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 15, 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 $0 \leq x \leq 1$.

$\underset{\mathbf{w}}{\operatorname{argmin}} f(\mathbf{w})$	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.
$\exp(a)$	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.

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.
$\mathbb{R}^d$	L'ensemble des vecteurs $\mathbf{x} = \big(x_1, \ldots, x_d\big)^T$ constitués de d composantes réelles $x_1, \ldots, x_d \in \mathbb{R}$.
$\mathbf{I}_{l\times d}$	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
$\|\mathbf{x}\|$	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
$\mathbf{x}^T$	La transposée d'une matrice ayant pour unique colonne le vecteur $\mathbf{x} \in \mathbb{R}^d$.
$\mathbf{X}^T$	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.
$\mathbf{1} = \big(1, \ldots, 1\big)^T$	Le vecteur de $\mathbb{R}^d$ dont toutes les composantes valent 1.
$\big(\mathbf{v}^T, \mathbf{w}^T\big)^T$	Le vecteur de longueur $d + d'$ obtenu en concaténant les $\mathbf{v} \in \mathbb{R}^d$ avec celles de $\mathbf{w} \in \mathbb{R}^{d'}$.
$\text{span}\{\mathbf{B}\}$	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 $\mathbf{B}$: $\text{span}\{\mathbf{B}\} = \big\{\mathbf{Ba} : \mathbf{a} \in \mathbb{R}^b\big\} \subseteq \mathbb{R}^a$.
$\det(\mathbf{C})$	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

$\mathbf{x} \sim p(\mathbf{z})$	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.
$\mathcal{N}(\mu,\sigma^2)$	La loi de probabilité d'une variable aléatoire normale centrée réduite (VA normale centrée réduite) $x \in \mathbb{R}$ 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 centrée réduite, moyenne, espérance, variance, loi de probabilité
$\mathcal{N}(\boldsymbol{\mu},\mathbf{C})$	La loi normale multivariée d'une VA normale centrée réduite 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 centrée réduite, moyenne, matrice de covariance

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.
$\mathbf{x}^{(r)}$	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

$h^{(\mathbf{w})}(\cdot)$	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
$\phi(\cdot)$	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 nœuds $i \in \mathcal{V}$ 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 nœud 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 nœuds 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 d'un nœud 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 nœud i dans un graphe $\mathcal{G}$. Voir aussi : graphe.
$d_{\max}^{(\mathcal{G})}$	Le degré pondéré maximal (parmi les degrés pondérés des nœuds) d'un graphe $\mathcal{G}$. Voir aussi : maximum, degré, graphe.
$\mathcal{D}^{(i)}$	Le jeu de données local $\mathcal{D}^{(i)}$ détenu par le nœud $i \in \mathcal{V}$ 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 nœud $i \in \mathcal{V}$. 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

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 :

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

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.

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.

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 [8,13]. 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 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.

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. En apprentissage incrémental, l'hypothèse (ou les paramètres du modèle) est mise à jour de manière incrémentale à chaque nouvelle observation, sans avoir besoin de retraiter les données précédentes.

Voir aussi : apprentissage automatique, données, paramètres du modèle, point de données, FMI, hypothèse, 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.

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 nœud 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 nœuds enfants en fonction d'un test élémentaire sur les caractéristiques de $\mathbf{x}$. Si le nœud 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 nœuds descendants. Ce processus de test et de transfert est répété jusqu'à ce que le point de données atteigne une feuille (un nœud sans enfant).

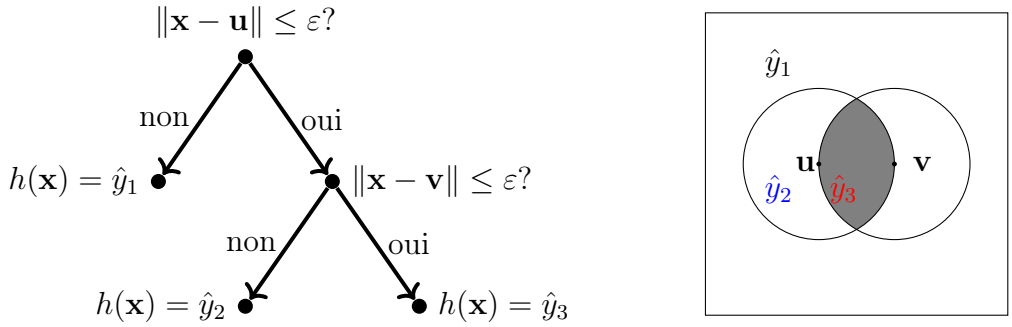


Fig. 1. À gauche : un arbre de décision est une représentation en organigramme d'une hypothèse $h : \mathcal{X} \to \mathbb{R}$ constante par morceaux. Chaque morceau correspond à une région de décision $\mathcal{R}_{\hat{y}} := \{\mathbf{x} \in \mathcal{X} : h(\mathbf{x}) = \hat{y}\}$. L'arbre de décision illustré s'applique à des vecteurs de caractéristiques numériques, i.e., $\mathcal{X} \subseteq \mathbb{R}^d$. Il est paramétré par un seuil $\varepsilon > 0$ et des vecteurs $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$. À droite : un arbre de décision partitionne l'espace des caractéristiques $\mathcal{X}$ en régions de décision. Chaque région $\mathcal{R}_{\hat{y}} \subseteq \mathcal{X}$ correspond à une feuille particulière de l'arbre.

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) ; 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.

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.

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 2). L'augmentation de données peut être une forme efficace de régularisation.

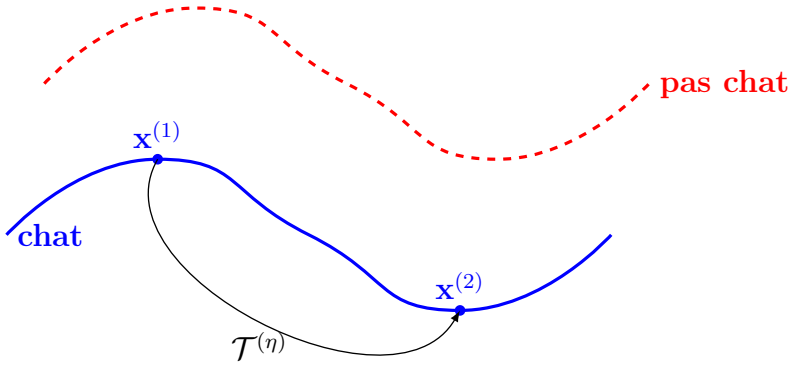


Fig. 2. L'augmentation de données exploite les symétries intrinsèques des points de données dans un certain espace des caractéristiques $\mathcal{X}$. On peut représenter une symétrie par un opérateur $\mathcal{T}^{(\eta)} : \mathcal{X} \to \mathcal{X}$, paramétré par un nombre $\eta \in \mathbb{R}$. 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 $\mathbf{x}^{(1)}$.

bandit manchot Le problème du bandit manchot modélise un scénario de prises de décision répétées dans lequel, à chaque instant k, un apprenant doit choisir une action parmi plusieurs possibles (on appellera ces actions des bras) dans un ensemble fini $\mathcal{A}$. Chaque bras $a \in \mathcal{A}$ génère une récompense stochastique $r^{(a)}$ prélevée selon une loi de probabilité inconnue, de moyenne $\mu^{(a)}$. Le but de l'apprenant est de maximiser la somme des récompenses au cours du temps en équilibrant stratégiquement l'exploration (collecte d'informations sur les bras incertains) et l'exploitation (choix des bras connus comme performants). Cet équilibre est quantifié par la notion de regret, qui mesure l'écart de performance en-

tre la stratégie de l'apprenant et la stratégie optimale qui sélectionnerait toujours le meilleur bras. Les problèmes de bandit manchot constituent un modèle fondamental en apprentissage incrémental, apprentissage par renforcement et en conception expérimentale séquentielle [14].

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 $B^2 := \left\|\mathbb{E}\{\widehat{\mathbf{w}}\} - \overline{\mathbf{w}}\right\|_2^2$.

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 » [15], [16], [17].

Voir aussi: point de données.

classification La classification est la tâche qui consiste à déterminer une étiquette discrète y pour un point de données donné, 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.

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 d'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.

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].

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 [18]. On illustre un exemple d'ensemble convexe et de fonction convexe dans la Figure 3.

Fig. 3. Gauche: Un ensemble convexe $\mathcal{C} \subseteq \mathbb{R}^d$. Droite: Une fonction convexe $f : \mathbb{R}^d \to \mathbb{R}$.

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

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

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 4 pour des exemples visuels.

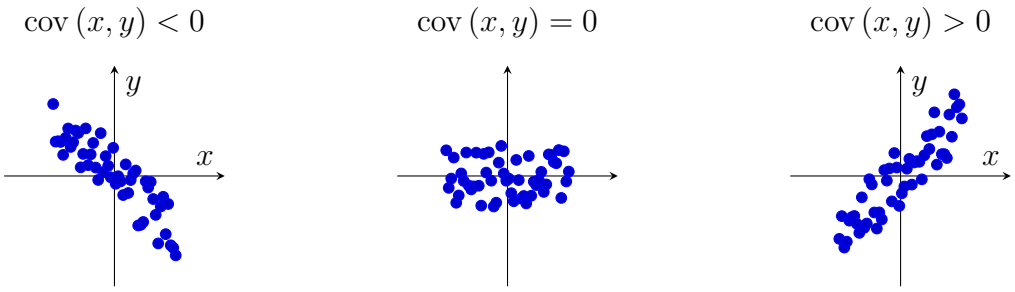


Fig. 4. Nuage de pointss illustrant des réalisations issues de trois modèles probabilistes différents pour deux VA avec des valeurs de covariance négative (gauche), nulle (centre) et positive (droite).

Voir aussi: modèle probabiliste, espérance.

degré d'un noeud Le degré $d^{(i)}$ d'un nœud $i \in \mathcal{V}$ dans un graphe non orienté est le nombre de voisins de ce nœud, 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 pour trouver le minimum d'une fonction $f(\mathbf{w})$ d'un argument vectoriel $\mathbf{w} \in \mathbb{R}^d$ dérivable. Considérons une estimation actuelle ou approximation $\mathbf{w}^{(k)}$ du minimum de la fonction $f(\mathbf{w})$. Nous souhaitons trouver un nouveau (et meilleur) vecteur $\mathbf{w}^{(k+1)}$ ayant une valeur objective

inférieure $f(\mathbf{w}^{(k+1)}) < f\big(\mathbf{w}^{(k)}\big)$ que l'estimation actuelle $\mathbf{w}^{(k)}$. Nous pouvons généralement y parvenir en utilisant un pas

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

avec une taille de pas $\eta > 0$ suffisamment petite. La figure 5 illustre l'effet d'un seul pas de descente de gradient (2).

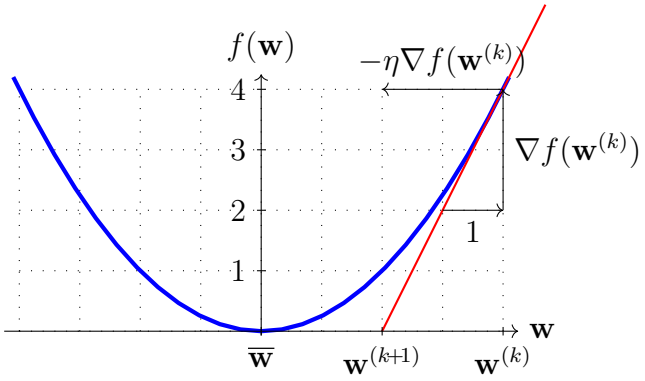


Fig. 5. Un seul pas (2) vers le minimiseur $\overline{\mathbf{w}}$ de $f(\mathbf{w})$.

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.

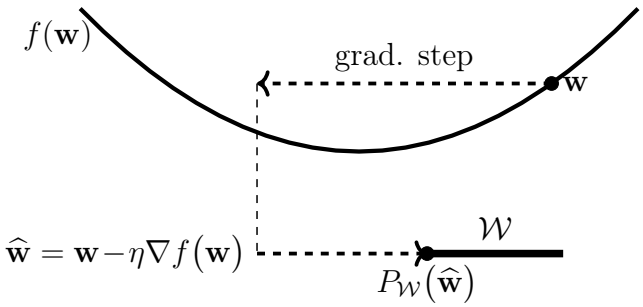


Fig. 6. La descente de gradient avec projection complète un pas classique avec une projection sur l'ensemble de contraintes $\mathcal{W}$.

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. 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 données. 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, 19].

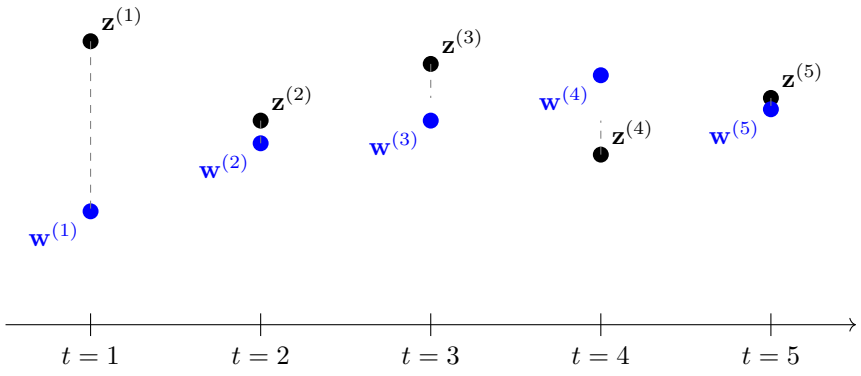


Fig. 7. Un exemple de descente de gradient en ligne qui met à jour les paramètres du modèle $\mathbf{w}^{(t)}$ en utilisant le point de données $\mathbf{z}^{(t)} = x^{(t)}$ arrivant à l'instant t. Cet exemple utilise la perte quadratique $L\left(\mathbf{z}^{(t)}, w\right) = (x^{(t)} - w)^2$.

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. 8). 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 [20].

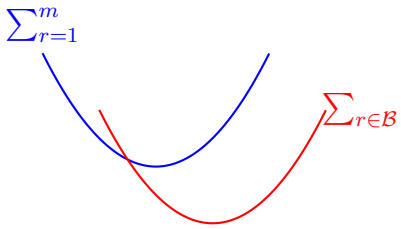


Fig. 8. La descente de gradient stochastique pour la MRE approxime le gradient $\sum_{r=1}^{m} \nabla_{\mathbf{w}} L\left(\mathbf{z}^{(r)}, \mathbf{w}\right)$ en remplaçant la somme sur tous les points de données de l'ensemble d'entraînement (indexés par $r = 1, \ldots, m$) par une somme sur un sous-ensemble 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.

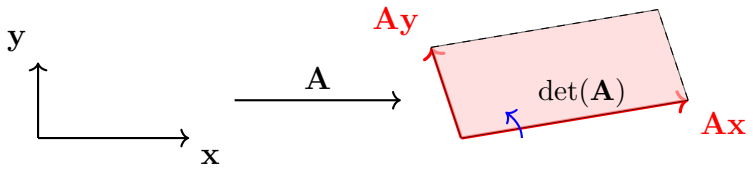
données Les données désignent des objets porteurs d'information. Ces objets peuvent être des entités physiques concrètes (comme des personnes ou des animaux), ou des concepts abstraits (comme des nombres). On utilise souvent des représentations (ou des approximations) des données originales, plus pratiques pour le traitement. Ces approximations reposent sur différents modèles de données, le modèle relationnel étant l'un des plus utilisés [21].

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], [22]. 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 $\det(\mathbf{A}) = \prod_{i=1}^{n} \lambda_i$ [23]. 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.

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].

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.

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).

entropie L'entropie quantifie l'incertitude ou l'imprévisibilité associée à une VA [24]. 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 [24] :

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

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 [24]. 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.

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.

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.

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.

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 9). 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.

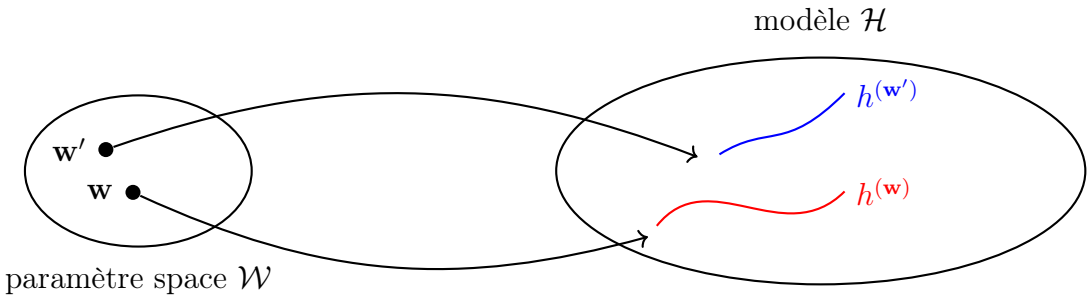


Fig. 9. L'espace des paramètres $\mathcal{W}$ d'un modèle d'apprentissage automatique $\mathcal{H}$ contient tous les choix possibles pour les paramètres du modèle. Chaque choix $\mathbf{w}$ pour les paramètres du modèle sélectionne une hypothèse $h^{(\mathbf{w})} \in \mathcal{H}$.

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 .

See also: 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 un modèle mathématique d'un processus physique (une expérience aléatoire) avec un résultat incertain. Formellement, un espace probabilisé $\mathcal{P}$ est un triplet $(\Omega, \mathcal{F}, P)$ où

- Ω est un espace échantillon contenant tous les résultats élémentaires possibles d'une expérience aléatoire ;
- $\mathcal{F}$ est une tribu (ou sigma-algèbre), une collection de sous-ensembles de Ω (appelés événements) qui satisfait certaines propriétés de fermeture par opérations sur les ensembles ;
- P est une mesure de probabilité, une fonction qui attribue une probabilité $P(\mathcal{A}) \in [0, 1]$ à chaque événement $\mathcal{A} \in \mathcal{F}$. Cette fonction doit satisfaire $P(\Omega) = 1$ et

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

pour toute suite dénombrable d'événements deux à deux disjoints $\mathcal{A}_1, \mathcal{A}_2, \ldots$ dans $\mathcal{F}$.

Les espaces probabilisés fournissent la base pour définir les VA et raisonner sur incertitude dans les applications d'apprentissage automatique [6, 25, 26].

Voir aussi: probabilité, modèle, échantillon, fonction, VA, incertitude, apprentissage automatique.

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 application linéaires opèrent dans de tels espaces.
Voir aussi : espace euclidien, modèle linéaire, application linéaire.

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], [27]. La figure 10 illustre l'espérance d'une VA discrète scalaire x prenant ses valeurs dans un ensemble fini.

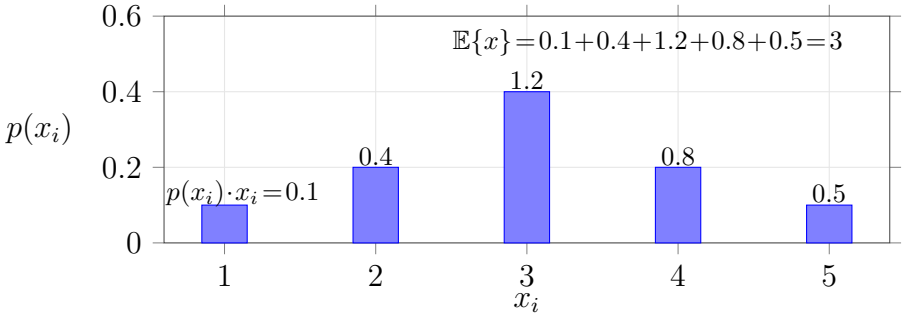


Fig. 10. L'espérance d'une VA discrète x s'obtient en sommant les valeurs possibles x_i, pondérées par leur probabilité correspondante $p(x_i) = \mathbb{P}(x = x_i)$.

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 [28]. 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 $L(\cdot, \cdot)$.

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 de regret cherchent à apprendre une hypothèse dont la perte est comparable à celle du meilleur expert [11, 12].

explicabilité On définit l'explicabilité (subjective) d'une méthode d'apprentissage automatique comme le niveau de simulabilité [29] 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 [29, 30]. 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 conditionnelle de ses prédictions, étant donné les prédictions de l'utilisateur [31, 32].

explication Une approche pour rendre les méthodes d'apprentissage automatique plus transparentes consiste à fournir une explication en complément de la prédiction produite par une méthode d'apprentissage automatique. Les explications peuvent prendre différentes formes. Il peut s'agir d'un texte en langage naturel ou d'une mesure quantitative indiquant l'importance des différentes caractéristiques d'un point de

données [33]. On peut aussi utiliser des formes visuelles d'explication, comme des cartes d'intensité pour la classification d'images [34].

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 $\mathbf{x}^{(0)}$ [35]. L'explication est donnée sous la forme d'une approximation locale $g \in \mathcal{H}'$ de $\widehat{h}$ (voir Fig. **??**). Cette 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.

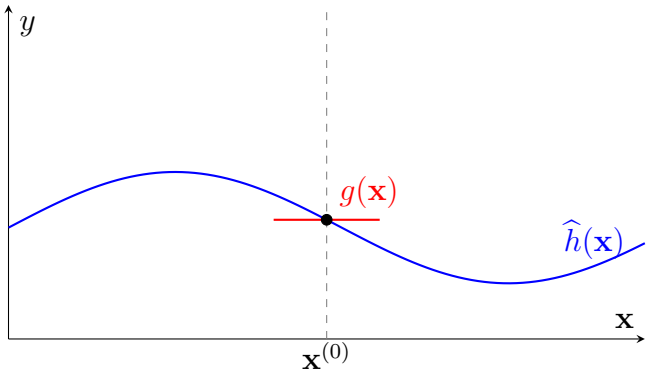


Fig. 11. Pour expliquer (comprendre) un modèle entraîné $\widehat{h} \in \mathcal{H}$, autour d'une vecteur de caractéristiques donnée $\mathbf{x}^{(0)}$, on peut utiliser une approximation locale $g \in \mathcal{H}'$.

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

fonction Une fonction est une règle mathématique qui associe à chaque élément $u \in \mathcal{U}$ exactement un élément $v \in \mathcal{V}$ [2]. On écrit cela $f : \mathcal{U} \to \mathcal{V}$, 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 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 $w_1, \ldots, w_d$.

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 $\mathbb{P}(\mathbf{x} \in \mathcal{R}) = \int_{\mathcal{R}} p(\mathbf{x}')dx'_1 \dots dx'_d$ [7, Ch. 3].

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 12 représente une fonction de perte pour un point de données donné, de caractéristiques $\mathbf{x}$ et d'étiquette y, en fonction de l'hypothèse $h \in \mathcal{H}$.

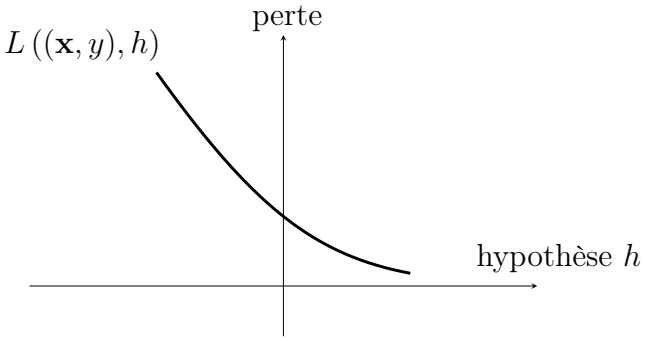


Fig. 12. Une fonction de perte $L\left((\mathbf{x}, y), h\right)$ pour un point de données fixé, de vecteur de caractéristiques $\mathbf{x}$ et d'étiquette y, et une hypothèse variable h. Les méthodes d'apprentissage automatique cherchent à trouver (ou apprendre) une hypothèse minimisant la perte.

fonction objective Une fonction objective est une application qui associe à chaque valeur d'une variable d'optimisation, comme les paramètres du modèle $\mathbf{w}$ d'une hypothèse $h^{(\mathbf{w})}$, une valeur objective $f(\mathbf{w})$. La valeur objective $f(\mathbf{w})$ peut être le risque ou le risque empirique d'une hypothèse $h^{(\mathbf{w})}$.

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.

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.

graphe Un graphe $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ est une paire qui consiste en un ensemble de nœuds $\mathcal{V}$ 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 nœuds [36]. 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 nœuds différents $\{i, i'\}$. Les graphes pondérés précisent également des poids numériques A_e pour chaque arête $e \in \mathcal{E}$.

généralisation Beaucoup de systèmes d'apprentissage automatique (et d'intelligence artificielle (IA)) actuels reposent sur la MRE : au noyau, ils entraînent un modèle (c'est-à-dire apprennent une hypothèse $\hat{h} \in \mathcal{H}$) en minimisant la perte moyenne (ou risque empirique) sur des points de données $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}$, qui constituent un ensemble d'entraînement $\mathcal{D}^{(\text{train})}$. La généralisation désigne la capacité d'une méthode d'apprentissage automatique à bien fonctionner en dehors de cet ensemble d'entraînement. Toute théorie mathématique sur la généralisation nécessite une notion mathématique du « en dehors de l'ensemble d'entraînement ». Par exemple, la théorie statistique de l'apprentissage utilise un modèle probabiliste tel que l'hypothèse i.i.d. pour la génération des données : les

points de données du ensemble d'entraînement sont des réalisations i.i.d. d'une loi de probabilité sous-jacente $p(\mathbf{z})$. Un tel modèle probabiliste permet d'explorer ce qui est en dehors de l'ensemble d'entraînement en tirant d'autres réalisations i.i.d. selon $p(\mathbf{z})$. De plus, grâce à l'hypothèse i.i.d., on peut définir le risque d'un modèle entraîné $\hat{h} \in \mathcal{H}$ comme la perte espérée $\bar{L}\big(\hat{h}\big)$. On peut également utiliser des bornes de concentration ou des résultats de convergence pour des suites de VA i.i.d. afin de borner l'écart entre le risque empirique $\widehat{L}\big(\hat{h}|\mathcal{D}^{(\text{train})}\big)$ d'un modèle entraîné et son risque [37]. Il est aussi possible d'étudier la généralisation sans utiliser de modèle probabiliste. Par exemple, on peut considérer des perturbations (déterministes) des points de données de l'ensemble d'entraînement pour étudier ce qui est en dehors. En général, on souhaite que le modèle entraîné soit robuste, c'est-à-dire que ses prédictionss ne changent pas trop pour de petites perturbations des points de données. Par exemple, pour un modèle entraîné à détecter un objet dans une photo prise avec un smartphone, le résultat de la détection ne devrait pas changer si l'on masque un petit nombre de pixels choisis aléatoirement dans l'image [38].

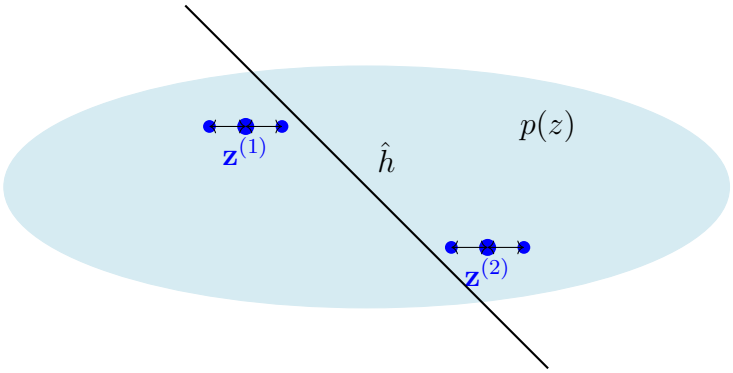


Fig. 13. Deux points de données $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}$ utilisés comme ensemble d'entraînement pour apprendre une hypothèse $\hat{h}$ via MRE. On peut évaluer $\hat{h}$ en dehors de l'ensemble d'entraînement soit avec l'hypothèse i.i.d. avec une loi de probabilité sous-jacente $p(\mathbf{z})$, soit en perturbant les points de données.

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).

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

incertitude L'incertitude désigne le degré de confiance — ou de manque de confiance — associé à une quantité comme une prédiction de modèle,

une estimation de paramètre ou une observation de point de données. En apprentissage automatique, l'incertitude provient de diverses sources, comme des données bruitées, un nombre limité d'échantillons d'entraînement, ou une ambiguïté dans les hypothèses du modèle. La théorie des probabilités fournit un cadre rigoureux pour représenter et quantifier cette incertitude.

indépendantes et identiquement distribuées (i.i.d.) Il peut être utile d'interpréter des points de données $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}$ comme des réalisations de VA i.i.d. suivant une loi de probabilité commune. Si ces VA sont à valeurs continues, leur fonction de densité de probabilité conjointe est $p\big(\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}\big) = \prod_{r=1}^{m} p\big(\mathbf{z}^{(r)}\big)$, où $p(\mathbf{z})$ est la fonction de densité de probabilité marginale commune des VA sous-jacentes (i.e., dont les points de données sont les réalisations).

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

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.

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 [39]. L'Union européenne a proposé sept exigences clés pour une IA digne de confiance (généralement basée sur des méthodes d'apprentissage automatique) [40] :

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.

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

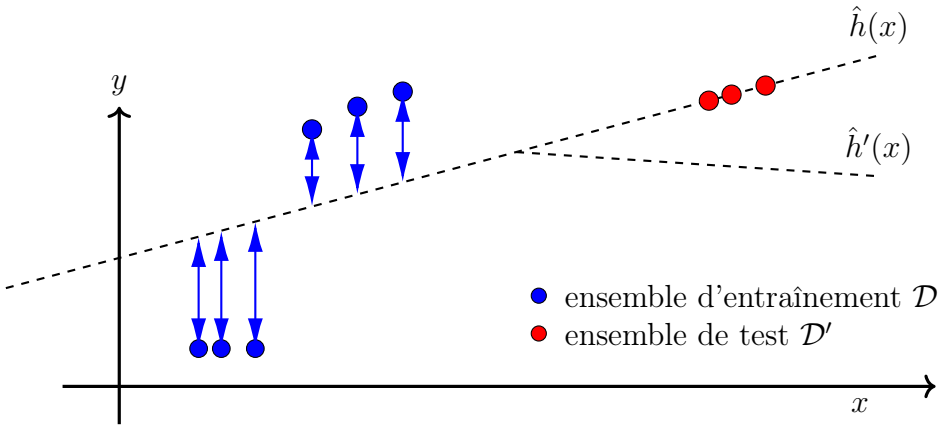


Fig. 14. Nous pouvons évaluer l'interprétabilité des modèles d'apprentissage automatique entraînés $\hat{h}$ et $\hat{h}'$ en comparant leurs prédictions aux pseudo-étiquettes générées par un utilisateur humain pour $\mathcal{D}'$.

capacité d'un humain à simuler mentalement le comportement du modèle [29, 32, 41–43]. L'idée est la suivante : si un utilisateur humain 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. 14 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 14, 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, intelligence artificielle digne de confiance (IA digne de confiance), régularisation, LIME.

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 de l'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 **??** illustre un jeu de données utilisant des vaches comme points de données.

Fig. 15. 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 **??** 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 [44], [45], [46], [47].
L'un des modèles de données les plus utilisés est le modèle relationnel, qui organise les données sous forme de tableau (ou relation) [21], [44]. 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 **??**. 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 **??**.

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 [48].

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 une 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.

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. 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, 25].

Voir aussi: 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. 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 [49].

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], [25], [50]. 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 $\mathbf{A}\mathbf{A}^\top = \boldsymbol{\Sigma}$. Cette représentation reste valable même lorsque $\boldsymbol{\Sigma}$ est singulière, auquel cas $\mathbf{A}$ n'est pas de plein rang [51, 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 :

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

Deuxièmement, la loi de probabilité $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ maximise l'entropie différentielle parmi toutes les distributions ayant la même matrice de covariance $\boldsymbol{\Sigma}$ [24].

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

lot Dans le contexte de la descente de gradient stochastique (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.

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 $\mathbf{X} := \big(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}\big)^T$ de taille $m \times d$.

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

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 [22], [52]. 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 $\mathbf{B}\mathbf{A} = \mathbf{I}$, ou une « inverse à droite » $\mathbf{C}$ telle que $\mathbf{A}\mathbf{C} = \mathbf{I}$. Pour les matrices rectangulaires ou singulières, la pseudo-inverse de Moore–Penrose, notée $\mathbf{A}^+$, fournit une généralisation unifiée de la matrice inverse [3].

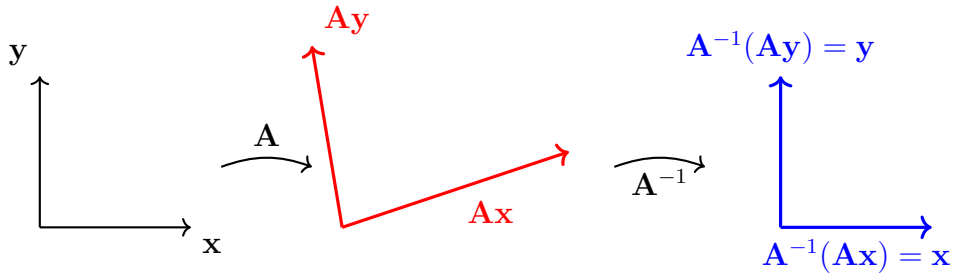


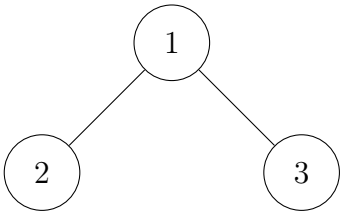
Fig. 16. Une matrice $\mathbf{A}$ représente une transformation linéaire de $\mathbb{R}^2$. La matrice inverse $\mathbf{A}^{-1}$ représente la transformation inverse.

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

matrice laplacienne La structure d'un graphe $\mathcal{G}$, avec pour nœuds $i = 1, \ldots, n$, 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 [53, 54]. Elle est définie terme à terme par (voir Figure 17)

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

Ici, $A_{i,i'}$ désigne le poids d'arête 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. 17. À gauche : Un graphe non orienté $\mathcal{G}$ avec trois nœuds $i = 1, 2, 3$. À droite : La matrice laplacienne $\mathbf{L}^{(\mathcal{G})} \in \mathbb{R}^{3\times 3}$ de $\mathcal{G}$.

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 son borne supérieure [2, Sec. 1.4].

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. avec une loi de probabilité commune $\mathbb{P}(\mathbf{z}; \mathbf{w})$ qui dépend des paramètres du modèles $\mathbf{w} \in \mathcal{W} \subseteq \mathbb{R}^n$. Les méthodes de maximum de vraisemblance apprennent les paramètres du modèles $\mathbf{w}$ en maximisant 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 $\max_{\mathbf{w}\in\mathcal{W}} \mathbb{P}(\mathcal{D}; \mathbf{w})$.

minimisation du risque empirique (MRE) La minimisation du risque empirique (MRE) (risque empirique) 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: minimum, risque empirique, hypothèse, modèle, perte, jeu de données, ensemble d'entraînement, apprentissage automatique.

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], [37]. 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.

modèle linéaire Considérons des points de données, chacun étant caractérisé par un vecteur de caractéristiques numérique $\mathbf{x} \in \mathbb{R}^d$. Un modèle linéaire est un espace des hypothèses constitué de toutes les applications linéaires,

$$\mathcal{H}^{(d)} := \left\{ h(\mathbf{x}) = \mathbf{w}^T \mathbf{x} : \mathbf{w} \in \mathbb{R}^d \right\}. \tag{5}$$

Notons que (5) définit une famille entière d'espace des hypothèsess, paramétrée par le nombre d de caractéristiques qui sont combinées linéairement pour former la prédiction $h(\mathbf{x})$. Le choix de d est guidé par les aspects computationnels (par exemple, réduire d signifie moins de calcul), les aspects statistiques (par exemple, augmenter d peut réduire l'erreur de prédiction) et l'interprétabilité. Un modèle linéaire utilisant

peu de caractéristiques soigneusement sélectionnées a tendance à être considéré comme plus interprétable [35, 55].

modèle local Considérons une collection de jeux de données locaux qui sont assignés aux nœuds d'un réseau d'apprentissage fédéré. Un modèle local $\mathcal{H}^{(i)}$ est un espace des hypothèses assigné à un nœud $i \in \mathcal{V}$. Différents nœuds peuvent se voir assigner des espace des hypothèsess différents, c'est-à-dire qu'en général $\mathcal{H}^{(i)} \neq \mathcal{H}^{(i')}$ pour des nœuds différents $i, i' \in \mathcal{V}$.

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 [28].

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,

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

voir par exemple [6] ou [2]. Nous utilisons également 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 une VA discrète $\widetilde{\mathbf{x}} = \mathbf{x}^{(I)}$ 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 empirique $\frac{1}{m} \sum_{r=1}^{m} \mathbf{x}^{(r)}$.

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 $\mathbf{z} = K\big(\mathbf{x}, \cdot\big)$ [56, 57]. 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 18).

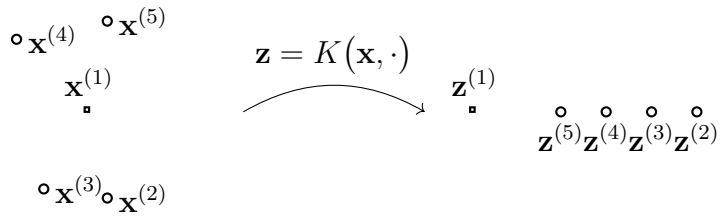


Fig. 18. Cinq points de données caractérisés par des vecteurs de caractéristiques $\mathbf{x}^{(r)}$ et étiquettes $y^{(r)} \in \{\circ, \square\}$, pour $r = 1, \ldots, 5$. Avec ces vecteurs de caractéristiques, il n'est pas possible de séparer les deux classes par une ligne droite (représentant la frontière de décision d'un classifieur linéaire). En revanche, le vecteurs de caractéristiques transformé $\mathbf{z}^{(r)} = K\big(\mathbf{x}^{(r)}, \cdot\big)$ permet de séparer les points de données à l'aide d'un classifieur linéaire.

méthodes basées sur le gradient Les méthodes basées sur le gradient sont des techniques itératives pour trouver le minimum (ou le maxi-

mum) 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.

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 [58]. 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 $\mathcal{H}$) [28]. 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 $p(\mathbf{x}, y)$ [28, 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 [59].

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 $\mathbf{x}$ [25, 28]. Le risque bayésien correspondant est donné par la variance a posteriori $\sigma^2_{y|\mathbf{x}}$ (voir Figure 19).

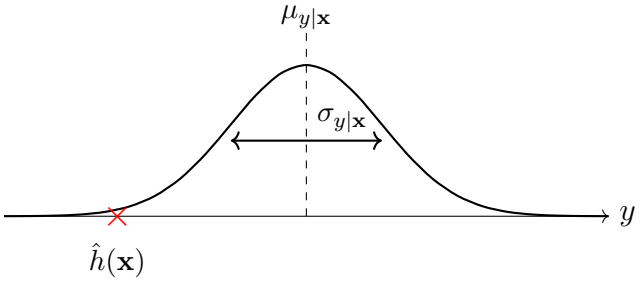


Fig. 19. Si les caractéristiques et l'étiquette d'un point de données suivent une loi normale multivariée, on peut atteindre le risque minimal (sous perte quadratique) en utilisant l'estimateur bayésien $\mu_{y|\mathbf{x}}$ pour prédire l'étiquette y d'un point de données avec des caractéristiques $\mathbf{x}$. 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 $\hat{h}$.

non régulière (ou non lisse) On qualifie une fonction de non régulière si elle n'est pas régulière [60].

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 [23].

Voir aussi: fonction, espace vectoriel.

noyau Considérons des points de données caractérisés par un vecteur de caractéristiques $\mathbf{x} \in \mathcal{X}$ avec un espace des caractéristiques générique $\mathcal{X}$. Un noyau (à valeurs réelles) $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ associe à chaque paire

de vecteurs de caractéristiques $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ un nombre réel $K\big(\mathbf{x}, \mathbf{x}'\big)$. La valeur $K\big(\mathbf{x}, \mathbf{x}'\big)$ est souvent interprétée comme une mesure de similarité entre $\mathbf{x}$ et $\mathbf{x}'$. Les méthode à noyaus utilisent un noyau pour transformer le vecteur de caractéristiques $\mathbf{x}$ en un nouveau vecteur de caractéristiques $\mathbf{z} = K\big(\mathbf{x}, \cdot\big)$. Ce nouveau vecteur de caractéristiques appartient à un espace des caractéristiques linéaire $\mathcal{X}'$, qui est (en général) différent de l'espace des caractéristiques original $\mathcal{X}$. L'espace des caractéristiques $\mathcal{X}'$ possède une structure mathématique spécifique : c'est un espace de Hilbert à noyau reproduisant [56, 57].

nuage de points Une technique de visualisation qui représente des points de données par des marqueurs dans un plan bidimensionnel. La Fig. 20 montre un exemple de nuage de points.

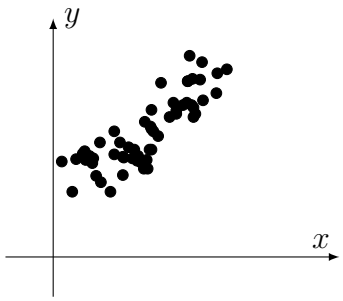


Fig. 20. Un nuage de points avec des marqueurs cercles, où les points de données représentent les conditions météorologiques quotidiennes en Finlande. Chaque point de données est caractérisé par sa température minimale diurne x comme caractéristique et sa température maximale diurne y comme étiquette. 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 : réduction de dimension.

opérateur proximal Étant donné une fonction convexe $f(\mathbf{w}')$, on définit son opérateur proximal comme suit [61, 62] :

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

Comme illustré à la Figure 21, 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, défini pour une fonction régulière et convexe $f(\mathbf{w}')$. En effet, effectuer un pas de gradient 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 $\rho = 1/\eta$.

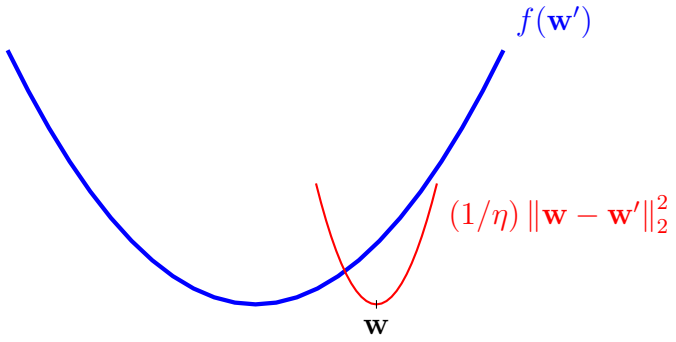


Fig. 21. Un pas généralisé met à jour un vecteur $\mathbf{w}$ en minimisant une version pénalisée de la fonction $f(\cdot)$. Le terme de pénalité correspond à la distance euclidienne quadratique pondérée entre la variable d'optimisation $\mathbf{w}'$ et le vecteur donné $\mathbf{w}$.

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.

paramètres du modèle Les paramètre d'un modèle sont des quantités utilisées pour sélectionner une fonction hypothèse spécifique à partir d'un modèle. On peut considérer une liste de paramètre 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.

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 22)

$$\widehat{\mathbf{w}} := \mathbf{w} - \eta \nabla f(\mathbf{w}). \tag{6}$$

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

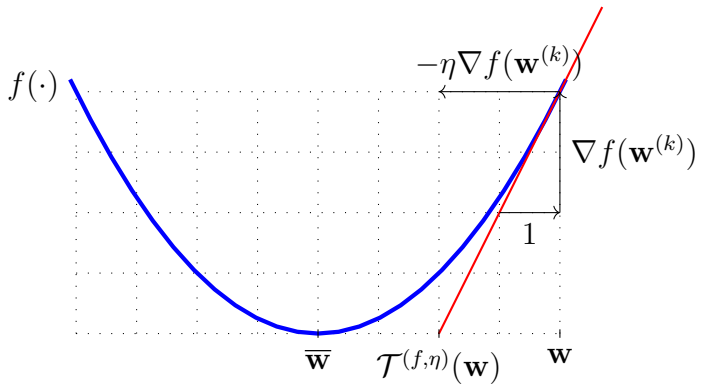


Fig. 22. Le pas classique (6) transforme un vecteur donné $\mathbf{w}$ en le vecteur mis à jour $\mathbf{w}'$. Il définit un opérateur $\mathcal{T}^{(f,\eta)}(\cdot) : \mathbb{R}^d \to \mathbb{R}^d : \mathbf{w} \mapsto \widehat{\mathbf{w}}$.

Notez que le pas (6) 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 (6) est d'optimiser

localement la fonction elle-même - au lieu de son approximation linéaire - telle que

$$\widehat{\mathbf{w}} = \underset{\mathbf{w}' \in \mathbb{R}^d}{\operatorname{argmin}} f(\mathbf{w}') + (1/\eta) \left\| \mathbf{w} - \mathbf{w}' \right\|_2^2 . \tag{7}$$

Nous utilisons intentionnellement le même symbole η pour le paramètre dans (7) que celui utilisé pour la taille de pas dans (6). Plus le η choisi dans (7) 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 (6), la mise à jour (7) 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 $f(\cdot)$ [61].

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 une donnée $\mathbf{z}$ et une hypothèse h.

perte logistique Considérons un point de données caractérisé par des caractéristiques $\mathbf{x}$ et une étiquette binaire $y \in \{-1, 1\}$. On utilise une hypothèse à valeurs réelles h pour prédire l'étiquette y à partir des caractéristiques $\mathbf{x}$. La perte logistique associée à cette prédiction est définie comme

$$L((\mathbf{x}, y), h) := \log(1 + \exp(-yh(\mathbf{x}))). \tag{8}$$

Il est important de noter que l'expression (8) de la perte logistique

s'applique uniquement dans le cas d'un espace des étiquettes $\mathcal{Y} = \{-1, 1\}$ et lorsque l'on utilise la règle de seuillage définie en (1).

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

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

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.

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 nœuds $i, i' \in \mathcal{V}$.

point de données Un point de données est un objet qui transmet de l'information [24]. 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 :

- 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.

- 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 23 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.

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. 23. 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.

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 [63, Article 5(1)(c)], [64].

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, 27, 65].

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], [23] :

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

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

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

Autrement dit, $P_{\mathcal{W}}\big(\mathbf{w}\big)$ 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 [18].

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 d'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 $\mathbf{X} = \mathbf{A}$ [66, Ch. 3]. Les paramètres du modèle appris par la régression Ridge sont donnés par

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

On peut alors définir la pseudo-inverse $\mathbf{A}^+ \in \mathbb{R}^{d \times m}$ avec la limite [67, 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, régression Ridge.

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.

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., alors $L\left((\mathbf{x}, y), h\right)$ devient la réalisation d'une VA. La 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.

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 [28].

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 $\mathcal{D}$.

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 [63]. 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.

réalisation Considérons une VA x qui associe à chaque élément (c'est-à-dire un résultat ou événement élémentaire) $\omega \in \mathcal{P}$ d'un espace probabilisé $\mathcal{P}$ un élément a d'un espace mesurable $\mathcal{N}$ [2, 6, 27]. Une réalisation de x est tout élément $a' \in \mathcal{N}$ pour lequel il existe un élément $\omega' \in \mathcal{P}$ tel que $x(\omega') = a'$.

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.

réduction de dimension Les méthodes de réduction de dimension associent des caractéristiques brutes (généralement nombreuses) à un ensemble (relativement petit) de nouvelles caractéristiques. Ces méthodes peuvent être utilisées pour visualiser des points de données en apprenant deux caractéristiques pouvant servir de coordonnées pour une représentation dans un 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.

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].

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.

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).

régression polynomiale La régression polynomiale 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).

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. centrées autour de ces points de données.

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 origin-

aux 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 24 illustre ces trois approches de régularisation. Ces approches sont étroitement liées et parfois entièrement équivalentes : la augmentation de données qui utilise des VA normales centrées réduites 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.

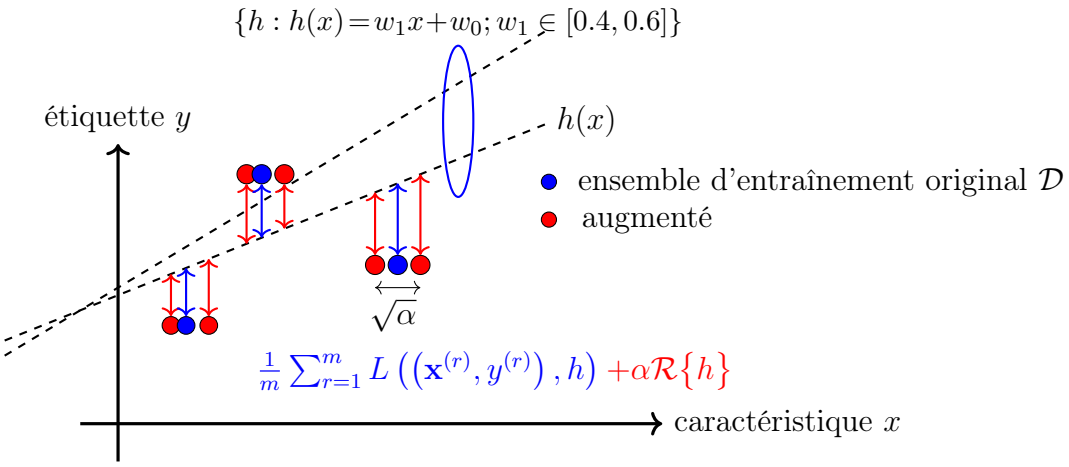


Fig. 24. Trois approches pour la régularisation: 1) augmentation de données; 2) pénalisation de la perte; et 3) élagage du modèle (via des contraintes sur les paramètres du modèle).

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 $\mathcal{C}^1$) [60, 68]. 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 25).

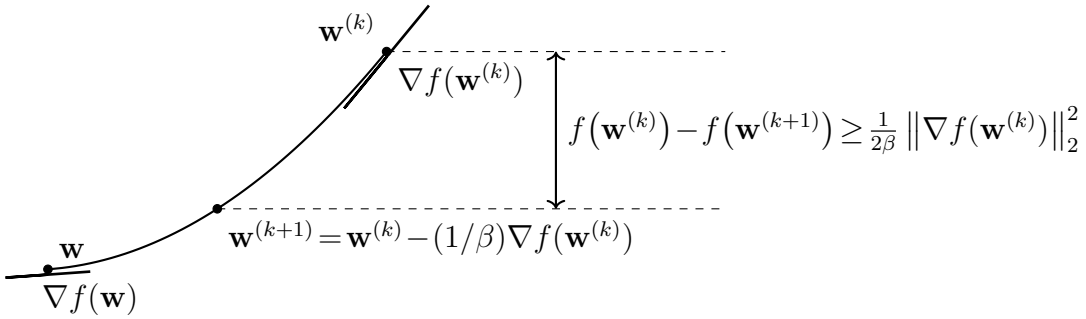


Fig. 25. Considérons une fonction objective $f(\mathbf{w})$ qui est β-*smooth*. Effectuer un pas avec une taille de pas $\eta = 1/\beta$ diminue la fonction objective d'au moins $\frac{1}{2\beta} \left\|\nabla f(\mathbf{w}^{(k)})\right\|_2^2$ [60, 68, 69]. Notez que la taille de pas $\eta = 1/\beta$ 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.

réseau d'apprentissage fédéré Un réseau d'apprentissage fédéré est un graphe non orienté pondéré dont les nœuds représentent des générateurs de données visant à entraîner un modèle local (ou personnalisé). Chaque nœud dans un réseau d'apprentissage fédéré représente un appareil capable de collecter un jeu de données local et, à son tour, d'entraîner un modèle local. Les méthodes d'apprentissage fédéré apprennent une hypothèse locale $h^{(i)}$, pour chaque nœud $i \in \mathcal{V}$, telle qu'elle engendre une faible perte sur les jeux de données locaux.

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.

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 [70].

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 applications noyau 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 [57].

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.

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.

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

transformation de caractéristiques Un trasnformation de caractéristiques est une application qui transforme les caractéristiques originales d'un point de données en de nouvelles caractéristiques. Les nouvelles caractéristiques obtenues peuvent être préférables aux caractéristiques d'origine pour plusieurs raisons. Par exemple, l'agencement des points de données peut devenir plus simple (ou plus linéaire) dans le nouvel espace des caractéristiques, permettant ainsi l'utilisation de modèles linéaires dans ce nouvel espace. Cette idée est un moteur central du développement des méthodes à noyau [57]. Par ailleurs, les couches cachées d'un réseau de neurones profond peuvent être interprétées comme une transformation de caractéristiques entraînable, suivie d'un modèle linéaire sous forme de couche de sortie. Une autre raison d'apprendre une transformation de caractéristiques peut être de réduire le surapprentissage et d'assurer une meilleure interprétabilité en apprenant un petit nombre de caractéristiques pertinentes [35]. Le cas particulier d'une transformation de caractéristiques produisant deux caractéristiques numériques est particulièrement utile pour la visualisa-

tion des données. En effet, on peut représenter les points de données dans un nuage de points en utilisant ces deux caractéristiques comme coordonnées.

transparence La transparence est une exigence fondamentale pour une IA digne de confiance [71]. Dans le contexte des méthodes d'apprentissage automatique, le terme est souvent utilisé de manière interchangeable avec explicabilité [31, 72]. 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 [55]. 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 [48] ou les cartes descriptives de systèmes d'IA [73] aident les praticiens à comprendre les cas d'usage prévus ainsi que les limitations du système [74].

tâche d'apprentissage Considérons un jeu de données $\mathcal{D}$ constitué de plusieurs points de données, chacun étant caractérisé par des caractéristiques $\mathbf{x}$. Par exemple, le jeu de données $\mathcal{D}$ peut être constitué des images d'une base de données particulière. Parfois, il peut être utile de représenter un jeu de données $\mathcal{D}$, ainsi que le choix des caractéristiques, par une loi de probabilité $p(\mathbf{x})$. Une tâche d'apprentissage associée à $\mathcal{D}$ consiste en un choix spécifique pour l'étiquette d'un point de données et l'espace des étiquettes correspondant. Étant donné un choix de fonction de perte et de modèle, une tâche d'apprentissage donne lieu à une instance de MRE. Ainsi, on pourrait aussi définir une tâche d'apprentissage via une instance de MRE, c'est-à-dire via une fonction objective. Remarquons que, pour un même jeu de données, on obtient différentes tâches d'apprentissage en utilisant différents choix de caractéristiques et d'étiquette d'un point de données. Ces tâches d'apprentissage sont liées, puisqu'elles sont basées sur le même jeu de données, et les résoudre conjointement (via des méthodes de apprentissage multitâche) est en général préférable à des résolutions distinctes [75], [76], [77].

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

$\mathbf{Ax} = \lambda\mathbf{x}$.

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 $\mathcal{D}$.

variable aléatoire (VA) Une VA est une fonction qui associe chaque événement élémentaire d'un espace probabilisé $\mathcal{P}$ à une valeur dans un espace d'arrivée [25], [6]. L'espace probabilisé est composé d'événements élémentaires et est muni d'une mesure de probabilité qui attribue des probabilités aux sous-ensembles de $\mathcal{P}$. Les différents types de VA comprennent :

- les VA binaires, qui associent chaque événement élémentaire 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 $\mathbb{R}$;
- les VA vectorielles, qui associent chaque événement élémentaire à un vecteur de l'espace euclidien $\mathbb{R}^d$.

La théorie des probabilités utilise le concept d'espaces mesurables pour définir rigoureusement et étudier les propriétés de (grandes) collections de VA [6].

Voir aussi: fonction, espace probabilisé, probabilité, espace euclidien.

variable aléatoire normale centrée réduite Une VA normale centrée réduite est une

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

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

Un vecteur aléatoire gaussien $\mathbf{x} \in \mathbb{R}^d$ ayant pour matrice de covariance $\mathbf{C}$ et pour moyenne $\boldsymbol{\mu}$ peut être construit via $\mathbf{x} := \mathbf{A}\big(\mathbf{z} + \boldsymbol{\mu}\big)$, où $\mathbf{A}$ est une matrice telle que $\mathbf{A}\mathbf{A}^T = \mathbf{C}$, et $\mathbf{z} := \big(z_1, \ldots, z_d\big)^T$ est un vecteur dont les composantes sont des VA normales centrées réduites i.i.d. $z_1, \ldots, z_d$.

Les vecteurs aléatoires gaussiens constituent un cas particulier des processus gaussiens, qui sont des transformations linéaires de suites infinies de VA normales centrées réduites [78].

Les VA normales sont largement utilisées comme modèles probabilistes pour l'analyse statistique en apprentissage automatique. Leur importance provient en partie du théorème central limite, qui stipule que la moyenne d'un nombre croissant de VA indépendantes (pas nécessairement normales) converge vers une VA normale [26].

Voir aussi : loi de probabilité, espace probabilisé.

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 $\mathbb{E}\big\{\big\|\mathbf{x} - \mathbb{E}\{\mathbf{x}\}\big\|_2^2\big\}$.

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).

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 centrées réduites 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., VA normale centrée réduite, loi normale multivariée, VA.

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

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

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. 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., 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 [18], [79].

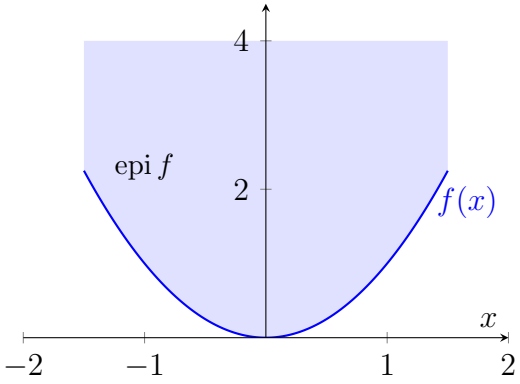


Fig. 26. Épigraphe de la fonction $f(x) = x^2$ (i.e., la zone colorée).

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 » [15], [16], [17].

Voir aussi: point de données.

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

[14] 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.

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

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

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

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

[19] 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

[20] 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.

[21] 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.

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

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

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

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

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

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

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

[29] 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

[30] 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.

[31] 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.

[32] 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

[33] 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/

[34] 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.

[35] 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.

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

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

[38] 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.

[39] 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

[40] 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

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

[42] 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

[43] 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.

[44] 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/

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

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

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

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

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

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

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

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

[53] 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.

[54] 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

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

[56] 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.

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

[58] 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.

[59] 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.

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

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

[62] 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.

[63] 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

[64] 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

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

[66] 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.

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

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

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

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

[71] 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

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

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

[74] 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.

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

[76] 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.

[77] 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.

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

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