, un troisième aspect fondamental du développement des méthodes d'apprentissage automatique est leur fiabilité [49]. L'Union européenne a proposé sept exigences clés pour une IA digne de confiance (généralement basée sur des méthodes d'apprentissage automatique) [50]:

1) Facteur humain et contrôle humain ;
2) Robustesse technique et sécurité ;
3) Respect de la vie privée et gouvernance des données ;
4) Transparence ;
5) Diversité, non-discrimination et équité ;
6) Bien-être sociétal et environnemental ;
7) Responsabilisation.

Voir aussi: aspects computationnels, aspects statistiques, apprentissage automatique, IA, robustesse, données, transparence.

interprétabilité Une méthode d'apprentissage automatique est interprétable pour un utilisateur humain si celui-ci peut comprendre le processus de décision de la méthode. Une approche pour définir précisément l'interprétabilité repose sur le concept de simulabilité, c'est-à-dire la capacité d'un humain à simuler mentalement le comportement du modèle [38, 41, 51–53]. L'idée est la suivante: si un utilisateur humain

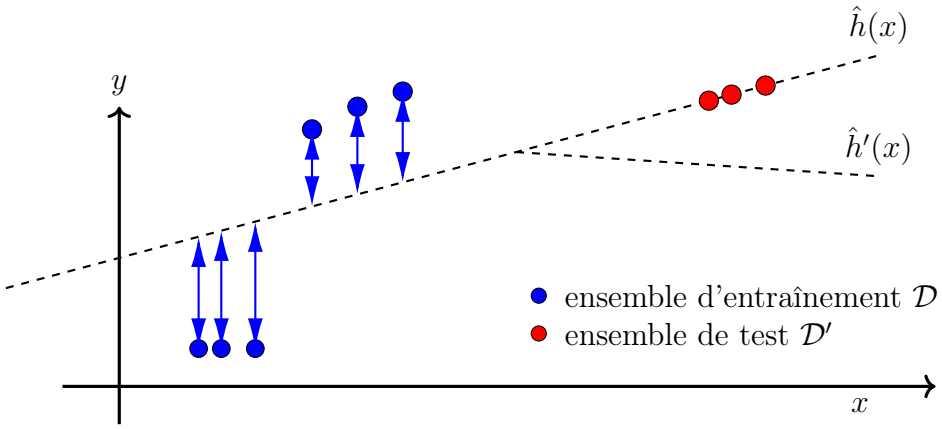


Fig. 18. Nous pouvons évaluer l'interprétabilité des modèles d'apprentissage automatique entraînés $\hat{h}$ et $\hat{h}'$ en comparant leurs prédictions aux pseudo-étiquettes générées par un utilisateur humain pour $\mathcal{D}'$.

comprend une méthode d'apprentissage automatique, alors il devrait être capable d'anticiper ses prédictions sur un ensemble de test. Nous illustrons un tel ensemble de test dans la Fig. 18 qui montre également deux hypothèses apprises, $\hat{h}$ et $\hat{h}'$. La méthode d'apprentissage automatique produisant l'hypothèse $\hat{h}$ est interprétable pour un utilisateur humain familier avec le concept de application linéaire. Puisque $\hat{h}$ correspond à une application linéaire, l'utilisateur peut anticiper les prédictions de $\hat{h}$ sur l'ensemble de test. En revanche, la méthode d'apprentissage automatique fournissant $\hat{h}'$ n'est pas interprétable, car son comportement ne correspond plus aux attentes de l'utilisateur. La notion d'interprétabilité est étroitement liée à celle d'explicabilité, car toutes deux visent à rendre les méthodes d'apprentissage automatique plus compréhensibles pour les humains. Comme illustré dans la Figure

18, l'interprétabilité d'une méthode d'apprentissage automatique $\hat{h}$ exige que l'utilisateur humain puisse anticiper ses prédictions sur un ensemble de test arbitraire. Cela contraste avec l'explicabilité, où l'utilisateur est aidé par des explications externes — comme des cartes de saillance ou des exemples de référence issus du ensemble d'entraînement — pour comprendre les prédictions de $\hat{h}$ sur un jeu de test spécifique $\mathcal{D}'$.
Voir aussi: explicabilité, IA digne de confiance, régularisation, LIME.

inversion de modèle L'inversion de modèle est une forme d'atteinte à la vie privée ciblant un système d'apprentissage automatique. Un adversaire cherche à déduire des données sensibles associées à des points de données individuels en exploitant un accès partiel au modèle entraîné $\hat{h} \in \mathcal{H}$. Cet accès consiste généralement à interroger le modèle via des entrées soigneusement choisies pour obtenir des prédictions $\hat{h}(\mathbf{x})$. Des techniques classiques d'inversion de modèle ont été illustrées dans le contexte de la classification d'images de visages, où des images sont reconstruites à partir des sorties (ou des gradients) du modèle, combinées avec des informations auxiliaires comme le nom d'une personne [54] (voir Fig. 19).

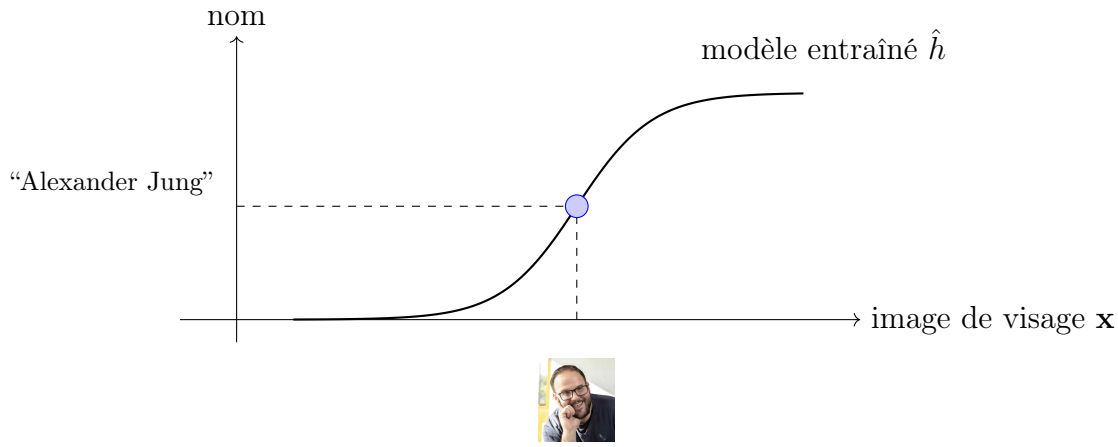


Fig. 19. Techniques d’inversion de modèle dans le contexte de la classification d’images de visage.

Voir aussi : modèle, atteinte à la vie privée, apprentissage automatique, donnée sensible, point de données, prédiction, classification, gradient, IA digne de confiance, protection de la vie privée.

inégalité de concentration Une borne supérieure sur la probabilité qu’une VA s’écarte davantage qu’un certain seuil de son espérance [55].

Voir aussi : probabilité, VA, espérance.

jeu de données Un jeu de données désigne une collection de points de données. Ces points de données portent des informations sur une certaine quantité d’intérêt (ou étiquette) dans une application d’apprentissage automatique. Les méthodes d’apprentissage automatique utilisent des jeux de données pour l’entraînement du modèle (par exemple via la MRE) et la validation du modèle.

Il est important de noter que notre notion de jeu de données est très flexible, car elle autorise des types de points de données très variés. En effet, les points de données peuvent être des objets physiques concrets (comme des humains ou des animaux) ou des objets abstraits (comme des nombres).

À titre d'exemple, la Figure 20 illustre un jeu de données utilisant des vaches comme points de données.

Fig. 20. Un troupeau de vaches dans les Alpes

Bien souvent, un ingénieur en apprentissage automatique n'a pas d'accès direct à un jeu de données. En effet, accéder au jeu de données de la Figure 20 impliquerait de visiter le troupeau de vaches dans les Alpes. À la place, il faut utiliser une approximation (ou représentation) du jeu de données plus pratique à manipuler.

Divers modèles mathématiques ont été développés pour représenter ou approximer les jeux de données [25], [56], [57], [58].

L'un des modèles de données les plus utilisés est le modèle relationnel, qui organise les données sous forme de tableau (ou relation) [24], [25]. Un tableau est composé de lignes et de colonnes:

- Chaque ligne du tableau représente un seul point de données.
- Chaque colonne du tableau correspond à un attribut spécifique du point de données. Les méthodes d'apprentissage automatique peuvent utiliser ces attributs comme caractéristiques ou étiquettes du point de données.

Par exemple, la Table 1 montre une représentation du jeu de données de la Figure 20. Dans le modèle relationnel, l'ordre des lignes est sans importance, et chaque attribut (colonne) doit être défini précisément par un domaine spécifiant l'ensemble des valeurs possibles.
Dans les applications de l'apprentissage automatique, ces domaines d'attributs deviennent l'espace des caractéristiques et l'espace des étiquettes.

Nom	**Poids**	**Âge**	**Taille**	**Température de l'estomac**
Zenzi	100	4	100	25
Berta	140	3	130	23
Resi	120	4	120	31

Table 1: Une relation (ou table) représentant le jeu de données de la Figure 20.

Bien que le modèle relationnel soit utile pour de nombreuses applications en apprentissage automatique, il peut s'avérer insuffisant vis-à-vis des exigences en matière de IA digne de confiance.
Des approches modernes, telles que les fiches descriptives des jeux de données, proposent une documentation plus complète, incluant des détails sur le processus de collecte des données, l'usage prévu et d'autres

informations contextuelles [59].

Voir aussi: point de données, données, caractéristique, espace des caractéristiques, espace des étiquettes, IA digne de confiance.

jeu de données local Le concept de jeu de données local se situe entre les notions de point de données et de jeu de données. Un jeu de données local est constitué de plusieurs points de données, chacun étant caractérisé par des caractéristiques et étiquettes. Contrairement à un jeu de données unique, utilisé dans les méthodes classiques d'apprentissage automatique, un jeu de données local peut être relié à d'autres jeux de données locaux par différentes formes de similarité. Ces similarités peuvent provenir de modèles probabilistes ou de l'infrastructure de communication, et sont représentées par les arêtes d'un réseau d'apprentissage fédéré.

Voir aussi: jeu de données, point de données, caractéristique, étiquette, apprentissage automatique, modèle probabiliste, réseau d'apprentissage fédéré.

loi (ou distribution) de probabilité Pour analyser les méthodes d'apprentissage automatique, il peut être utile d'interpréter les points de données comme des réalisations i.i.d., plural=i.i.d. d'une VA. Les attributs de ces points de données sont alors régis par la loi de probabilité de cette VA. La loi de probabilité d'une VA binaire $y \in \{0, 1\}$ est entièrement déterminée par les probabilités $\mathbb{P}(y = 0)$ et $\mathbb{P}(y = 1) = 1 - \mathbb{P}(y = 0)$. La loi de probabilité d'une VA à valeurs réelles $x \in \mathbb{R}$ peut être spécifiée par une fonction de densité de probabilité $p(x)$ telle que $\mathbb{P}(x \in [a, b]) \approx p(a)|b - a|$. Dans le cas le plus général, une loi de probabilité est définie par une mesure de

probabilité [6, 32].

Voir aussi: i.i.d., plural=i.i.d., réalisation, VA, probabilité, fonction de densité de probabilité.

loi des grands nombres La loi des grands nombres désigne la convergence de la moyenne d'un nombre croissant (et grand) de VA i.i.d., plural=i.i.d.s vers la moyenne de leur loi de probabilité commune. Il existe plusieurs versions de la loi des grands nombres selon les notions de convergence utilisées [34].

Voir aussi: i.i.d., plural=i.i.d., VA, moyenne, loi de probabilité.

loi normale multivariée La loi normale multivariée, notée $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, est un modèle probabiliste fondamental pour les vecteurs de caractéristiques numériques de dimension d fixe. Elle définit une famille de lois de probabilité sur des VA vectorielles $\mathbf{x} \in \mathbb{R}^d$ [7], [32], [60]. Chaque distribution de cette famille est entièrement spécifiée par son vecteur moyenne $\boldsymbol{\mu} \in \mathbb{R}^d$ et sa matrice de covariance $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times d}$. QUand la matrice de covariance $\boldsymbol{\Sigma}$ est inversible, la loi de probabilité correspondante est caractérisée par la fonction de densité de probabilité suivante:

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

IL faut noter que cette fonction de densité de probabilité n'est définie que si $\boldsymbol{\Sigma}$ est inversible. Plus généralement, toute VA $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ admet la représentation suivante:

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

où $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ est un vecteur normal centré réduit et $\mathbf{A} \in \mathbb{R}^{d \times d}$

vérifie $\mathbf{A}\mathbf{A}^\top = \mathbf{\Sigma}$. Cette représentation reste valable même lorsque $\mathbf{\Sigma}$ est singulière, auquel cas $\mathbf{A}$ n'est pas de plein rang [61, Ch. 23]. La famille des lois normales multivariées se distingue parmi les modèles probabilistes numériques pour au moins deux raisons. Premièrement, elle est stable par transformations affines, c'est-à-dire:

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

Deuxièmement, la loi de probabilité $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ maximise l'entropie différentielle parmi toutes les distributions ayant la même matrice de covariance $\mathbf{\Sigma}$ [30].

Voir aussi: modèle probabiliste, loi de probabilité, vecteur normal centré réduit, entropie différentielle, VA normale.

lot Dans le contexte de la SGD, un lot désigne un sous-ensemble choisi aléatoirement dans l'ensemble d'entraînement complet. On utilise les points de données de ce sous-ensemble pour estimer le gradient de l'erreur d'entrainement et, par la suite, mettre à jour les paramètres du modèle.

Voir aussi: SGD, ensemble d'entraînement, point de données, gradient, erreur d'entrainement, paramètres du modèle.

matrice Une matrice de taille $m \times d$ est un tableau bidimensionnel de nombres, noté

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

Ici, $A_{r,j}$ désigne l'entrée de la matrice située à la r-ième ligne et à la j-ième colonne. Les matrices servent à représenter divers objets mathématiques [62], notamment :

- Des systèmes d'équations linéaires: on peut utiliser une matrice pour représenter un système du type

  $$\begin{pmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \quad \text{de manière compacte par} \quad \mathbf{A}\mathbf{w} = \mathbf{y}.$$

  Un exemple important de système linéaire est la condition d'optimalité pour les paramètres du modèle en régression linéaire.

- Des applications linéaires: considérons un espace vectoriel $\mathcal{U}$ de dimension d et un espace vectoriel $\mathcal{V}$ de dimension m. En fixant une base $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(d)}$ de $\mathcal{U}$ et une base $\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(m)}$ de $\mathcal{V}$, toute matrice $\mathbf{A} \in \mathbb{R}^{m \times d}$ définit naturellement une application linéaire $\alpha : \mathcal{U} \to \mathcal{V}$ telle que

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

- Des jeux de données: une matrice peut représenter un jeu de données : chaque ligne correspond à un point de données et chaque colonne à une caractéristique ou une étiquette associée à ce point.

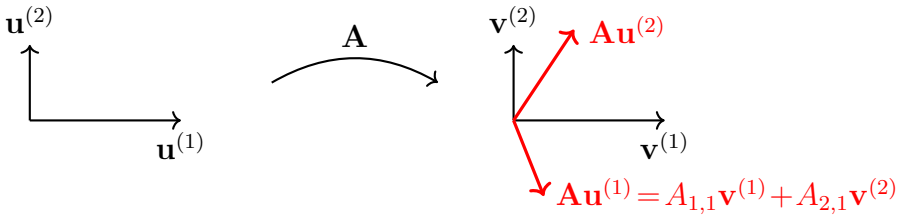


Fig. 21. Une matrice $\mathbf{A}$ définit une application linéaire entre deux espaces vectoriels.

Voir aussi: application linéaire, jeu de données, modèle linéaire.

matrice de caractéristiques Considérons un jeu de données $\mathcal{D}$ avec m points de données de vecteurs de caractéristiques $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)} \in \mathbb{R}^d$. Il est pratique de rassembler les vecteurs de caractéristiques individuels dans une matrice de caractéristiques $\mathbf{X} := \left(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}\right)^T$ de taille $m \times d$.

Voir aussi: jeu de données, point de données, vecteur de caractéristiques, caractéristique.

matrice de covariance La matrice de covariance d'une VA $\mathbf{x} \in \mathbb{R}^d$ est définie comme $\mathbb{E}\Big\{\big(\mathbf{x} - \mathbb{E}\{\mathbf{x}\}\big)\big(\mathbf{x} - \mathbb{E}\{\mathbf{x}\}\big)^T\Big\}$.

Voir aussi: VA, covariance.

matrice inverse On définit la matrice inverse $\mathbf{A}^{-1}$ d'une matrice carrée $\mathbf{A} \in \mathbb{R}^{n \times n}$ de rang maximal, c'est-à-dire dont les colonnes sont linéairement indépendantes. Dans ce cas, on dit que $\mathbf{A}$ est inversible, et son inverse satisfait:

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

Une matrice carrée est inversible si et seulement si son déterminant est non nul. Les matrices inverses sont fondamentales pour la résolution de systèmes d'équations linéaires et dans la solution explicite de la régression linéaire [26], [63]. Le concept de matrice inverse peut être étendu aux matrices non carrées ou de rang non maximal. On peut définir une « inverse à gauche » $\mathbf{B}$ telle que $\mathbf{BA} = \mathbf{I}$, ou une « inverse à droite » $\mathbf{C}$ telle que $\mathbf{AC} = \mathbf{I}$. Pour les matrices rectangulaires ou singulières, la pseudo-inverse de Moore–Penrose, notée $\mathbf{A}^+$, fournit une généralisation unifiée de la matrice inverse [3].

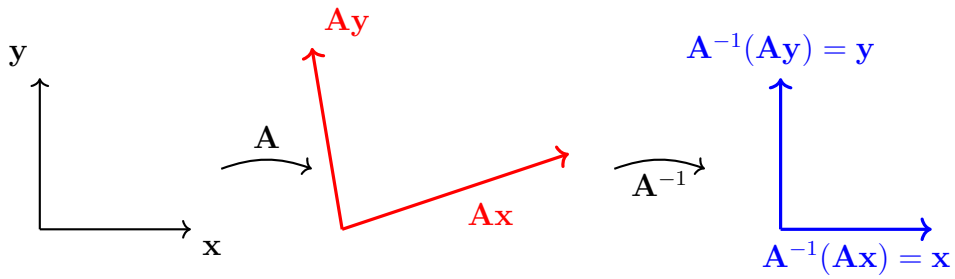


Fig. 22. Une matrice $\mathbf{A}$ représente une transformation linéaire de $\mathbb{R}^2$. La matrice inverse $\mathbf{A}^{-1}$ représente la transformation inverse.

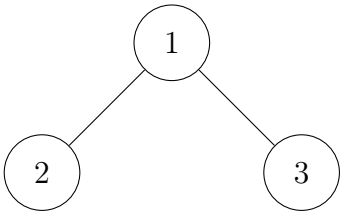
Voir aussi: déterminant, régression linéaire, pseudo-inverse.

matrice laplacienne La structure d'un graphe $\mathcal{G}$, avec pour sommets $i = 1, \ldots, n$, peut être analysée à l'aide des propriétés de matrices spéciales associées à $\mathcal{G}$. L'une de ces matrices est la matrice laplacienne de $\mathcal{G}$: $\mathbf{L}^{(\mathcal{G})} \in \mathbb{R}^{n \times n}$, définie pour un graphe $\mathcal{G}$ non orienté et pondéré [64, 65].

Elle est définie terme à terme par (voir Figure 23)

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

Ici, $A_{i,i'}$ désigne le poids d'une arête $\{i,i'\} \in \mathcal{E}$.



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

Fig. 23. À gauche: Un graphe non orienté $\mathcal{G}$ avec trois sommets $i = 1, 2, 3$. À droite: La matrice laplacienne $\mathbf{L}^{(\mathcal{G})} \in \mathbb{R}^{3\times 3}$ de $\mathcal{G}$.

Voir aussi: graphe, poids d'arête.

maximum Le maximum d'un ensemble $\mathcal{A} \subseteq \mathbb{R}$ de nombres réels est le plus grand élément de cet ensemble, si un tel élément existe. Un ensemble $\mathcal{A}$ a un maximum s'il est majoré et atteint sa borne supérieure [2, Sec. 1.4]. Voir aussi: borne supérieure.

maximum de vraisemblance Considérons des points de données $\mathcal{D} = \{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}\}$ que l'on interprète comme les réalisations de VA i.i.d., plural=i.i.d.s avec une loi de probabilité commune $\mathbb{P}(\mathbf{z}; \mathbf{w})$ qui dépend des paramètres du modèle $\mathbf{w} \in \mathcal{W} \subseteq \mathbb{R}^n$. Les méthodes de maximum de vraisemblance apprennent les paramètres du modèle $\mathbf{w}$ en maximisant

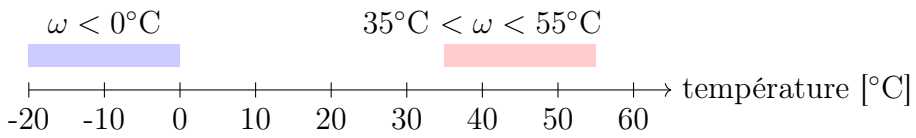


Fig. 24. Un espace échantillon constitué de toutes les valeurs de température possibles ω pouvant être mesurées dans une station FMI. Deux sous-ensembles mesurables de températures, notés $\mathcal{A}^{(1)}$ et $\mathcal{A}^{(2)}$, sont mis en évidence. Pour toute valeur réelle de température ω, il est possible de déterminer si $\omega \in \mathcal{A}^{(1)}$ et si $\omega \in \mathcal{A}^{(2)}$.

la probabilité $\mathbb{P}(\mathcal{D}; \mathbf{w}) = \prod_{r=1}^{m} \mathbb{P}(\mathbf{z}^{(r)}; \mathbf{w})$ du jeu de données observé. Ainsi, l'estimateur du maximum de vraisemblance est une solution au problème d'optimisation $\max_{\mathbf{w} \in \mathcal{W}} \mathbb{P}(\mathcal{D}; \mathbf{w})$.

Voir aussi: loi de probabilité, problème d'optimisation, modèle probabiliste.

mesurable Considérons une expérience aléatoire, comme l'enregistrement de la température de l'air dans une station météo FMI. L'espace échantillon correspondant, noté Ω, est l'ensemble de tous les résultats possibles ω (par exemple, toutes les valeurs possibles de température en degrés Celsius). Dans de nombreuses applications d'apprentissage automatique, on ne s'intéresse pas au résultat exact ω, mais seulement à savoir s'il appartient à un sous-ensemble $\mathcal{A} \subseteq \Omega$ (par exemple, « la température est-elle inférieure à zéro degré ? »). On dit qu'un tel sous-ensemble $\mathcal{A}$ est mesurable s'il est possible de déterminer, pour tout résultat ω, si $\omega \in \mathcal{A}$ ou non. En principe, les ensembles mesurables pourraient être choisis

librement (par exemple, en fonction de la résolution de l'appareil de mesure). Cependant, il est souvent utile d'imposer certaines conditions de complétude à la collection des ensembles mesurables. Par exemple, l'espace échantillon lui-même doit être mesurable, et l'union de deux ensembles mesurables doit aussi être mesurable. Ces conditions de complétude peuvent être formalisées via la notion de tribu [6, 66, 67]. Un espace mesurable est un couple $\left(\mathcal{X}, \mathcal{F}\right)$ constitué d'un ensemble quelconque $\mathcal{X}$ et d'une collection $\mathcal{F}$ de sous-ensembles mesurables de $\mathcal{X}$ qui forment une tribu.

Voir aussi : probabilité, espace échantillon.

minimisation de la variation totale généralisée (GTVMin) GTVMin est une instance de minimisation du risque empirique régularisé (MRER) utilisant la variation totale généralisée (GTV) des paramètres locaux des modèles comme terme de régularisation [68].

Voir aussi : MRER, GTV, terme de régularisation, paramètres du modèle.

minimisation du risque empirique (MRE) La minimisation du risque empirique (MRE) est le problème d'optimisation qui consiste à trouver une hypothèse (dans un modèle) qui minimise la perte moyenne (ou risque empirique) sur un jeu de données $\mathcal{D}$ donné (c'est-à-dire, l'ensemble d'entraînement). De nombreuses méthodes d'apprentissage automatique sont obtenues à partir du risque empirique via des choix de conception spécifiques pour le jeu de données, le modèle et la perte [8, Ch. 3].

Voir aussi: problème d'optimisation, hypothèse, modèle, minimum,

perte, risque empirique, jeu de données, ensemble d'entraînement, apprentissage automatique.

minimisation du risque empirique régularisé (MRER) La MRE classique consiste à apprendre une hypothèse (ou à entraîner un modèle) $h \in \mathcal{H}$ en se basant uniquement sur le risque empirique $\widehat{L}\big(h|\mathcal{D}\big)$ calculé sur un ensemble d'entraînement $\mathcal{D}$. Pour rendre la MRE moins sujette au surapprentissage, on peut appliquer une régularisation en ajoutant un terme de régularisation (pondéré) $\mathcal{R}\big\{h\big\}$ dans l'objectif d'apprentissage. Cela conduit à la minimisation du risque empirique régularisé (MRER) :

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

Le paramètre $\alpha \geq 0$ contrôle l'intensité de la régularisation. Pour $\alpha = 0$, on retrouve la MRE standard, sans régularisation. Lorsque α augmente, l'hypothèse apprise est de plus en plus biaisée vers des petites valeurs de $\mathcal{R}\big\{h\big\}$. Le terme $\alpha\mathcal{R}\big\{h\big\}$ dans la fonction objective de (5) peut être interprété comme une estimation de l'augmentation moyenne de la perte qui peut se produire lors de la prédiction d'étiquettes pour des points de données en dehors de l'ensemble d'entraînement. Cette intuition peut être formalisée de plusieurs manières. Par exemple, dans le cas d'un modèle linéaire entraîné avec la perte quadratique et le terme de régularisation $\mathcal{R}\big\{h\big\} = \|\mathbf{w}\|_2^2$, le terme $\alpha\mathcal{R}\big\{h\big\}$ correspond à l'augmentation attendue de la perte induite par l'ajout de VA normales aux vecteurs de caractéristiques de l'ensemble d'entraînement [8, Ch. 3]. Une construction rigoureuse du terme de régularisation $\mathcal{R}\big\{h\big\}$ découle des majorations approchées de l'erreur de généralisation. L'instance

de MRER ainsi obtenue est appelée minimisation du risque structurel (MRS) [69, Sec. 7.2].

Voir aussi : MRE, hypothèse, modèle, risque empirique, ensemble d'entraînement, surapprentissage, régularisation, terme de régularisation, paramètre, fonction objective, perte, étiquette, point de données, modèle linéaire, perte quadratique, VA normale, vecteur de caractéristiques, généralisation, MRS.

minimisation du risque structurel (MRS) La MRS est une instance de MRER, dans laquelle le modèle $\mathcal{H}$ peut être exprimé comme une union dénombrable de sous-modèles telle que $\mathcal{H} = \bigcup_{n=1}^{\infty} \mathcal{H}^{(n)}$. Chaque sous-modèle $\mathcal{H}^{(n)}$ permet d'évaluer une borne supérieure approchée de l'erreur de généralisation encourue lors de l'application de la MRE pour entraîner $\mathcal{H}^{(n)}$. Ces bornes individuelles — une pour chaque sous-modèle — sont ensuite combinées pour former un terme de régularisation utilisé dans l'objectif de la MRER. Ces bornes supérieures approchées (une pour chaque $\mathcal{H}^{(n)}$) sont alors combinées pour construire un terme de régularisation pour la MRER [46, Sec. 7.2].

Voir aussi: MRER, modèle, généralisation, MRE, terme de régularisation, risque.

minimum Étant donné un ensemble de nombres réels, le minimum est le plus petit de ces nombres. Notons que pour certains ensembles, comme l'ensemble des nombres réels négatifs, le minimum n'existe pas.

modèle Dans le contexte de l'apprentissage automatique, le terme « modèle » désigne typiquement l'espace des hypothèses sous-jacent à une méthode

d'apprentissage automatique [8], [46]. Cependant, ce terme est également utilisé dans d'autres domaines avec des significations différentes. Par exemple, un modèle probabiliste désigne un ensemble paramétré de lois de probabilité.

Voir aussi: apprentissage automatique, espace des hypothèses, modèle probabiliste, loi de probabilité.

modèle linéaire Considérons une application d'apprentissage automatique impliquant des points de données, chacun représenté par un vecteur de caractéristiques numérique $\mathbf{x} \in \mathbb{R}^d$. Un modèle linéaire définit un espace des hypothèses constitué de toutes les applications linéaires réelles de $\mathbb{R}^d$ vers $\mathbb{R}$ telles que

$$\mathcal{H}^{(d)} := \left\{ h : \mathbb{R}^d \to \mathbb{R} \mid h(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} \text{ pour un certain } \mathbf{w} \in \mathbb{R}^d \right\}. \quad (6)$$

Chaque valeur de d définit un espace des hypothèses différent, correspondant au nombre de caractéristiques utilisées pour calculer la prédiction $h(\mathbf{x})$. Le choix de d est souvent guidé non seulement par des considérations sur les aspects computationnels (par exemple, moins de variables réduisent le calcul) et sur les aspects statistiques (par exemple, plus de variables réduisent généralement le biais et le risque), mais aussi par l'interprétabilité. Un modèle linéaire utilisant un petit nombre de caractéristiques bien choisies est généralement considéré comme plus interprétable [70], [44]. Le modèle linéaire est attractif car il peut généralement être entraîné à l'aide de méthodes d'optimisation convexes extensives [71], [72]. En outre, les modèles linéaires permettent souvent une analyse statistique rigoureuse, y compris des limites fondamentales

sur le risque minimal atteignable [55]. Ils sont aussi utiles pour analyser des modèles plus complexes et non linéaires comme les RNA. Par exemple, un réseau de neurones profond peut être vu comme la composition d'une transformation de caractéristiques—mis en œuvre par les couches d'entrée et cachées—et d'un modèle linéaire dans la couche de sortie. De même, un arbre de décision peut être interprété comme appliquant un transformation de caractéristiques encodée en one-hot basée sur des régions de décision, suivie d'un modèle linéaire assignant une prédiction à chaque région. Plus généralement, tout modèle entraîné $\hat{h} \in \mathcal{H}$ qui est dérivable en un certain $\mathbf{x}'$ peut être approximé localement par une application linéaire $g(\mathbf{x})$. La Figure 25 illustre une telle approximation linéaire locale, définie par le gradient $\nabla\hat{h}(\mathbf{x}')$. Remarquons que le gradient n'est défini que là où $\hat{h}$ est dérivable. Pour garantir la robustesse dans le contexte d'IA digne de confiance, on peut préférer des modèles dont l'application associée $\hat{h}$ est lipschitzienne. Un résultat classique de l'analyse mathématique—le théorème de Rademacher—affirme que si $\hat{h}$ est L-lipschitzienne sur un ouvert $\Omega \subseteq \mathbb{R}^d$, alors $\hat{h}$ est dérivable presque partout sur Ω [73, Th. 3.1].

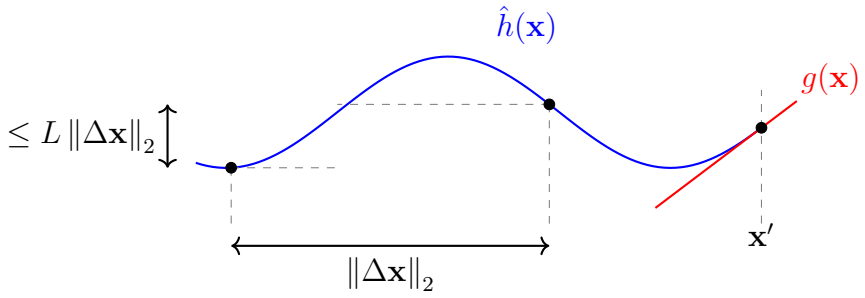


Fig. 25. Un modèle entraîné $\hat{h}(\mathbf{x})$ qui est dérivable en un point $\mathbf{x}'$ peut être approximé localement par une application linéaire $g \in \mathcal{H}^{(d)}$. Cette approximation locale est déterminée par le gradient $\nabla\hat{h}(\mathbf{x}')$.

Voir aussi: modèle, espace des hypothèses, application linéaire, interprétabilité, LIME.

modèle local Considérons une collection d'appareils représentés par les sommets $\mathcal{V}$ d'un réseau d'apprentissage fédéré. Un modèle local $\mathcal{H}^{(i)}$ est un espace des hypothèses attribué à un sommet $i \in \mathcal{V}$. Des sommets différents peuvent avoir des espaces des hypothèses différents, c'est-à-dire qu'en général $\mathcal{H}^{(i)} \neq \mathcal{H}^{(i')}$ pour des sommets $i, i' \in \mathcal{V}$ distincts.

Voir aussi : appareil, réseau d'apprentissage fédéré, modèle, espace des hypothèses.

modèle probabiliste Un modèle probabiliste interprète les points de données comme des réalisations de VA selon une loi de probabilité conjointe. Cette loi de probabilité conjointe implique généralement des paramètre qui doivent être choisis manuellement ou appris via des méthodes d'inférence statistique telles que l'estimation par maximum de vraisemblance [37].

Voir aussi: modèle, point de données, réalisation, VA, loi de probabilité, paramètre, maximum de vraisemblance.

modèle à blocs stochastiques (SBM) Le SBM est un modèle génératif probabiliste pour un graphe non orienté $\mathcal{G} = \big(\mathcal{V}, \mathcal{E}\big)$ défini sur un ensemble de sommets $\mathcal{V}$ [74]. Dans sa variante la plus simple, le SBM génère un graphe en assignant aléatoirement à chaque sommet $i \in \mathcal{V}$ un indice de groupe $c_i \in \{1, \ldots, k\}$. Une paire de sommets distincts du graphe est reliée par une arête avec une probabilité $p_{i,i'}$ qui dépend uniquement des étiquettes $c_i, c_{i'}$. La présence d'arêtes entre différentes paires de sommets est statistiquement indépendante.

Voir aussi : modèle, graphe, groupe, probabilité, étiquette.

moindre contraction absolue et opérateur de sélection (Lasso) Le Lasso est une instance de MRS. Il apprend les poids $\mathbf{w}$ d'une application linéaire $h(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ à partir d'un ensemble d'entraînement. Le Lasso est obtenu à partir de régression linéaire en ajoutant la norme ℓ_1 pondérée $\alpha \left\|\mathbf{w}\right\|_1$ à la moyenne de la perte quadratique subie sur le ensemble d'entraînement.

Voir aussi : MRS, poids, application linéaire, ensemble d'entraînement, régression linéaire, norme, perte quadratique.

moyenne La moyenne d'une VA $\mathbf{x}$, à valeurs dans un espace euclidien $\mathbb{R}^d$, est son espérance $\mathbb{E}\{\mathbf{x}\}$. Elle est définie comme l'intégrale de Lebesgue de $\mathbf{x}$ par rapport à la loi de probabilité sous-jacente P (par exemple, voir [2] ou [6]), c'est-à-dire

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

Il est utile de considérer la moyenne comme la solution du problème de minimisation du risque suivant [7]:

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

On utilise aussi ce terme pour désigner la moyenne d'une séquence finie $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)} \in \mathbb{R}^d$. Cependant, ces deux définitions sont essentiellement équivalentes. En effet, on peut utiliser la séquence $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)} \in \mathbb{R}^d$ pour construire un VA discret $\widetilde{\mathbf{x}} = \mathbf{x}^{(I)}$, où l'indice I est choisi uniformément au hasard dans l'ensemble $\{1, \ldots, m\}$. La moyenne de $\widetilde{\mathbf{x}}$ est précisément la moyenne $(1/m)\sum_{r=1}^{m} \mathbf{x}^{(r)}$.

Voir aussi: VA, espérance, loi de probabilité.

méthode d'optimisation Une méthode d'optimisation est un algorithme qui prend en entrée une représentation d'un problème d'optimisation et fournit en sortie une solution (approchée) [21], [72], [75].

Voir aussi : algorithme, problème d'optimisation.

méthode du point fixe La méthode du point fixe est une méthode itérative permettant de résoudre un problème d'optimisation. Elle construit une suite $\mathbf{w}^{(0)}, \mathbf{w}^{(1)}, \ldots$ en appliquant de manière répétée un opérateur $\mathcal{F}$, c'est-à-dire:

$$\mathbf{w}^{(k+1)} = \mathcal{F}\mathbf{w}^{(k)}, \quad \text{pour } k = 0, 1, \ldots. \tag{7}$$

L'opérateur $\mathcal{F}$ est choisi de sorte que tout point fixe soit une solution $\widehat{\mathbf{w}}$ du problème d'optimisation donné. Par exemple, étant donnée une fonction $f(\mathbf{w})$ convexe et entropie différentielle, les points fixes de l'opérateur $\mathcal{F} : \mathbf{w} \mapsto \mathbf{w} - \nabla f(\mathbf{w})$ coïncident avec les minimiseurs de

$f(\mathbf{w})$. De manière générale, pour un problème d'optimisation donné dont la solution est $\widehat{\mathbf{w}}$, il peut exister plusieurs opérateurs $\mathcal{F}$ ayant pour points fixes $\widehat{\mathbf{w}}$. Il est donc clairement souhaitable d'utiliser un opérateur $\mathcal{F}$ tel que l'itération (7) réduise la distance à la solution :

$$\underbrace{\left\|\mathbf{w}^{(k+1)} - \widehat{\mathbf{w}}\right\|_2}_{\overset{(7)}{=}\left\|\mathcal{F}\mathbf{w}^{(k)} - \mathcal{F}\widehat{\mathbf{w}}\right\|_2} \leq \left\|\mathbf{w}^{(k)} - \widehat{\mathbf{w}}\right\|_2 .$$

On exige donc que $\mathcal{F}$ soit au minimum non-expansif, c'est-à-dire que l'itération (7) ne produise pas des paramètres du modèle plus éloignés de la solution $\widehat{\mathbf{w}}$. Mieux encore, chaque itération (7) devrait faire progresser la solution, en réduisant effectivement la distance à $\widehat{\mathbf{w}}$. Cette exigence peut être formulée précisément à l'aide de la notion d'opérateur contractant [76], [77]. On dit que $\mathcal{F}$ est un opérateur contractant s'il existe un facteur $\kappa \in [0, 1)$ tel que

$$\left\|\mathcal{F}\mathbf{w} - \mathcal{F}\mathbf{w}'\right\|_2 \leq \kappa \left\|\mathbf{w} - \mathbf{w}'\right\|_2 \quad \text{pour tout } \mathbf{w}, \mathbf{w}'.$$

Avec un opérateur contractant $\mathcal{F}$, l'itération (7) génère une suite $\mathbf{w}^{(k)}$ qui converge rapidement. En particulier [2, Th. 9.23], on a:

$$\left\|\mathbf{w}^{(k)} - \widehat{\mathbf{w}}\right\|_2 \leq \kappa^k \left\|\mathbf{w}^{(0)} - \widehat{\mathbf{w}}\right\|_2 .$$

Ici, $\left\|\mathbf{w}^{(0)} - \widehat{\mathbf{w}}\right\|_2$ est la distance entre l'initialisation et la solution. Il s'avère que la méthode du point fixe (7) utilisant un opérateur ferme non-expansif $\mathcal{F}$ est garantie de converger vers un point fixe de $\mathcal{F}$ [76, Cor. 5.16]. La Fig. 26 montre des exemples d'un opérateur ferme non-expansif, d'un opérateur non-expansif, et d'un opérateur contractant, tous définis

sur $\mathbb{R}$. Un autre exemple d'opérateur ferme non-expansif est l'opérateur proximal d'une fonction convexe [76], [78].

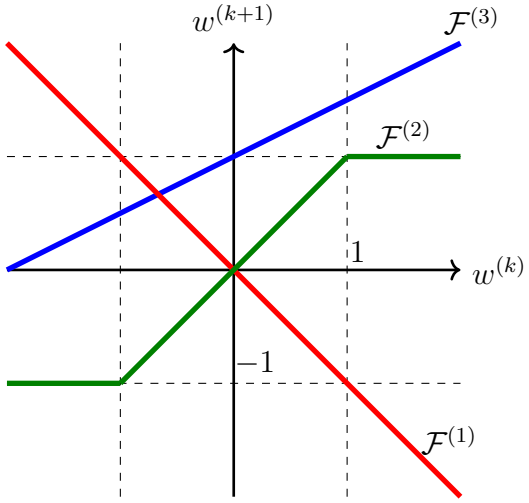


Fig. 26. Exemple d'opérateur non-expansif $\mathcal{F}^{(1)}$, opérateur ferme non-expansif $\mathcal{F}^{(2)}$, et opérateur contractant $\mathcal{F}^{(3)}$.

Voir aussi: problème d'optimisation, dérivable, convexe fonction, paramètres du modèle, opérateur contractant, opérateur proximal.

méthode à noyau Une méthode à noyau est une méthode d'apprentissage automatique qui utilise un noyau K pour transformer le vecteur de caractéristiques initial (brut) $\mathbf{x}$ d'un point de données en un nouveau (transformé) vecteur de caractéristiques $\mathbf{z} = K\big(\mathbf{x}, \cdot\big)$ [79, 80]. La motivation derrière cette transformation est que, grâce à un noyau approprié, les points de données possèdent une géométrie « plus favorable » dans l'espace des caractéristiques transformé. Par exemple, dans un problème de classification binaire, l'utilisation des vecteurs de caractéristiques

transformés $\mathbf{z}$ peut permettre d'appliquer des modèles linéaires, même si les points de données ne sont pas linéairement séparables dans l'espace des caractéristiques initial (voir Figure 27).

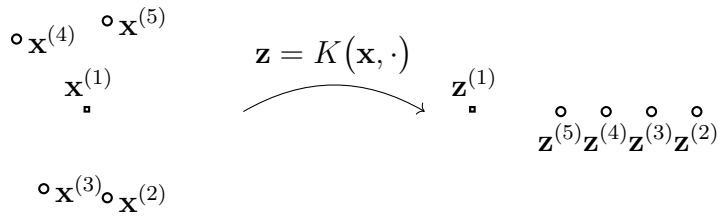


Fig. 27. Cinq points de données caractérisés par des vecteurs de caractéristiques $\mathbf{x}^{(r)}$ et étiquettes $y^{(r)} \in \{\circ, \square\}$, pour $r = 1, \ldots, 5$. Avec ces vecteurs de caractéristiques, il n'est pas possible de séparer les deux classes par une ligne droite (représentant la frontière de décision d'un classifieur linéaire). En revanche, le vecteurs de caractéristiques transformé $\mathbf{z}^{(r)} = K(\mathbf{x}^{(r)}, \cdot)$ permet de séparer les points de données à l'aide d'un classifieur linéaire.

Voir aussi: noyau, vecteur de caractéristiques, espace des caractéristiques, classifieur linéaire.

méthodes basées sur le gradient Les méthodes basées sur le gradient sont des techniques itératives pour trouver le minimum (ou le maximum) d'une fonction objective des paramètres du modèle dérivable. Ces méthodes construisent une suite d'approximations d'un choix optimal des paramètres du modèle qui aboutit à une valeur minimum (ou maximum) de la fonction objective. Comme leur nom l'indique, les méthodes basées sur le gradient utilisent les gradients de la fonction objective

évalués lors des itérations précédentes pour construire de nouveaux paramètres du modèle (espérons-le) améliorés. Un exemple important d'une méthode basée sur le gradient est la descente de gradient.
Voir aussi: gradient, minimum, maximum, dérivable, fonction objective, paramètres du modèle, descente de gradient.

niveau de référence Considérons une méthode d'apprentissage automatique qui produit une hypothèse apprise (ou un modèle entraîné) $\hat{h} \in \mathcal{H}$. On évalue la qualité d'un modèle entraîné en calculant la perte moyenne sur un ensemble de test. Mais comment pouvons-nous évaluer si la performance obtenue sur l'ensemble de test est suffisamment bonne ? Comment déterminer si le modèle entraîné est proche de l'optimal et qu'il est peu utile d'investir davantage de ressources (pour la collecte de données ou le calcul) pour l'améliorer ? À cette fin, il est utile d'avoir un niveau de référence avec lequel comparer la performance du modèle entraîné. Cette référence peut être obtenue à partir de performances humaines, par exemple le taux de mauvaise classification de diagnostic du cancer par inspection visuelle de la peau par des dermatologues [81]. Une autre source pour une référence est une méthode d'apprentissage automatique existante, mais pour une raison quelconque inadaptée. Par exemple, la méthode d'apprentissage automatique déjà existante peut être trop coûteuse en calcul pour l'application visée. Néanmoins, son erreur sur l'ensemble de test peut toujours servir de référence. Une autre approche, un peu plus rigoureuse, pour construire une référence est via un modèle probabiliste. Dans de nombreux cas, étant donné un

modèle probabiliste $p(\mathbf{x}, y)$, on peut déterminer précisément le risque minimal atteignable parmi toutes les hypothèses (même sans appartenir à l'espace des hypothèses $\mathcal{H}$) [37]. Ce risque minimal atteignable (appelé risque bayésien) est le risque de l'estimateur bayésien pour l'étiquette y d'un point de données, étant données ses caractéristiques $\mathbf{x}$. Notons que, pour un choix fixé de fonction de perte, l'estimateur bayésien (s'il existe) est complètement déterminé par la loi de probabilité $p(\mathbf{x}, y)$ [37, Ch. 4]. Cependant, calculer l'estimateur bayésien et le risque bayésien présente deux défis principaux:

1) La loi de probabilité $p(\mathbf{x}, y)$ est inconnue et doit être estimée.
2) Même si $p(\mathbf{x}, y)$ est connue, le calcul exact du risque bayésien peut être trop coûteux [82].

Un modèle probabiliste largement utilisé est la loi normale multivariée $(\mathbf{x}, y) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ pour des points de données caractérisés par des caractéristiques et des étiquettes numériques. Ici, pour la perte quadratique, l'estimateur bayésien est donné par la moyenne a posteriori $\mu_{y|\mathbf{x}}$ de l'étiquette y, étant données les caractéristiques $\mathbf{x}$ [32, 37]. Le risque bayésien correspondant est donné par la variance a posteriori $\sigma^2_{y|\mathbf{x}}$ (voir Figure 28).

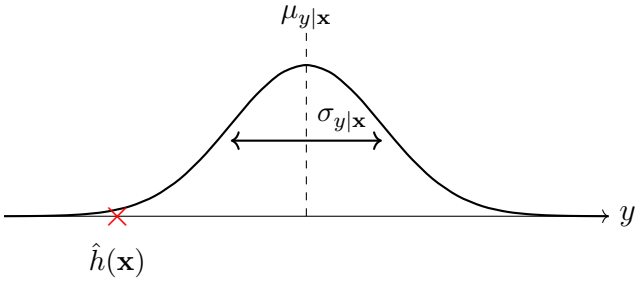


Fig. 28. Si les caractéristiques et l'étiquette d'un point de données suivent une loi normale multivariée, on peut atteindre le risque minimal (sous perte quadratique) en utilisant l'estimateur bayésien $\mu_{y|\mathbf{x}}$ pour prédire l'étiquette y d'un point de données avec des caractéristiques $\mathbf{x}$. Le risque minimal correspondant est donné par la variance a posteriori $\sigma^2_{y|\mathbf{x}}$. On peut utiliser cette quantité comme référence pour la perte moyenne d'un modèle entraîné $\hat{h}$.

Voir aussi: risque bayésien, estimateur bayésien.

non régulière (ou non lisse) On qualifie une fonction de non régulière si elle n'est pas régulière [75].

Voir aussi: fonction, régulière.

norme Une norme est une fonction qui associe à chaque élément (vecteur) d'un espace vectoriel un réel positif ou nul. Cette fonction doit être homogène, définie positive, et satisfaire l'inégalité triangulaire [27].

Voir aussi: fonction, espace vectoriel, vecteur.

noyau (fonction) Considérons un ensemble de points de données, chacun représenté par un vecteur de caractéristiques $\mathbf{x} \in \mathcal{X}$, où $\mathcal{X}$ désigne

l'espace des caractéristiques. Une fonction noyau, ou noyau, (à valeurs réelles) est une fonction $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ qui associe à chaque paire de vecteurs de caractéristiques $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ un réel $K\big(\mathbf{x}, \mathbf{x}'\big)$. Cette valeur est généralement interprétée comme une mesure de similarité entre $\mathbf{x}$ et $\mathbf{x}'$. La propriété caractéristique d'un noyau est qu'il est symétrique, c'est-à-dire, $K\big(\mathbf{x}, \mathbf{x}'\big) = K\big(\mathbf{x}', \mathbf{x}\big)$, et que, pour tout ensemble fini de vecteurs de caractéristiques $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$, la matrice

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

est semi-définie positive. Un noyau définit naturellement une transformation d'un vecteur de caractéristiques $\mathbf{x}$ en une fonction $\mathbf{z} = K\big(\mathbf{x}, \cdot\big)$. Cette fonction $\mathbf{z}$ associe à une entrée $\mathbf{x}' \in \mathcal{X}$ la valeur $K\big(\mathbf{x}, \mathbf{x}'\big)$. On peut considérer la fonction $\mathbf{z}$ comme un nouveau vecteur de caractéristiques appartenant à un espace des caractéristiques $\mathcal{X}'$ qui est typiquement différent de $\mathcal{X}$. Ce nouvel espace des caractéristiques $\mathcal{X}'$ possède une structure mathématique particulière, à savoir, c'est un espace de Hilbert à noyau reproduisant (RKHS) [80], [79]. Puisque $\mathbf{z}$ appartient à une RKHS, qui est un espace vectoriel, on peut l'interpréter comme un vecteur de caractéristiques généralisé. À noter qu'un vecteur de caractéristiques de longueur finie $\mathbf{x} = \big(x_1, \ldots, x_d\big)^T \in \mathbb{R}^d$ peut être vu comme une fonction $\mathbf{x} : \{1, \ldots, d\} \to \mathbb{R}$ associant une valeur réelle à chaque indice $j \in \{1, \ldots, d\}$.

Voir aussi : vecteur de caractéristiques, espace des caractéristiques,

espace de Hilbert, méthode à noyau.

noyau (matrice) Le noyau d'une matrice $\mathbf{A} \in \mathbb{R}^{d' \times d}$, noté null($\mathbf{A}$), est l'ensemble de tous les vecteurs $\mathbf{n} \in \mathbb{R}^d$ tels que

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

Considérons une méthode d'apprentissage de caractéristiques qui utilise la matrice $\mathbf{A}$ pour transformer un vecteur de caractéristiques $\mathbf{x} \in \mathbb{R}^d$ d'un point de données en un nouveau vecteur de caractéristiques $\mathbf{z} = \mathbf{Ax} \in \mathbb{R}^{d'}$. Le noyau null($\mathbf{A}$) caractérise toutes les directions de l'espace des caractéristiques original $\mathbb{R}^d$ dans lesquelles la transformation $\mathbf{Ax}$ reste inchangée. En d'autres termes, ajouter un vecteur du noyau à un vecteur de caractéristiques $\mathbf{x}$ n'affecte pas la représentation transformée $\mathbf{z}$. Cette propriété peut être exploitée pour imposer des invariances dans les prédictions (calculées à partir de $\mathbf{Ax}$). La figure 29 illustre une telle invariance. Elle montre des versions tournées de deux chiffres manuscrits, qui s'alignent approximativement le long de courbes unidimensionnelles dans le espace des caractéristiques original. Ces courbes sont parallèles à une direction vecteur $\mathbf{n} \in \mathbb{R}^d$. Pour garantir que le modèle entraîné soit invariant à de telles rotations, on peut choisir la matrice de transformation $\mathbf{A}$ de façon que $\mathbf{n} \in$ null($\mathbf{A}$). Cela garantit que $\mathbf{Ax}$ — et donc la prédiction résultante — soit approximativement insensible à la rotation de l'image d'entrée. Voir aussi: matrice.

Démo Python : cliquer ici

nuage de points Une technique de visualisation qui représente des points de données par des marqueurs dans un plan bidimensionnel. La Fig. 30

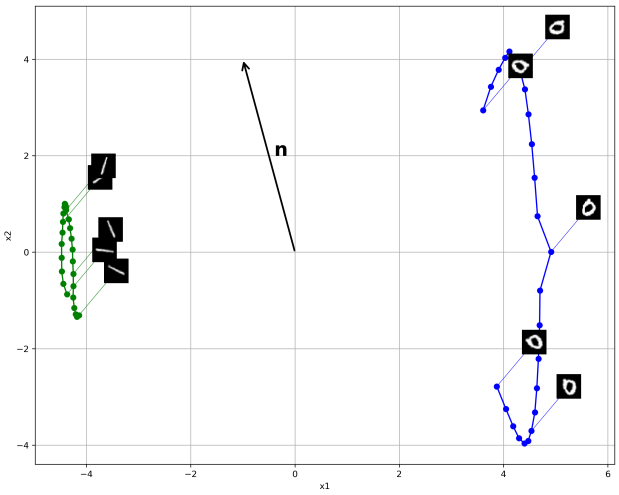


Fig. 29. Images tournées de deux chiffres manuscrits. Les rotations s'alignent approximativement le long de courbes linéaires parallèles au vecteur **n**.

montre un exemple de nuage de points.

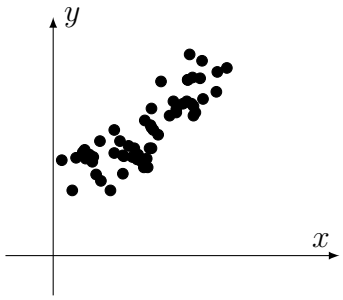


Fig. 30. Un nuage de points avec des marqueurs cercles, où les points de données représentent les conditions météorologiques quotidiennes en Finlande. Chaque point de données est caractérisé par sa température minimale diurne x comme caractéristique et sa température maximale diurne y comme étiquette. Les températures ont été mesurées à la station météo FMI Helsinki Kaisaniemi durant la période du 01.09.2024 au 28.10.2024.

Un nuage de points permet une inspection visuelle des points de données naturellement représentés par des vecteurs de caractéristiques dans des espaces de grande dimension.

Voir aussi: point de données, minimum, caractéristique, maximum, étiquette, FMI, vecteur de caractéristiques, réduction de dimension.

opérateur contractant Un opérateur $\mathcal{F} : \mathbb{R}^d \to \mathbb{R}^d$ est une contraction si, pour un certain $\kappa \in [0, 1[$,

$$\|\mathcal{F}\mathbf{w} - \mathcal{F}\mathbf{w}'\|_2 \leq \kappa \|\mathbf{w} - \mathbf{w}'\|_2 \text{ est vérifiée pour tous } \mathbf{w}, \mathbf{w}' \in \mathbb{R}^d.$$

opérateur proximal Étant donné une fonction convexe $f(\mathbf{w}')$, on définit son opérateur proximal comme suit [76, 78]:

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

Comme illustré à la Figure 31, évaluer l'opérateur proximal revient à minimiser une version pénalisée de $f(\mathbf{w}')$. Le terme de pénalité est la distance euclidienne quadratique pondérée à un vecteur donné $\mathbf{w}$. L'opérateur proximal peut être interprété comme une généralisation du pas de gradient, défini pour une fonction régulière et convexe $f(\mathbf{w}')$. En effet, effectuer un pas avec une taille de pas η à partir du vecteur actuel $\mathbf{w}$ revient à appliquer l'opérateur proximal à la fonction linéarisée $\tilde{f}(\mathbf{w}') = (\nabla f(\mathbf{w}))^T(\mathbf{w}' - \mathbf{w})$, avec $\rho = 1/\eta$.

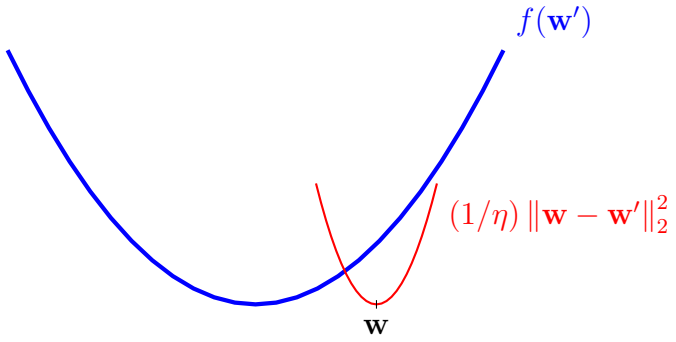


Fig. 31. Un pas généralisé met à jour un vecteur $\mathbf{w}$ en minimisant une version pénalisée de la fonction $f(\cdot)$. Le terme de pénalité correspond à la distance euclidienne quadratique pondérée entre la variable d'optimisation $\mathbf{w}'$ et le vecteur donné $\mathbf{w}$.

Voir aussi: convexe, fonction, généralisation, pas, régulière, taille de pas.

paramètre Les paramètres d'un modèle en apprentissage automatique sont des quantités ajustables (c'est-à-dire apprenables ou modifiables) qui permettent de choisir parmi différentes fonctions hypothèse. Par exemple, le modèle linéaire $\mathcal{H} := \{h^{(\mathbf{w})} : h^{(\mathbf{w})}(x) = w_1 x + w_2\}$ correspond à l'ensemble des fonctions hypothèse $h^{(\mathbf{w})}(x) = w_1 x + w_2$ avec un choix particulier des paramètres $\mathbf{w} = \big(w_1, w_2\big)^T \in \mathbb{R}^2$. Un autre exemple de paramètres est le poids attribué à une connexion entre deux neurones dans un RNA.

Voir aussi: apprentissage automatique, modèle, hypothèse, fonction, modèle linéaire, poids

paramètres du modèle Les paramètres d'un modèle sont des quantités utilisées pour sélectionner une fonction hypothèse spécifique à partir d'un modèle. On peut considérer une liste de paramètres de modèle comme un identifiant unique d'une fonction hypothèse, de la même manière qu'un numéro de sécurité sociale identifie une personne en France.

Voir aussi: modèle, paramètre, hypothèse, application.

pas de gradient (pas) Étant donnée une fonction dérivable à valeurs réelles $f(\cdot) : \mathbb{R}^d \to \mathbb{R}$ et un vecteur $\mathbf{w} \in \mathbb{R}^d$, le pas de gradient met à jour $\mathbf{w}$ en ajoutant le gradient négatif mis à l'échelle $\nabla f(\mathbf{w})$ pour obtenir le nouveau vecteur (voir Figure 32)

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

Mathématiquement, le pas est un opérateur (typiquement non-linéaire) $\mathcal{T}^{(f,\eta)}$ paramétré par la fonction f et la taille de pas η.

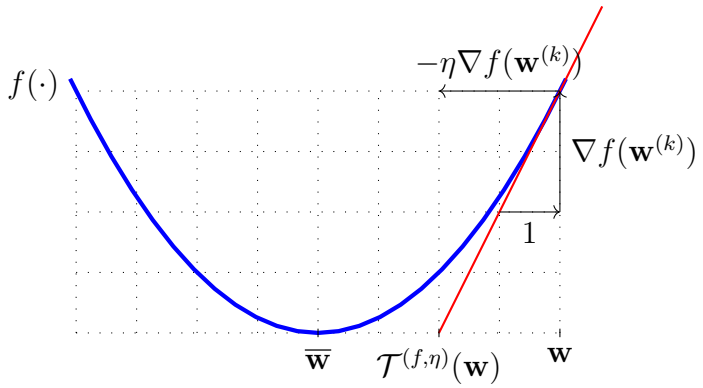


Fig. 32. Le pas classique (8) transforme un vecteur donné $\mathbf{w}$ en le vecteur mis à jour $\mathbf{w}'$. Il définit un opérateur $\mathcal{T}^{(f,\eta)}(\cdot) : \mathbb{R}^d \to \mathbb{R}^d : \mathbf{w} \mapsto \widehat{\mathbf{w}}$.

Notez que le pas (8) optimise localement - dans un voisinage dont la taille est déterminée par la taille de pas η - une approximation linéaire de la fonction $f(\cdot)$. Une généralisation naturelle de (8) est d'optimiser localement la fonction elle-même - au lieu de son approximation linéaire - telle que

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

Nous utilisons intentionnellement le même symbole η pour le paramètre dans (9) que celui utilisé pour la taille de pas dans (8). Plus le η choisi dans (9) est grand, plus la mise à jour avancera vers la réduction de la valeur de la fonction $f(\widehat{\mathbf{w}})$. Notez que, tout comme le pas (8), la mise à jour (9) définit aussi un opérateur (typiquement non-linéaire) paramétré par la fonction $f(\cdot)$ et le paramètre η. Pour une fonction convexe $f(\cdot)$, cet opérateur est connu sous le nom de opérateur proximal de $f(\cdot)$ [78].

Voir aussi: dérivable, fonction, gradient, taille de pas, voisinage, généralisation, paramètre, taux d'apprentissage, convexe, opérateur proximal.

perte (ou coût) En apprentissage automatique, on utilise une fonction de perte $L(\mathbf{z}, h)$ pour mesurer l'erreur commise lorsqu'une hypothèse est appliquée à une point de données. Par léger abus de langage, on utilise le terme perte à la fois pour désigner la fonction de perte L elle-même et la valeur spécifique $L(\mathbf{z}, h)$ associée à un point de données $\mathbf{z}$ et une hypothèse h.

Voir aussi: apprentissage automatique, fonction de perte, hypothèse, point de données.

perte logistique Considérons un point de données caractérisé par des car-

actéristiques $\mathbf{x}$ et une étiquette binaire $y \in \{-1, 1\}$. ON utilise une hypothèse à valeurs réelle h pour prédire l'étiquette y à partir des caractéristiques $\mathbf{x}$. La perte logistique associée à cette prédiction est définie comme suit :

$$L\left((\mathbf{x}, y), h\right) := \log\left(1 + \exp\left(-yh(\mathbf{x})\right)\right). \tag{10}$$

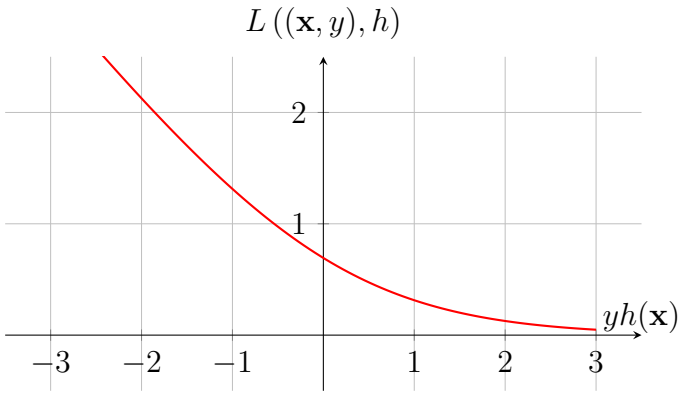


Fig. 33. La perte logistique associée à la prédiction $h(\mathbf{x}) \in \mathbb{R}$ pour un point de données d'étiquette $y \in \{-1, 1\}$.

Il est à noter que l'expression (10) de la perte logistique est valable uniquement lorsque l'espace des étiquettes est $\mathcal{Y} = \{-1, 1\}$ et que la règle de décision utilisée est celle du seuil définie en (1).

Voir aussi : point de données, caractéristique, étiquette, hypothèse, perte, prédiction, espace des étiquettes.

perte quadratique La perte quadratique mesure l'erreur de prédiction d'une hypothèse h lorsqu'elle prédit une étiquette numérique $y \in \mathbb{R}$ à

partir des caractéristiques $\mathbf{x}$ d'un point de données. Elle est définie par

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

Voir aussi: perte, prédiction, hypothèse, étiquette, caractéristique, point de données.

poids Considérons un espace des hypothèses paramétré $\mathcal{H}$. On utilise le terme poids pour désigner des paramètres du modèle numériques utilisés pour pondérer les caractéristiques ou leurs transformations afin de calculer $h^{(\mathbf{w})} \in \mathcal{H}$. Un modèle linéaire utilise des poids $\mathbf{w} = \big(w_1, \ldots, w_d\big)^T$ pour calculer la combinaison linéaire $h^{(\mathbf{w})}(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$. Les poids sont également utilisés dans les RNA pour former des combinaisons linéaires de caractéristiques ou des sorties de neurones dans les couches cachées.

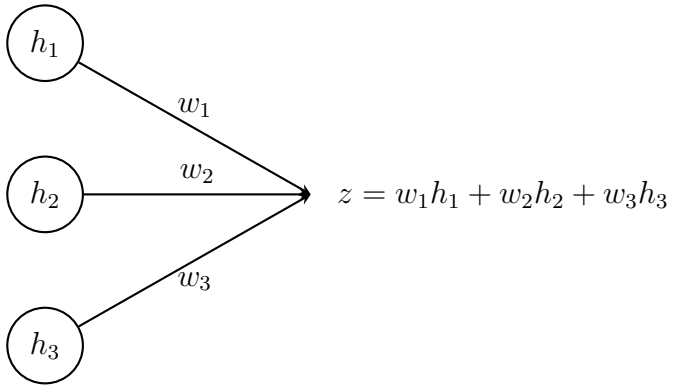


Fig. 34. Une section d'un RNA contenant une couche cachée avec des sorties (ou activations) h_1, h_2 et h_3. Ces sorties sont combinées linéairement pour calculer z, qui peut être utilisé soit comme sortie du RNA, soit comme entrée d'une autre couche.

Voir aussi: espace des hypothèses, paramètres du modèle, caractéristique, modèle linéaire, RNA.

poids d'arête Chaque arête $\{i, i'\}$ d'un réseau d'apprentissage fédéré est associée à un poids d'arête non négatif $A_{i,i'} \geq 0$. Un poids d'arête nul $A_{i,i'} = 0$ indique l'absence d'une arête entre les sommets $i, i' \in \mathcal{V}$.
Voir aussi: réseau d'apprentissage fédéré.

point de données Un point de données est un objet qui transmet de l'information [30]. Parmi les exemples courants, on trouve des étudiants, des signaux radio, des arbres, des images, des VA, des nombres réels ou encore des protéines. On décrit les points de données d'un même type en les caractérisant selon deux catégories de propriétés:

1. Les caractéristiques sont des propriétés mesurables ou calculables du point de données. Elles peuvent être extraites automatiquement à l'aide de capteurs, d'ordinateurs ou d'autres systèmes de collecte de données. Par exemple, pour un point de données représentant un patient, une caractéristique pourrait être la masse corporelle.
2. Les étiquettes sont des faits de plus haut niveau (ou des quantités d'intérêt) associés au point de données. Leur détermination requiert souvent une expertise humaine ou un savoir spécifique au domaine. Pour un patient, un diagnostic de cancer posé par un médecin constituerait une étiquette.

La figure 35 prend une image comme exemple de point de données, avec ses caractéristiques et étiquettes. Il est important de noter que

la distinction entre caractéristiques et étiquettes n'est pas inhérente au point de données lui-même: il s'agit d'un choix de modélisation propre à l'application d'apprentissage automatique. La distinction entre caractéristiques et étiquettes n'est pas toujours tranchée. Une propriété considérée comme une étiquette dans un certain contexte (par exemple, un diagnostic de cancer) peut être traitée comme une caractéristique dans un autre — en particulier lorsqu'une automatisation fiable (par exemple, par analyse d'image) permet de la déterminer sans intervention humaine. De manière générale, l'apprentissage automatique vise à prédire l'étiquette d'un point de données à partir de ses caractéristiques.
Voir aussi: données, caractéristique, étiquette, jeu de données.

point de données étiqueté Un point de données dont l'étiquette est connue ou a été déterminée par un certain moyen, pouvant nécessiter une intervention humaine.
Voir aussi: point de données, étiquette.

porte dérobée Une attaque par porte dérobée désigne la manipulation intentionnelle du processus d'entraînement d'une méthode d'apprentissage automatique. Cette manipulation peut être mise en œuvre par perturbation de l'ensemble d'entraînement (via une attaque par empoisonnement de données) ou de l'algorithme d'optimisation utilisé par une méthode basée sur la MRE. Le but d'une telle attaque est de forcer l'hypothèse apprise $\hat{h}$ à produire certaines prédictions spécifiques pour une plage donnée de caractéristiques. Cette plage de caractéristiques joue le rôle de clé (ou déclencheur) permettant d'activer une porte dérobée, c'est-à-

Un seul point de données

Caractéristiques:

- $x_1, \ldots, x_{d_1}$: Intensités de couleur des pixels de l'image.
- x_{d_1+1}: Horodatage de la capture de l'image.
- x_{d_1+2}: Localisation spatiale de la capture.

Étiquettes:

- y_1: Nombre de vaches visibles.
- y_2: Nombre de loups visibles.
- y_3: État du pâturage (par ex. sain, en surpâturage).

Fig. 35. Illustration d'un point de données sous forme d'image. Différentes propriétés de l'image peuvent être utilisées comme caractéristiques, et des faits plus abstraits comme étiquettes.

dire de provoquer des prédictions anormales. Seul l'attaquant connaît la clé $\mathbf{x}$ et la prédiction anormale correspondante $\hat{h}(\mathbf{x})$.

Voir aussi : empoisonnement de données, attaque, IA digne de confiance.

principe de minimisation des données La réglementation européenne sur la protection des données inclut un principe de minimisation des données. Ce principe impose au responsable du traitement de limiter la collecte des informations personnelles à ce qui est directement pertinent et nécessaire pour atteindre un objectif spécifié. Les données doivent être conservées uniquement aussi longtemps que nécessaire pour remplir cet objectif [83, Article 5(1)(c)], [84].

Voir aussi: données.

probabilité On associe une valeur de probabilité, typiquement choisie dans l'intervalle $[0, 1]$, à chaque événement pouvant se produire dans une expérience aléatoire [6, 7, 36, 85].

problème d'optimisation Un problème d'optimisation est une structure mathématique constituée d'une fonction objective $f : \mathcal{U} \to \mathcal{V}$ définie sur une variable d'optimisation $\mathbf{w} \in \mathcal{U}$, ainsi que d'un ensemble réalisable $\mathcal{W} \subseteq \mathcal{U}$. L'ensemble d'arrivée $\mathcal{V}$ est supposé totalement ordonné, ce qui signifie que pour deux éléments $\mathbf{a}, \mathbf{b} \in \mathcal{V}$, on peut déterminer si $\mathbf{a} < \mathbf{b}$, $\mathbf{a} = \mathbf{b}$, ou $\mathbf{a} > \mathbf{b}$. Le but de l'optimisation est de trouver les valeurs $\mathbf{w} \in \mathcal{W}$ pour lesquelles l'objectif $f(\mathbf{w})$ est extrémal — c'est-à-dire minimal ou maximal [21], [72], [75].

Voir aussi: fonction objective.

processus gaussien Un processus gaussien est une collection de VA $\{f(\mathbf{x})\}_{\mathbf{x}\in\mathcal{X}}$

indexée par des valeurs d'entrée $\mathbf{x}$ d'un certain espace d'entrée $\mathcal{X}$, telle que, pour tout sous-ensemble fini $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)} \in \mathcal{X}$, les VA correspondantes $f(\mathbf{x}^{(1)}), \ldots, f(\mathbf{x}^{(m)})$ suivent une loi normale multivariée conjointe:

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

Pour un espace d'entrée $\mathcal{X}$ donné, un processus gaussien est complètement spécifié (ou paramétré) par: 1) une fonction moyenne $\mu(\mathbf{x}) = \mathbb{E}\{f(\mathbf{x})\}$; et 2) une fonction de covariance $K\big(\mathbf{x}, \mathbf{x}'\big) = \mathbb{E}\{\big(f(\mathbf{x}) - \mu(\mathbf{x})\big)\big(f(\mathbf{x}') - \mu(\mathbf{x}')\big)\}$. Exemple: On peut interpréter la distribution de température en Finlande (à un instant donné) comme la réalisation d'un processus gaussien $f(\mathbf{x})$, où chaque entrée $\mathbf{x} = (\text{lat}, \text{lon})$ désigne une localisation géographique. Les mesures de température provenant des stations météo du FMI constituent des échantillons de $f(\mathbf{x})$ en des lieux spécifiques (voir Fig. 36). Un processus gaussien permet de prédire la température à proximité des stations du FMI et de quantifier l'incertitude des prédictions.

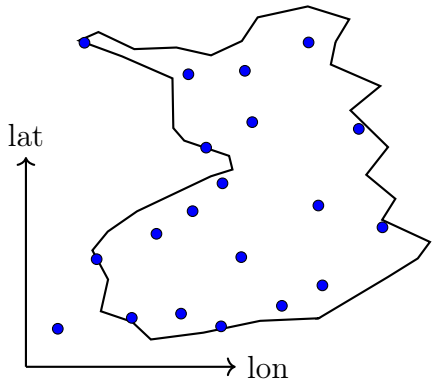


Fig. 36. On peut interpréter la distribution de température en Finlande comme une réalisation d'un processus gaussien indexé par les coordonnées géographiques et échantillonné aux stations météo du FMI (points bleus).

Voir aussi: VA, moyenne, fonction, covariance, réalisation, FMI, échantillon, incertitude.

processus stochastique Un processus stochastique est une collection de VA définies sur un même espace probabilisé, indexée par un ensemble $\mathcal{I}$ [32, 34, 86]. L'ensemble d'indexation $\mathcal{I}$ représente généralement le temps ou l'espace, ce qui permet de modéliser des phénomènes aléatoires évoluant dans le temps ou l'espace — par exemple, le bruit d'un capteur ou les séries temporelles financières. Les processus stochastiques ne se limitent pas à ces cadres temporels ou spatiaux. Par exemple, les graphes tels que le graphe d'Erdős-Rényi (graphe ER) ou le modèle à blocs stochastiques (SBM) peuvent également être vus comme des processus stochastiques. Ici, l'ensemble d'indexation $\mathcal{I}$ consiste en des paires de sommets indexant des VA dont les valeurs représentent la

présence ou le poids d'une arête entre deux nœuds. De plus, les processus stochastiques émergent naturellement dans l'analyse des algorithme stochastiques, tels que la SGD, qui construisent une suite de VA.

Voir aussi : VA, SBM, SGD, incertitude, modèle probabiliste.

produit de Kronecker Le produit de Kronecker de deux matrices $\mathbf{A} \in \mathbb{R}^{m\times n}$ et $\mathbf{B} \in \mathbb{R}^{p\times q}$ est une matrice par blocs notée $\mathbf{A} \otimes \mathbf{B}$ et définie comme suit [3], [27]:

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

Le produit de Kronecker est un cas particulier du produit tensoriel pour matrices et est largement utilisé en statistique multivariée, en algèbre linéaire, et dans les modèles d'apprentissage automatique structurés. Il satisfait l'identité $(\mathbf{A} \otimes \mathbf{B})(\mathbf{x} \otimes \mathbf{y}) = (\mathbf{A}\mathbf{x}) \otimes (\mathbf{B}\mathbf{y})$ pour des vecteurs $\mathbf{x}$ et $\mathbf{y}$ de dimensions compatibles.

Voir aussi: apprentissage automatique, modèle.

projection Considérons un sous-ensemble $\mathcal{W} \subseteq \mathbb{R}^d$ de l'espace euclidien de dimension d. On définit la projection $P_{\mathcal{W}}(\mathbf{w})$ d'un vecteur $\mathbf{w} \in \mathbb{R}^d$ sur $\mathcal{W}$ comme

$$P_{\mathcal{W}}(\mathbf{w}) = \underset{\mathbf{w}'\in\mathcal{W}}{\operatorname{argmin}} \|\mathbf{w} - \mathbf{w}'\|_2 . \tag{11}$$

Autrement dit, $P_{\mathcal{W}}(\mathbf{w})$ est le vecteur dans $\mathcal{W}$ qui est le plus proche de $\mathbf{w}$. La projection est bien définie uniquement pour les sous-ensembles $\mathcal{W}$ pour lesquels le minimum ci-dessus existe [21].

Voir aussi: espace euclidien, minimum.

protection de la vie privée Considérons une méthode d'apprentissage automatique $\mathcal{A}$ qui lit un jeu de données $\mathcal{D}$ et produit une sortie $\mathcal{A}(\mathcal{D})$. Cette sortie peut être les paramètres du modèle appris $\widehat{\mathbf{w}}$ ou la prédiction $\hat{h}(\mathbf{x})$ obtenue pour un point de données spécifique aux caractéristiques $\mathbf{x}$. De nombreuses applications importantes d'apprentissage automatique impliquent des points de données représentant des êtres humains. Chaque point de données est caractérisé par des caractéristiques $\mathbf{x}$, éventuellement une étiquette y et une donnée sensible s (par exemple, un diagnostic médical récent). De manière générale, la protection de la vie privée signifie qu'il doit être impossible de déduire, à partir de la sortie $\mathcal{A}(\mathcal{D})$, les données sensibles des points de données présents dans $\mathcal{D}$. Mathématiquement, la protection de la vie privée exige la non-inversibilité de l'application $\mathcal{A}(\mathcal{D})$. En pratique, rendre $\mathcal{A}(\mathcal{D})$ simplement non inversible est souvent insuffisant ; il faut que $\mathcal{A}(\mathcal{D})$ soit suffisamment non inversible pour garantir la protection de la vie privée.

Voir aussi : apprentissage automatique, jeu de données, paramètres du modèle, prédiction, point de données, caractéristique, étiquette, donnée sensible, application.

prédiction Une prédiction est une estimation ou une approximation d'une certaine quantité d'intérêt. L'apprentissage automatique se concentre sur l'apprentissage ou la recherche d'une fonction hypothèse qui prend en entrée les caractéristiques $\mathbf{x}$ d'un point de données et fournit une prédiction $\widehat{y} := h(\mathbf{x})$ pour son étiquette y.

Voir aussi: apprentissage automatique, hypothèse, application, caractéristique, point de données, étiquette.

pseudo-inverse La pseudo-inverse de Moore–Penrose $\mathbf{A}^{+}$ d'une matrice $\mathbf{A} \in \mathbb{R}^{m \times d}$ généralise la notion de matrice inverse [3]. La pseudo-inverse apparaît naturellement dans le cadre de la régression Ridge appliquée à un jeu de données avec des étiquettes arbitraires $\mathbf{y}$ et une matrice de caractéristiques $\mathbf{X} = \mathbf{A}$ [71, Ch. 3]. Les paramètres du modèle appris par la régression Ridge sont donnés par

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

On peut alors définir la pseudo-inverse $\mathbf{A}^{+} \in \mathbb{R}^{d \times m}$ avec la limite [87, Ch. 3]

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

Voir aussi: matrice inverse, régression Ridge, jeu de données, étiquette, matrice de caractéristiques, paramètres du modèle.

regret Le regret d'une hypothèse h par rapport à une autre hypothèse h' (considérée comme niveau de référence) est défini comme la différence entre la perte engendrée par h et celle engendrée par h' [11]. L'hypothèse de référence h' est aussi appelée un expert.

Voir aussi: niveau de référence, perte, expert.

risque Considérons une hypothèse h utilisée pour prédire l'étiquette y d'un point de données basée sur ses caractéristiques $\mathbf{x}$. Nous mesurons la qualité d'une prédiction particulière en utilisant une fonction de perte $L\left((\mathbf{x}, y), h\right)$. Si nous interprétons les points de données comme les réalisations de VA i.i.d., plural=i.i.d., alors $L\left((\mathbf{x}, y), h\right)$ devient la réalisation d'une VA. L'hypothèse i.i.d. nous permet de définir le risque d'une

hypothèse comme l'espérance de la perte $\mathbb{E}\big\{L\left((\mathbf{x}, y), h\right)\big\}$. Notons que le risque de h dépend à la fois du choix spécifique de la fonction de perte et de la loi de probabilité des points de données.

Voir aussi: hypothèse, étiquette, point de données, caractéristique, prédiction, fonction de perte, réalisation, i.i.d., plural=i.i.d. VA, hypothèse i.i.d., perte, loi de probabilité.

risque bayésien Considérons un modèle probabiliste avec une loi de probabilité conjointe $p(\mathbf{x}, y)$ pour les caractéristiques $\mathbf{x}$ et l'étiquette y d'un point de données. Le risque bayésien est le minimum possible de risque qui peut être atteint par toute hypothèse $h : \mathcal{X} \to \mathcal{Y}$. Toute hypothèse atteignant le risque bayésien est appelée un estimateur bayésien [37].

Voir aussi: modèle probabiliste, risque, estimateur bayésien.

risque empirique Le risque empirique $\widehat{L}\big(h|\mathcal{D}\big)$ d'une hypothèse sur un jeu de données $\mathcal{D}$ correspond à la perte moyenne encourue par h lorsqu'elle est appliquée aux différents points de données de $\mathcal{D}$.

Voir aussi: risque, hypothèse, jeu de données, perte, point de données.

robustesse La robustesse est une exigence clé pour une IA digne de confiance. Elle désigne la capacité d'un système d'apprentissage automatique à maintenir des performances acceptables même lorsqu'il est soumis à différentes formes de perturbations. Ces perturbations peuvent affecter les caractéristiques d'un point de données dans le but de manipuler la prédiction produite par un modèle d'apprentissage automatique entraîné. La robustesse englobe également la stabilité des méthodes basées sur la MRE face à des perturbations de l'ensemble d'entraînement, notamment

dans le cadre d'attaques par empoisonnement de données.

Voir aussi : IA digne de confiance, apprentissage automatique, caractéristique, point de données, prédiction, modèle, stabilité, MRE, ensemble d'entraînement, empoisonnement de données, attaque.

Règlement général sur la protection des données (RGPD) Le RGPD a été promulgué par l'Union européenne (UE) et est entré en vigueur le 25 mai 2018 [83]. Il garantit la protection de la vie privée et des droits liés aux données des individus au sein de l'UE. Le RGPD a des implications importantes sur la manière dont les données sont collectées, stockées et utilisées dans les applications d'apprentissage automatique. Parmi ses dispositions principales, on trouve:

- Principe de minimisation des données: les systèmes d'apprentissage automatique ne doivent utiliser que la quantité de données personnelles strictement nécessaire à leur finalité.
- Transparence et explicabilité: les systèmes d'apprentissage automatique doivent permettre aux utilisateurs de comprendre comment sont prises les décisions les concernant.
- Droits des personnes concernées: les utilisateurs doivent pouvoir accéder à leurs données personnelles, les rectifier, les supprimer, et s'opposer aux décisions automatisées ainsi qu'au profilage.
- Responsabilité: les organisations doivent garantir une sécurité robuste des données et prouver leur conformité au RGPD par la documentation et des audits réguliers.

Voir aussi: données, apprentissage automatique, principe de minimisation des données, transparence, explicabilité.

réalisation Considérons une VA $\mathbf{x}$ qui associe à chaque issue $\omega \in \mathcal{P}$ d'un espace probabilisé $\mathcal{P}$ un élément a d'un espace mesurable $\mathcal{N}$ [2], [6], [36]. Une réalisation de $\mathbf{x}$ est tout élément $\mathbf{a} \in \mathcal{N}$ tel qu'il existe un $\omega \in \mathcal{P}$ vérifiant $\mathbf{x}(\omega) = \mathbf{a}$.

Voir aussi : VA, espace probabilisé.

récompense Une récompense désigne une quantité observée (ou mesurée) qui permet d'estimer la perte subie par la prédiction (ou décision) d'une hypothèse $h(\mathbf{x})$. Par exemple, dans une application d'apprentissage automatique pour véhicules autonomes, $h(\mathbf{x})$ pourrait représenter la direction actuelle du volant d'un véhicule. On peut construire une récompense à partir des mesures d'un capteur de collision indiquant si le véhicule se dirige vers un obstacle. Une faible récompense est donnée à la direction $h(\mathbf{x})$ si le véhicule avance dangereusement vers un obstacle.

Voir aussi: perte, prédiction, hypothèse, apprentissage automatique.

réduction de dimension La réduction de dimension désigne les méthodes qui apprennent une transformation $h : \mathbb{R}^d \to \mathbb{R}^{d'}$ d'un ensemble (généralement grand) de caractéristiques brutes $x_1, \ldots, x_d$ en un ensemble plus petit de caractéristiques informatives $z_1, \ldots, z_{d'}$. L'utilisation d'un ensemble réduit de caractéristiques présente plusieurs avantages :

- **Avantage statistique :** elle réduit généralement le risque de surapprentissage, car réduire le nombre de caractéristiques réduit souvent la dimension effective d'un modèle.

- **Avantage computationnel :** utiliser moins de caractéristiques signifie moins de calculs lors de l'entraînement des modèles d'apprentissage automatique. Par exemple, les méthodes de régression linéaire doivent inverser une matrice dont la taille dépend du nombre de caractéristiques.
- **Visualisation :** la réduction de dimension est également utile pour la visualisation des données. Par exemple, on peut apprendre une transformation qui produit deux caractéristiques z_1, z_2, que l'on peut utiliser comme coordonnées d'un nuage de points. La Fig. 37 montre un nuage de points de chiffres manuscrits placés selon des caractéristiques transformées. Ici, les points de données sont initialement représentés par un grand nombre de niveaux de gris (un par pixel).

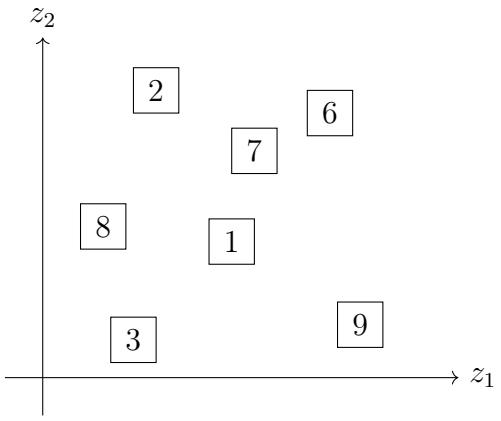


Fig. 37. Exemple de réduction de dimension : des données image en haute dimension (par exemple, des images haute résolution de chiffres manuscrits) sont projetées en 2D en utilisant des caractéristiques apprises (z_1, z_2) et visualisées dans un nuage de points.

Voir aussi : surapprentissage, dimension effective, modèle, nuage de points.

région de décision Considérons une fonction hypothèse qui renvoie des valeurs d'un ensemble fini $\mathcal{Y}$. Pour chaque valeur (catégorie) d'étiquette $a \in \mathcal{Y}$, l'hypothèse h détermine un sous-ensemble de valeurs de caractéristiques $\mathbf{x} \in \mathcal{X}$ telles que $h(\mathbf{x}) = a$. On appelle ce sous-ensemble une région de décision de l'hypothèse h.

Voir aussi: hypothèse, application, étiquette, caractéristique.

régression Les problèmes de régression se concentrent sur la prédiction d'une étiquette numérique uniquement à partir des caractéristiques d'un point de données [8, Ch. 2].

Voir aussi: prédiction, étiquette, caractéristique, point de données.

régression linéaire La régression linéaire vise à apprendre une fonction hypothèse linéaire pour prédire une étiquette numérique à partir des caractéristiques numériques d'un points de données. La qualité d'une fonction hypothèse linéaire est mesurée par la moyenne de la perte quadratique subie sur un ensemble de point de données étiquetés, que nous appelons l'ensemble d'entraînement.

Voir aussi: régression, hypothèse, application, étiquette, caractéristique, point de données, perte quadratique, point de données étiqueté, ensemble d'entraînement.

régression logistique La régression logistique apprend une fonction hypothèse (ou classifieur) linéaire $h(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ pour prédire une étiquette binaire y à partir du vecteur de caractéristiques numérique $\mathbf{x}$ d'un

point de données. La qualité d'une telle fonction hypothèse linéaire est mesurée à l'aide de la perte logistique moyenne sur un ensemble de points de données étiquetés (c'est-à-dire l'ensemble d'entraînement).

Voir aussi: régression, hypothèse, application, classifieur, étiquette, vecteur de caractéristiques, point de données, perte logistique, point de données étiqueté, ensemble d'entraînement.

régression polynomiale La régression polynomiale est une instance de MRE vise à apprendre une fonction hypothèse polynomiale pour prédire une étiquette numérique à partir des caractéristiques numériques d'un point de données. Pour des points de données caractérisés par une seule caractéristique numérique, la régression polynomiale utilise l'espace des hypothèses $\mathcal{H}_d^{(\text{poly})} := \{h(x) = \sum_{j=0}^{d-1} x^j w_j\}$. La qualité d'une fonction hypothèse polynomiale est mesurée via la perte quadratique moyenne encourue sur un ensemble de points de données étiquetés (appelé ensemble d'entraînement).

Voir aussi: régression, MRE, perte quadratique.

régression Ridge La régression Ridge apprend les poids $\mathbf{w}$ d'une fonction hypothèse linéaire $h^{(\mathbf{w})}(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$. La qualité d'un choix particulier des paramètres du modèle $\mathbf{w}$ est mesurée par la somme de deux composantes. La première composante est la moyenne de la perte quadratique subie par $h^{(\mathbf{w})}$ sur un ensemble de points de données étiquetés (i.e., l'ensemble d'entraînement). La deuxième composante est la norme euclidienne au carré, mise à l'échelle, $\alpha\|\mathbf{w}\|_2^2$ avec un paramètre de régularisation $\alpha > 0$. Ajouter $\alpha\|\mathbf{w}\|_2^2$ à la moyenne de la perte quadratique équivaut

à remplacer chaque points de données initial par la réalisation d'une infinité sw VA i.i.d., plural=i.i.d. centrées autour de ces points de données.

Voir aussi: régression, poids, hypothèse, application, paramètres du modèle, perte quadratique, point de données étiqueté, ensemble d'entraînement, norme, régularisation, paramètre, point de données, réalisation, i.i.d., plural=i.i.d., VA.

régularisation Un défi majeur des applications modernes d'apprentissage automatique est qu'elles utilisent souvent de grands modèles, avec une dimension effective de l'ordre du milliard. Entraîner un modèle de grande dimension à l'aide de méthodes de MRE basiques conduit souvent au surapprentissage: l'hypothèse apprise a de bonnes performances sur l' ensemble d'entraînement mais insuffisantes en dehors de celui-ci. La régularisation désigne des modifications apportées à une instance donnée de MRE afin d'éviter le surapprentissage, c'est-à-dire pour garantir que l'hypothèse apprise fonctionne presque aussi bien en dehors de l'ensemble d'entraînement. Il existe trois manières de mettre en œuvre la régularisation:

1) Élaguer le modèle: on réduit le modèle original $\mathcal{H}$ pour obtenir un modèle plus petit $\mathcal{H}'$. Dans le cas d'un modèle paramétrique, cette réduction peut se faire via des contraintes sur les paramètres du modèle (par exemple $w_1 \in [0.4, 0.6]$ pour le poids de la caractéristique x_1 dans la régression linéaire).

2) Pénaliser la perte: on modifie la fonction objective de la MRE en

ajoutant un terme de pénalité à l'erreur d'entrainement. Ce terme estime combien la perte (ou le risque) attendue est plus grande que la perte moyenne sur l'ensemble d'entraînement.

3) Augmentation de données: on peut agrandir l'ensemble d'entraînement $\mathcal{D}$ en ajoutant des copies perturbées des points de données originaux de $\mathcal{D}$. Une telle perturbation consiste par exemple à ajouter la réalisation d'une VA au vecteur de caractéristiques d'un point de données.

La figure 38 illustre ces trois approches de régularisation. Ces approches sont étroitement liées et parfois entièrement équivalentes: la augmentation de données qui utilise des VA normales pour perturber les vecteurs de caractéristiques de l'ensemble d'entraînement dans le cas de la régression linéaire a le même effet que l'ajout du terme de pénalité $\lambda \|\mathbf{w}\|_2^2$ à l'erreur d'entrainement (ce qui correspond à la régression Ridge). Le choix de la méthode de régularisation peut dépendre des ressources de calcul disponibles. Par exemple, il peut être bien plus facile de mettre en œuvre une augmentation de données que de réaliser un élagage de modèle.

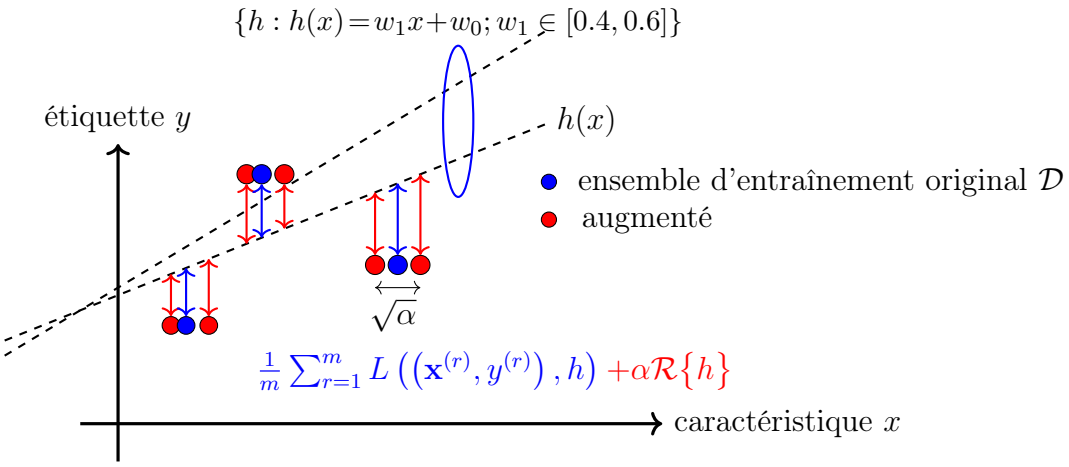


Fig. 38. Trois approches pour la régularisation: 1) augmentation de données; 2) pénalisation de la perte; et 3) élagage du modèle (via des contraintes sur les paramètres du modèle).

Voir aussi: surapprentissage, augmentation de données, validation, sélection du modèle.

régulière (ou lisse) Une fonction à valeurs réelles $f : \mathbb{R}^d \to \mathbb{R}$ est dite régulière (ou lisse) si elle est dérivable et si son gradient $\nabla f(\mathbf{w})$ est continu en tout point $\mathbf{w} \in \mathbb{R}^d$ (on parle aussi de fonction de classe $\mathcal{C}^1$) [75, 88]. Une fonction régulière f est dite dérivable de gradient β-lipschitzien (ou β-*smooth*) si son gradient $\nabla f(\mathbf{w})$ vérifie:

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

La constante β mesure le degré de régularité de la fonction f: plus β est petit, plus f est lisse. Les problèmes d'optimisation comportant une fonction objective régulière peuvent être résolus efficacement par des

méthodes basées sur le gradient. En effet, les méthodes basées sur le gradient approximent la fonction objective localement autour d'un point courant $\mathbf{w}$ en utilisant son gradient. Cette approximation est pertinente lorsque le gradient ne varie pas trop rapidement. Cette affirmation intuitive peut être rendue rigoureuse en étudiant l'effet d'un seul pas avec une taille de pas $\eta = 1/\beta$ (voir Figure 39).

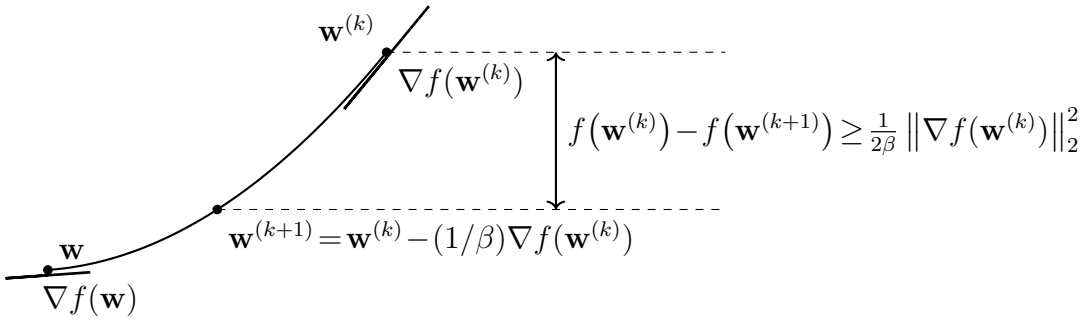


Fig. 39. Considérons une fonction objective $f(\mathbf{w})$ qui est β-*smooth*. Effectuer un pas avec une taille de pas $\eta = 1/\beta$ diminue la fonction objective d'au moins $\frac{1}{2\beta} \left\|\nabla f(\mathbf{w}^{(k)})\right\|_2^2$ [75, 88, 89]. Notez que la taille de pas $\eta = 1/\beta$ devient plus grande lorsque β diminue. Ainsi, pour des fonctions objectives plus lisses (c'est-à-dire avec un plus petit β), on peut effectuer des pas plus grands.

Voir aussi: fonction, dérivable, gradient, problème d'optimisation, fonction objective, méthodes basées sur le gradient, pas, taille de pas.

réseau de neurones artificiels (RNA) Un RNA est une représentation graphique (circulation de signaux) d'une fonction qui associe les caractéristiques d'un point de données en entrée à une prédiction de l'étiquette correspondante en sortie. L'unité fondamentale d'un RNA est le neurone

artificiel, qui applique une fonction d'activation à ses entrées pondérées. Les sorties de ces neurones servent d'entrées à d'autres neurones, formant des couches interconnectées.

Voir aussi: fonction, caractéristique, point de données, prédiction, étiquette, fonction d'activation.

réseau de neurones profond Un réseau de neurones profond est un RNA avec un nombre (relativement) élevé de couches cachées. L'apprentissage profond est un terme générique désignant les méthodes d'apprentissage automatique qui utilisent un réseau de neurones profond comme modèle [90].

Voir aussi: RNA, apprentissage automatique, modèle.

réseau d'apprentissage fédéré Un réseau d'apprentissage fédéré consiste en un graphe non orienté et pondéré $\mathcal{G}$. Les sommets de $\mathcal{G}$ représentent des appareils ayant chacun accès à un jeu de données local et capables d'entraîner un modèle local. Les arêtes de $\mathcal{G}$ représentent à la fois les liens de communication entre les appareils et les similarités statistiques entre leurs jeux de données locaux. Une approche rigoureuse pour entraîner les modèles locaux est minimisation de la variation totale généralisée (GTVMin). Les solutions de GTVMin correspondent à des paramètres du modèle locaux qui équilibrent de manière optimale la perte subie sur les jeux de données locaux et leur divergence le long des arêtes de $\mathcal{G}$.

Voir aussi : apprentissage fédéré, graphe, appareil, GTVMin.

semi-définie positive Une matrice symétrique (à valeurs réelles) $\mathbf{Q} = \mathbf{Q}^T \in$

$\mathbb{R}^{d\times d}$ est dite semi-définie positive si $\mathbf{x}^T\mathbf{Q}\mathbf{x} \geq 0$ pour tout vecteur $\mathbf{x} \in \mathbb{R}^d$. La propriété d'être semi-définie positive peut être étendue des matrices aux noyaux symétriques (à valeurs réelles) $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ (avec $K(\mathbf{x}, \mathbf{x}') = K(\mathbf{x}', \mathbf{x})$) de la manière suivante: pour tout ensemble fini de vecteurs de caractéristiques $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}$, la matrice résultante $\mathbf{Q} \in \mathbb{R}^{m\times m}$ avec pour coefficients $Q_{r,r'} = K\left(\mathbf{x}^{(r)}, \mathbf{x}^{(r')}\right)$ est semi-définie positive [80].

Voir aussi: noyau, application, vecteur de caractéristiques.

sous-apprentissage Considérons une méthode d'apprentissage automatique qui utilise la MRE pour apprendre une hypothèse minimisant le risque empirique sur un ensemble d'entraînement donné. On dit que cette méthode est en situation de sous-apprentissage si elle n'est pas capable d'apprendre une hypothèse avec un risque empirique suffisamment faible sur l'ensemble d'entraînement. En général, une méthode en situation de sous-apprentissage ne parviendra pas non plus à apprendre une hypothèse avec un risque faible.

Voir aussi: apprentissage automatique, MRE, hypothèse, minimum, risque empirique, ensemble d'entraînement, risque.

stabilité La stabilité est une propriété souhaitable d'une méthode d'apprentissage automatique $\mathcal{A}$ qui associe un jeu de données $\mathcal{D}$ (par exemple un ensemble d'entraînement) à une sortie $\mathcal{A}(\mathcal{D})$. Cette sortie peut correspondre aux paramètres du modèle appris ou à la prédiction produite par le modèle entraîné pour un point de données donné. Intuitivement, $\mathcal{A}$ est stable si de petits changements dans l'entrée $\mathcal{D}$ entraînent de petits

changements dans la sortie $\mathcal{A}(\mathcal{D})$. Il existe plusieurs définitions formelles de la stabilité permettant d'établir des bornes sur l'erreur de généralisation ou le risque de la méthode (voir [46, Ch. 13]). Pour illustrer cette notion, considérons les trois jeux de données représentés dans la Fig. 40, équiprobables selon la même loi de probabilité de génération de données. Étant donné que les paramètres du modèle optimaux sont déterminés par cette distribution sous-jacente, une méthode d'apprentissage automatique précise $\mathcal{A}$ devrait produire une sortie identique (ou très similaire) $\mathcal{A}(\mathcal{D})$ pour ces trois jeux de données. Autrement dit, toute méthode $\mathcal{A}$ utile doit pouvoir résister à la variabilité des réalisations échantillonnées selon une même loi de probabilité : elle doit être stable.

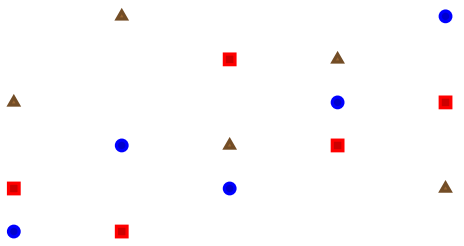


Fig. 40. Trois jeux de données $\mathcal{D}^{(*)}$, $\mathcal{D}^{(\square)}$ et $\mathcal{D}^{(\triangle)}$, chacun échantillonné indépendamment selon la même loi de probabilité de génération de données. Une méthode d'apprentissage automatique stable devrait fournir des sorties similaires lorsqu'elle est entraînée sur chacun de ces jeux de données.

Voir aussi : apprentissage automatique, jeu de données, ensemble

d'entraînement, paramètres du modèle, prédiction, modèle, point de données, généralisation, risque, données, loi de probabilité, échantillon, réalisation.

stochastique Une méthode est dite si elle comporte une composante aléatoire ou si elle est régie par des lois probabilistes. Les méthodes d'apprentissage automatique utilisent l'aléatoire pour réduire la complexité computationnelle (voir, par exemple, SGD) ou pour modéliser l'incertitude dans les modèles probabilistes.

Voir aussi: incertitude, modèle probabiliste, SGD.

surapprentissage Considérons une méthode d'apprentissage automatique qui utilise la MRE pour apprendre une hypothèse avec le risque empirique minimal sur un ensemble d'entraînement donné. Une telle méthode fait du surapprentissage sur l'ensemble d'entraînement si elle apprend une hypothèse avec un petit risque empirique sur l'ensemble d'entraînement mais une perte significativement plus grande en dehors de cet ensemble.

Voir aussi:

glserm, généralisation, validation, écart de généralisation.

sélection du modèle En apprentissage automatique, la sélection du modèles fait référence au processus de choix entre différents modèles candidats. Dans sa forme la plus élémentaire, la sélection du modèle consiste à : 1) entraîner chaque modèle candidat ; 2) calculer l'erreur de validation pour chaque modèle entraîné ; et 3) choisir le modèle ayant la plus petite erreur de validation [8, Ch. 6].

Voir aussi: apprentissage automatique, modèle, erreur de validation.

taille d'échantillon Le nombre de points de données individuels contenus dans un jeu de données.

Voir aussi: point de données, jeu de données.

taille de pas Voir taux d'apprentissage.

taux d'apprentissage Considérons une méthode itérative d'apprentissage automatique pour trouver ou apprendre une hypothèse utile $h \in \mathcal{H}$. Une telle méthode itérative répète des étapes computationnelles (de mise à jour) similaires qui ajustent ou modifient l'hypothèse actuelle afin d'obtenir une hypothèse améliorée. Un exemple bien connu de cette méthode itérative est la descente de gradient et ses variantes, SGD et la descente de gradient avec projection. Un paramètre clé d'une méthode itérative est le taux d'apprentissage. Le taux d'apprentissage contrôle l'ampleur selon laquelle l'hypothèse courante peut être modifiée durant une seule itération. Un exemple bien connu de tel paramètre est la taille de pas utilisée lors d'une descente de gradient [8, Ch. 5].

Voir aussi: apprentissage automatique, hypothèse, descente de gradient, SGD, descente de gradient avec projection, paramètre, taille de pas.

terme de régularisation Un terme de régularisation assigne à chaque hypothèse h d'un espace des hypothèses $\mathcal{H}$ une mesure quantitative $\mathcal{R}\big\{h\big\}$ représentant à quel point l'erreur de prédiction sur un ensemble d'entraînement pourrait différer des erreurs de prédiction sur des points de données en dehors de l'ensemble d'entraînement. La régression Ridge

utilise le terme de régularisation $\mathcal{R}\big\{h\big\} := \|\mathbf{w}\|_2^2$ pour des fonctions hypothèses linéaires de la forme $h^{(\mathbf{w})}(\mathbf{x}) := \mathbf{w}^T\mathbf{x}$ [8, Ch. 3]. Le Lasso utilise quant à lui $\mathcal{R}\big\{h\big\} := \|\mathbf{w}\|_1$ pour des hypothèses linéaires de même forme [8, Ch. 3].

Voir aussi : hypothèse, espace des hypothèses, prédiction, ensemble d'entraînement, point de données, régression Ridge, application, Lasso.

théorème central limite (TCL) Considérons une séquence de VA i.i.d., plural=i.i.d.s $x^{(r)}$, pour $r = 1, 2, \ldots$, ayant toutes une moyenne nulle et une variance finie $\sigma^2 > 0$. Le TCL affirme que la somme normalisée

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

converge en loi vers une VA normale de moyenne nulle et de variance σ^2 quand $m \to \infty$ [91, Proposition 2.17]. Une manière élégante de démontrer le TCL est d'utiliser la fonction caractéristique de la somme normalisée $s^{(m)}$. Soit $\phi(t) = \mathbb{E}\big\{\exp\big(jtx\big)\big\}$ (avec l'unité imaginaire $j = \sqrt{-1}$) la fonction caractéristique commune de chaque somme et des $x^{(r)}$, et soit $\phi^{(m)}(t)$ la fonction caractéristique de $s^{(m)}$. Définissons un opérateur $\mathcal{T}$ agissant sur les fonctions caractéristiques tel que

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

Cette méthode du point fixe capture l'effet de l'ajout récursif d'une VA i.i.d., plural=i.i.d. $\mathbf{x}^{(m)}$ et de la renormalisation. L'application itérative de $\mathcal{T}$ conduit à la convergence de $\phi^{(m)}(t)$ vers le point fixe

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

qui est la fonction caractéristique d'une VA normale de moyenne nulle et de variance σ^2. Les généralisations du TCL autorisent des VA dépendantes ou non identiquement distribuées [91, Sec. 2.8].

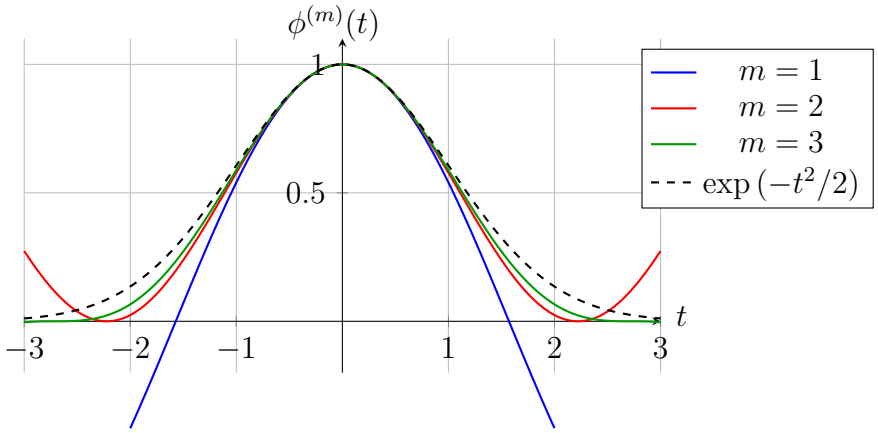


Fig. 41. Les fonctions caractéristiques des sommes normalisées de VA i.i.d., plural=i.i.d.s $x^{(r)} \in \{-1, 1\}$ pour $r = 1, \ldots, m$, comparées à la limite gaussienne.

Voir aussi: VA, VA normale.

transformation de caractéristiques Une transformation de caractéristiques désigne une fonction

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

qui transforme un vecteur de caractéristiques $\mathbf{x} \in \mathcal{X}$ d'un point de données en un nouveau vecteur de caractéristiques $\mathbf{x}' \in \mathcal{X}'$, où $\mathcal{X}'$ est généralement différent de $\mathcal{X}$. La représentation transformée $\mathbf{x}'$ est souvent plus utile que l'originale $\mathbf{x}$. Par exemple, la géométrie des

points de données peut devenir plus linéaire dans $\mathcal{X}'$, ce qui permet d'appliquer un modèle linéaire à $\mathbf{x}'$. Cette idée est centrale dans la conception des méthodes à noyau [80]. D'autres avantages incluent la réduction du surapprentissage et l'amélioration de l'interprétabilité [44]. Un cas d'usage courant est la visualisation de données, où une transformation de caractéristiques à deux dimensions permet de représenter les points de données dans un nuage de points en 2D. Certaines méthodes d'apprentissage automatique utilisent des transformations de caractéristiques apprises à partir des données. C'est le cas, par exemple, des couches cachées d'un réseau de neurones profond, qui agissent comme des transformations successives [48]. Une manière rigoureuse d'entraîner une transformation de caractéristiques est d'utiliser la MRE avec une fonction de perte mesurant la qualité de reconstruction, par exemple $L = \|\mathbf{x} - r(\mathbf{x}')\|^2$, où $r(\cdot)$ est une application entraînable visant à reconstruire $\mathbf{x}$ à partir de la version transformée $\mathbf{x}'$.

Voir aussi : caractéristique, application, méthode à noyau, apprentissage de caractéristiques, PCA.

transparence La transparence est une exigence fondamentale pour une IA digne de confiance [92]. Dans le contexte des méthodes d'apprentissage automatique, le terme est souvent utilisé de manière interchangeable avec explicabilité [40, 93]. Cependant, dans le cadre plus large des systèmes d'IA, la transparence va au-delà de l'explicabilité et inclut de fournir des informations sur les limitations, la fiabilité et l'utilisation prévue du système. Dans les systèmes de diagnostic médical, la transparence exige de révéler le niveau de confiance associé aux prédictions produites

par un modèle entraîné. Dans l'évaluation du crédit, les décisions prises par des systèmes d'IA doivent être accompagnées d'explications sur les facteurs contributifs, tels que le revenu ou l'historique de crédit. Ces explications permettent aux humains (par exemple, un demandeur de prêt) de comprendre et de contester les décisions automatisées. Certaines méthodes d'apprentissage automatique offrent intrinsèquement une certaine transparence. Par exemple, la régression logistique fournit une mesure quantitative de la fiabilité d'une classification à travers la valeur $|h(\mathbf{x})|$. Les arbres de décision en sont un autre exemple, car ils permettent d'utiliser des règles de décision lisibles par l'humain [70]. La transparence implique aussi de signaler clairement lorsqu'un utilisateur interagit avec un système d'IA. Par exemple, un chatbot alimenté par l'IA doit informer l'utilisateur qu'il interagit avec un système automatisé et non un humain. Enfin, la transparence suppose une documentation complète précisant l'objectif et les choix de conception du système d'IA. Des outils comme les fiches techniques de modèle [59] ou les cartes descriptives de systèmes d'IA [94] aident les praticiens à comprendre les cas d'usage prévus ainsi que les limitations du système [95].

Voir aussi: IA digne de confiance, apprentissage automatique, explicabilité, IA, prédiction, modèle, régression logistique, classification, arbre de décision.

tâche d'apprentissage Considérons un jeu de données $\mathcal{D}$ composé de plusieurs points de données $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}$. Par exemple, $\mathcal{D}$ peut représenter une collection d'images dans une base de données. Une tâche d'apprentissage est définie en spécifiant les propriétés (ou attributs) d'un point de don-

nées qui sont utilisées comme caractéristiques et étiquettes. Étant donné un choix de modèle $\mathcal{H}$ et de fonction de perte, une tâche d'apprentissage conduit à une instance de MRE et peut donc être représentée par la fonction objective associée $\widehat{L}\big(h|\mathcal{D}\big)$ pour $h \in \mathcal{H}$. Fait important, plusieurs tâches d'apprentissage distinctes peuvent être construites à partir du même jeu de données en sélectionnant différents ensembles de caractéristiques et étiquettes.

Une image de vaches pâturant dans la campagne autrichienne.

Tâche 1 (régression) :

- Les caractéristiques : les valeurs RVB de tous les pixels de l'image.
- Les étiquette : le nombre de vaches représentées.

Tâche 2 (classification) :

- Les caractéristiques : l'intensité moyenne du vert de l'image.
- Les étiquette : faut-il déplacer les vaches vers un autre endroit (oui/non) ?

Fig. 42. Deux tâches d'apprentissage construites à partir d'un seul jeu de données d'images. Ces tâches diffèrent par la sélection des caractéristiques et le choix du étiquette (c'est-à-dire l'objectif), mais sont toutes deux dérivées du même jeu de données.

Ces tâches sont intrinsèquement liées, et les résoudre conjointement, par exemple à l'aide de méthodes d'apprentissage multitâche, est souvent

plus efficace que de les traiter indépendamment [96], [97], [98].

Voir aussi: jeu de données, modèle, fonction de perte, fonction objective, apprentissage multitâche, espace des étiquettes.

valeur propre On qualifie de valeur propre d'une matrice carrée $\mathbf{A} \in \mathbb{R}^{d \times d}$ le nombre $\lambda \in \mathbb{R}$ s'il existe un vecteur non nul $\mathbf{x} \in \mathbb{R}^d \setminus \{\mathbf{0}\}$ tels que $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$.

validation Considérons une hypothèse $\hat{h}$ apprise à l'aide d'une méthode d'apprentissage automatique, par exemple en résolvant la MRE sur un ensemble d'entraînement $\mathcal{D}$. La validation désigne la pratique consistant à évaluer la perte encourue par l'hypothèse $\hat{h}$ sur un ensemble de points de données qui ne sont pas contenus dans le ensemble d'entraînement $\mathcal{D}$.

Voir aussi: hypothèse, apprentissage automatique, MRE, ensemble d'entraînement, perte, point de données.

variable aléatoire (VA) Une VA est une fonction qui associe les issues d'une expérience aléatoire à un espace de valeurs [6], [32]. Mathématiquement, une VA est une fonction $x : \Omega \to \mathcal{X}$ définie sur l'espace échantillon Ω d'un espace probabilisé. Différents types de VA existent, notamment

- les VA binaires, qui associent chaque issue à un élément d'un ensemble binaire (par exemple, $\{-1, 1\}$ ou $\{\text{chat}, \text{pas chat}\}$);
- les VA à valeurs réelles, qui prennent des valeurs dans les nombres réels $\mathbb{R}$;

- les VA vectorielles, qui associent les issues à l'espace euclidien $\mathbb{R}^d$.

La théorie des probabilités utilise le concept d'espaces mesurables pour définir rigoureusement et étudier les propriétés des collections de VA [6]. Voir aussi: fonction, espace probabilisé, probabilité, vecteur, espace euclidien.

variable aléatoire normale (VA normale) Une VA normale centrée réduite est une VA réelle x dont la fonction de densité de probabilité est donnée par [7], [32], [34]

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

À partir d'une VA normale centrée réduite x, on peut construire une VA normale générale x' de moyenne μ et de variance σ^2 via $x' := \sigma x + \mu$. La loi de probabilité d'une VA normale est appelée "loi normale", notée $\mathcal{N}(\mu, \sigma^2)$.

Un vecteur aléatoire normal $\mathbf{x} \in \mathbb{R}^d$ de matrice de covariance $\mathbf{C}$ et de moyenne $\boldsymbol{\mu}$ peut être construit comme suit [32], [34], [60]

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

où $\mathbf{z} := \left(z_1, \ldots, z_d\right)^T$ est un vecteur de VA normales centrées réduites i.i.d., plural=i.i.d., et $\mathbf{A} \in \mathbb{R}^{d \times d}$ est une matrice telle que $\mathbf{A}\mathbf{A}^T = \mathbf{C}$. La loi de probabilité d'un vecteur aléatoire normal est appelée loi normale multivariée, notée $\mathcal{N}(\boldsymbol{\mu}, \mathbf{C})$.

Nous pouvons interpréter un vecteur aléatoire normal $\mathbf{x} = \left(x_1, \ldots, x_d\right)$ comme un processus stochastique indexé par l'ensemble $\mathcal{I} = \{1, \ldots, d\}$.

Un processus gaussien est un processus stochastique défini sur un ensemble d'indices arbitraire $\mathcal{I}$, tel que toute restriction à un sous-ensemble fini $\mathcal{I}' \subseteq \mathcal{I}$ donne un vecteur aléatoire normal [99].

Les VA normales sont largement utilisées comme modèles probabilistes dans l'analyse statistique des méthodes d'apprentissage automatique. Leur importance provient en partie du théorème central limite (TCL), qui est une formulation mathématique précise de la règle empirique suivante : La moyenne de nombreuses VA indépendantes (pas nécessairement normales) tend vers une VA normale [33].

La loi normale multivariée est également remarquable en ce qu'elle représente l'incertitude maximale : parmi toutes les VA vectorielles ayant une matrice de covariance $\mathbf{C}$ donnée, la VA $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{C})$ maximise l'entropie différentielle [30, Th. 8.6.5]. Cela fait des processus gaussiens un choix naturel pour modéliser l'incertitude (ou le manque d'information) en l'absence de structure supplémentaire.

Voir aussi: loi normale multivariée, processus gaussien, modèle probabiliste, TCL, entropie différentielle.

variance La variance d'une VA réelle x est définie comme l'espérance $\mathbb{E}\big\{\big(x - \mathbb{E}\{x\}\big)^2\big\}$ de la différence au carré entre x et son espérance $\mathbb{E}\{x\}$. On étend cette définition aux VA vectorielles $\mathbf{x}$ avec $\mathbb{E}\big\{\big\|\mathbf{x} - \mathbb{E}\{\mathbf{x}\}\big\|_2^2\big\}$.

Voir aussi: VA, espérance.

variation totale généralisée (GTV) La variation totale généralisée (GTV) est une mesure de la variation des modèles locaux entraînés $h^{(i)}$ (ou leurs paramètres de modèle $\mathbf{w}^{(i)}$) assignés aux sommets $i = 1, \ldots, n$ d'un

graphe pondéré non orienté $\mathcal{G}$ avec arêtes $\mathcal{E}$. Étant donnée une mesure $d^{(h,h')}$ de la divergence (ou écart) entre deux fonctions hypothèses h, h', la GTV est définie par

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

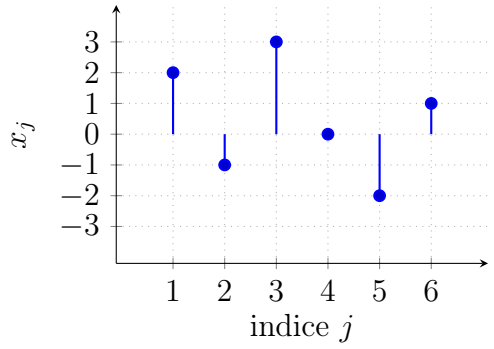
Ici, $A_{i,i'} > 0$ désigne le poids de l'arête non orientée $\{i, i'\} \in \mathcal{E}$.

Voir aussi : modèle local, paramètres du modèle, graphe, divergence, hypothèse, application.

vecteur Un vecteur est un élément d'un espace vectoriel. Dans le contexte de l'apprentissage automatique, un exemple particulièrement important d'espace vectoriel est l'espace euclidien $\mathbb{R}^d$, où $d \in \mathbb{N}$ est la dimension (finie) de l'espace. Un vecteur $\mathbf{x} \in \mathbb{R}^d$ peut être représenté comme une liste ou un tableau unidimensionnel de nombres réels, c'est-à-dire $x_1, \ldots, x_d$ avec $x_j \in \mathbb{R}$ pour $j = 1, \ldots, d$. La valeur x_j est la j-ième entrée du vecteur $\mathbf{x}$. Il peut également être utile de voir un vecteur $\mathbf{x} \in \mathbb{R}^d$ comme une fonction qui associe à chaque indice $j \in \{1, \ldots, d\}$ une valeur $x_j \in \mathbb{R}$, c'est-à-dire $\mathbf{x} : j \mapsto x_j$. Cette perspective est particulièrement utile pour l'étude des méthodes à noyau.

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

(a)



(b)

Fig. 43. Deux représentations équivalentes du vecteur $\mathbf{x} = \big(2, -1, 3, 0, -2, 1\big)^T \in \mathbb{R}^6$. (a) Comme un tableau numérique. (b) Comme une application $j \mapsto x_j$.

Voir aussi : espace euclidien, espace vectoriel, application linéaire.

vecteur de caractéristiques Un vecteur de caractéristiques est un vecteur $\mathbf{x} = \big(x_1, \ldots, x_d\big)^T$ dont les composantes sont des caractéristiques individuelles $x_1, \ldots, x_d$. De nombreuses méthodes d'apprentissage automatique utilisent des vecteurs de caractéristiques appartenant à un espace euclidien de dimension finie $\mathbb{R}^d$. Cependant, pour certaines méthodes d'apprentissage automatique, il peut être plus pratique de travailler avec des vecteurs de caractéristiques appartenant à un espace vectoriel de dimension infinie (par exemple, voir la méthode à noyau).

Voir aussi: caractéristique, apprentissage automatique, espace euclidien, espace vectoriel, méthode à noyau.

vecteur normal centré réduit Un vecteur normal centré réduit est un

vecteur aléatoire $\mathbf{x} = \big(x_1, \ldots, x_d\big)^T$ dont les composantes sont des VA normales i.i.d., plural=i.i.d. $x_j \sim \mathcal{N}(0,1)$. Il s'agit d'un cas particulier de loi normale multivariée, $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

Voir aussi: i.i.d., plural=i.i.d., VA normale, loi normale multivariée, VA.

voisinage Le voisinage d'un sommet $i \in \mathcal{V}$ est le sous-ensemble de sommets constitué des voisins de i.

Voir aussi: voisins.

voisins Les voisins d'un sommet $i \in \mathcal{V}$ dans un réseau d'apprentissage fédéré sont les sommets $i' \in \mathcal{V} \setminus \{i\}$ qui sont connectés (via une arête) au sommet i.

Voir aussi: réseau d'apprentissage fédéré.

écart de généralisation L'écart de généralisation est la différence entre la performance d'un modèle entraîné sur l'ensemble d'entraînement $\mathcal{D}^{(\text{train})}$ et sa performance sur des points de données extérieurs à $\mathcal{D}^{(\text{train})}$. Cette notion peut être précisée en utilisant un modèle probabiliste permettant de calculer le risque d'un modèle entraîné comme l'espérance de la perte. Toutefois, la loi de probabilité sous-jacente à cette espérance est généralement inconnue et doit être estimée d'une manière ou d'une autre. Les techniques de validation utilisent différentes constructions pour l'ensemble de validation, différent du ensemble d'entraînement, pour estimer l'écart de généralisation.

Voir aussi : validation, généralisation, MRE, fonction de perte.

échantillon Une séquence (ou liste) finie de points de données $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}$,

obtenu ou interprété comme la réalisation de m VA i.i.d., plural=i.i.d.s suivant une même loi de probabilité $p(\mathbf{z})$. La longueur m de la séquence est appelée taille d'échantillon.

Voir aussi: point de données, réalisation, i.i.d., plural=i.i.d., VA, loi de probabilité, taille d'échantillon.

épigraphe L'épigraphe d'une fonction à valeurs réelles $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ est l'ensemble des points situés sur sa courbe ou au dessus:

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

Une fonction est convexe si et seulement si son épigraphe est un ensemble convexe [21], [100].

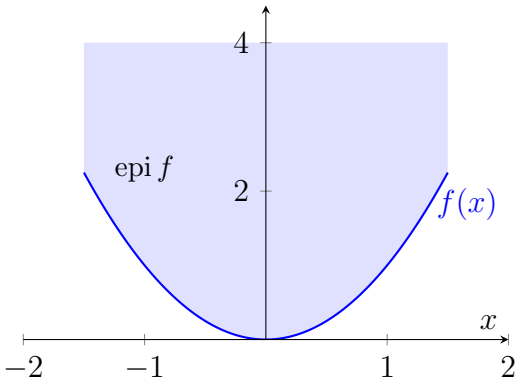


Fig. 44. Épigraphe de la fonction $f(x) = x^2$ (i.e., la zone colorée).

Voir aussi: fonction, convexe.

étiquette Une étiquette est un fait ou une quantité d'intérêt de plus haut niveau associée à un point de données. Par exemple, si le point de données est une image, l'étiquette peut indiquer si l'image contient un

chat ou non. Les synonymes de « étiquette », couramment utilisés dans certains domaines, incluent « variable réponse », « variable de sortie » et « cible » [18], [19], [20].

Voir aussi: point de données.

événement Considérons une VA $\mathbf{x}$, définie sur un espace probabilisé $\mathcal{P}$, qui prend ses valeurs dans un espace mesurable $\mathcal{X}$. Un événement $\mathcal{A} \subseteq \mathcal{X}$ est un sous-ensemble de $\mathcal{X}$ tel que la probabilité $\mathbb{P}(\mathbf{x} \in \mathcal{A})$ est bien définie. En d'autres termes, l'image réciproque $\mathbf{x}^{-1}(\mathcal{A})$ d'un événement appartient à la tribu de $\mathcal{P}$.

Voir aussi : point de données, hypothèse i.i.d., VA, modèle probabiliste.

Index

References

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

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

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

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

[5] 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] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. Cambridge, MA, USA: MIT Press, 2022. [Online]. Available: http://ebookcentral.proquest.com/lib/aalto-ebooks/detail.action?docID=6925615

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

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

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

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

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

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

[16] R. Sutton and A. Barto, *Reinforcement learning: An introduction*, 2nd ed. Cambridge, MA: MIT press, 2018.

[17] 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.

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

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

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

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

[22] 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

[23] 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.

[24] 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.

[25] 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/

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

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

[28] 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.

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

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

[31] N. Young, *An Introduction to Hilbert Space*. New York, NY, USA: Cambridge Univ. Press, 1988.

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

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

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

[35] R. B. Ash, *Probability and Measure Theory*, 2nd ed. New York: Academic Press, 2000.

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

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

[38] 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

[39] 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.

[40] 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.

[41] 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

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

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

[44] 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.

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

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

[47] 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.

[48] 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.

[49] 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

[50] 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

[51] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," Mar. 2017. [Online]. Available: https://arxiv.org/abs/1702.08608

[52] 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

[53] 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.

[54] 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.

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

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

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

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

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

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

[61] A. Lapidoth, *A Foundation in Digital Communication*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2017.

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

[63] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge, UK: Cambridge Univ. Press, 1991.

[64] 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.

[65] 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

[66] R. Durrett, *Probability: Theory and Examples*, 4th ed. Cambridge: Cambridge University Press, 2010.

[67] W. Rudin, *Real and Complex Analysis*, international edition ed. McGraw-Hill Book Co., Singapore, 1987.

[68] 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.

[69] 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.

[70] 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.

[71] 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.

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

[73] 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

[74] 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

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

[76] 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.

[77] V. I. Istrățescu, *Fixed Point Theory: An Introduction*. Dordrecht, The Netherlands: D. Reidel, 1981.

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

[79] 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.

[80] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector*

Machines, Regularization, Optimization, and Beyond. Cambridge, MA, USA: MIT Press, 2002.

[81] 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.

[82] 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.

[83] 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

[84] 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

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

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

[87] A. Ben-Israel and T. N. E. Greville, *Generalized Inverses: Theory and Applications*, 2nd ed. New York, NY, USA: Springer-Verlag, 2003.

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

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

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

[91] A. van der Vaart, *Asymptotic Statistics*. Cambridge, U.K.: Cambridge Univ. Press, 1998.

[92] 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

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

[94] M. Mitchell et al., "Model cards for model reporting," in *Proc.*

Conf. Fairness, Accountability, Transparency, 2019, pp. 220–229, doi: 10.1145/3287560.3287596.

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

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

[97] 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.

[98] 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.

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

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