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

Alexander Jung[1], Konstantina Olioumtsevits[1], et Juliette Gronier[2]
[1]Aalto University [2]ENS Lyon

July 8, 2025

please cite as: A. Jung, K. Olioumtsevits, and J. Gronier, *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. Certaines figures de ce dictionnaire ont été produites avec l'aide de Salvatore Rastelli.

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`.

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 $a \in \mathbb{R}$.
$\log(a)$	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].

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, loi de probabilité
$\text{cov}\,(x, y)$	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, point de données, 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, 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 : 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'label space $\mathcal{Y}$ d'une méthode d'apprentissage automatique comprend toutes les valeurs d'étiquette qu'un point de données peut porter. L'label space 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 label space 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 label space discret, comme $\mathcal{Y} = \{0, 1\}$ ou $\mathcal{Y} = \{chat, chien, souris\}$, sont appelés problèmes (ou méthodes) de classification. Voir aussi : label space, étiquette, jeu de données, apprentissage automatique.
$\mathcal{B}$	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, caractéristiques, 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 : espace des caractéristiques, label space, hypothèse.
$\mathcal{H}$	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, hypothèse, apprentissage automatique.
$d_{\text{eff}}(\mathcal{H})$	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 fonction hypothèse $\hat{h}$ apprise par une méthode d'apprentissage automatique. La méthode est entraînée sur des points de données modélisés comme des réalisations de VA. Puisque les données sont des réalisations de VA, la fonction hypothèse apprise $\hat{h}$ est également une réalisation d'une VA. Voir aussi : biais, hypothèse, points de données, réalisation, VA.
V	La variance (des paramètres) de la fonction hypothèse apprise par une méthode d'apprentissage automatique. La méthode est entraînée sur des points de données modélisés comme des réalisations de VA. Puisque les données sont des réalisations de VA, la fonction hypothèse apprise $\hat{h}$ est également une réalisation d'une VA. Voir aussi : variance, hypothèse, paramètres, points de données, réalisations, VA.
$L\left((\mathbf{x}, y), h\right)$	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 : 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, poids
$\phi(\cdot)$	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, transformation de caractéristiques

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 tâche d'apprentissages. 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. Le appareil peut accéder à un jeu de données local et entraîner un local model. Voir aussi : appareil, réseau d'apprentissage fédéré, jeu de données local, local model
$\mathcal{G}^{(\mathcal{C})}$	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, voisinage
$d_{\max}^{(\mathcal{G})}$	Le degré pondéré maximal (parmi les degrés pondérés des nœuds) d'un graphe $\mathcal{G}$. Voir aussi : 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 le 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

k-fold cross-validation (k-fold CV) k-fold CV is a method for learning and validating a hypothèse using a given jeu de données. This method divides the jeu de données evenly into k subsets or folds and then executes k repetitions of modèle training (e.g., via minimisation du risque empirique (MRE)) and validation. Each repetition uses a different fold as the ensemble de validation and the remaining $k-1$ folds as a ensemble d'entraînement. The final output is the average of the erreur de validations obtained from the k repetitions.

k-moyennes The k-moyennes algorithme is a hard clustering method which assigns each point de données of a jeu de données to precisely one of k different clusters. The method alternates between updating the cluster assignments (to the cluster with the nearest moyenne) and, given the updated cluster assignments, re-calculating the cluster moyennes [8, Ch. 8].

absolute error loss Consider a point de données with caractéristiques $\mathbf{x} \in \mathcal{X}$ and numeric étiquette $y \in \mathbb{R}$. The absolute error perte incurred by a hypothèse $h : \mathcal{X} \to \mathbb{R}$ is defined as $|y - h(\mathbf{x})|$, i.e., the absolute difference between the prédiction $h(\mathbf{x})$ and the true étiquette y.

accuracy Consider point de donnéess characterized by caractéristiques $\mathbf{x} \in \mathcal{X}$ and a categorical label y which takes on values from a finite label space

$\mathcal{Y}$. The accuracy of a hypothèse $h : \mathcal{X} \to \mathcal{Y}$, when applied to the point de donnéess in a jeu de données $\mathcal{D} = \left\{\left(\mathbf{x}^{(1)}, y^{(1)}\right), \ldots, \left(\mathbf{x}^{(m)}, y^{(m)}\right)\right\}$, is then defined as $1 - (1/m) \sum_{r=1}^{m} L^{(0/1)}\left(\left(\mathbf{x}^{(r)}, y^{(r)}\right), h\right)$ using the 0/1 loss $L^{(0/1)}(\cdot, \cdot)$.

algebraic connectivity The algebraic connectivity of an undirected graphe is the second-smallest valeur propre λ_2 of its matrice laplacienne. A graphe is connected if and only if $\lambda_2 > 0$.

algorithme An algorithm is a precise, step-by-step specification for how to produce an output from a given input within a finite number of computational steps [9]. For example, an algorithm for training a modèle linéaire explicitly describes how to transform a given ensemble d'entraînement into paramètres du modèle through a sequence of gradient steps. This informal characterization can be formalized rigorously via different mathematical modèles [10]. One very simple modèle of an algorithm is a collection of possible executions. Each execution is a sequence:

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

that respects the constraints inherent to the computer executing the algorithm. Algorithms may be deterministic, where each input results uniquely in a single execution, or randomized, where executions can vary probabilistically. Randomized algorithms can thus be analyzed by modeling execution sequences as outcomes of random experiments, viewing the algorithm as a stochastic process [7, 11, 12]. Crucially, an algorithm encompasses more than just a mapping from input to output;

it also includes the intermediate computational steps $s_1, \ldots, s_T$.

analyse en composantes principales (ACP) PCA determines a linear transformation de caractéristiques such that the new caractéristiques allow us to reconstruct the original caractéristiques with the minimum reconstruction error [8].

appareil Tout système physique qui peut être utilisé pour stocker et traiter des donnéess. 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 programming interface (API) An API is a formal mechanism that allows software components to interact in a structured and modular way [13]. In the context of apprentissage automatique, APIs are commonly used to provide access to a trained apprentissage automatique modèle. Users—whether humans or machines—can submit the vecteur de caractéristiques of a point de données and receive a corresponding prédiction. Suppose a trained apprentissage automatique modèle is defined as $\widehat{h}(x) := 2x + 1$. Through an API, a user can input $x = 3$ and receive the output $\widehat{h}(3) = 7$ without knowledge of the detailed structure of the apprentissage automatique modèle or its training. In practice, the modèle is typically deployed on a server connected to the internet. Clients send requests containing caractéristique values to

the server, which responds with the computed prediction $\widehat{h}(\mathbf{x})$. APIs promote modularity in apprentissage automatique system design: one team can develop and train the model, while another handles integration and user interaction. Publishing a trained modèle via an API also offers practical advantages:

- The server can centralize computational resources which are required to compute prédictions.
- The internal structure of the modèle remains hidden (useful for protecting IP or trade secrets).

However, APIs are not without risk: techniques such as model inversion can potentially reconstruct a modèle from its prédictions on carefully selected vecteur de caractéristiquess.

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 issue d'un espace des hypothèses (ou modèle) par la minimisation d'une fonction de perte [8, 14]. Une formulation précise de ce principe est donnée par le 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].

apprentissage fédéré FL is an umbrella term for apprentissage automatique methods that train modèles in a collaborative fashion using decentralized données and computation.

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 deep net pour les deux tâches et de ne permettre qu'aux poids de la couche de sortie finale d'être différents.

apprentissage semi-supervisé SSL methods use unlabeled point de donnéess to support the learning of a hypothèse from labeled datapoints [15]. This approach is particularly useful for apprentissage automatique applications that offer a large amount of unlabeled point de donnéess, but only a limited number of labeled datapoints.

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

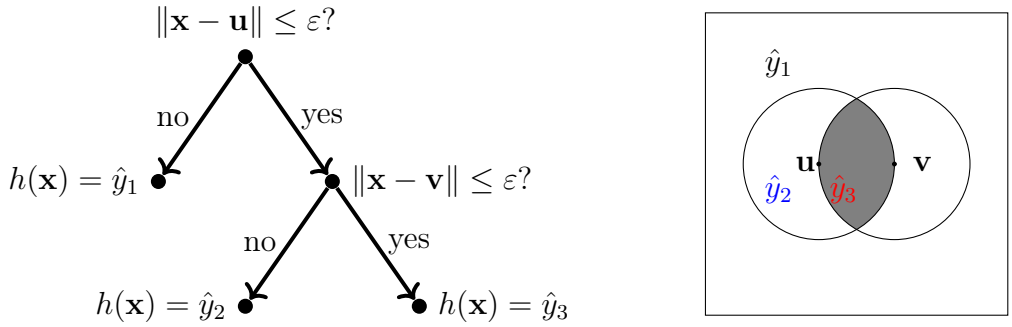


Fig. 1. Left: A decision tree is a flow-chart-like representation of a piecewise constant hypothèse $h : \mathcal{X} \to \mathbb{R}$. Each piece is a région de décision $\mathcal{R}_{\hat{y}} := \{\mathbf{x} \in \mathcal{X} : h(\mathbf{x}) = \hat{y}\}$. The depicted decision tree can be applied to numeric vecteur de caractéristiquess, i.e., $\mathcal{X} \subseteq \mathbb{R}^d$. It is parametrized by the threshold $\varepsilon > 0$ and the vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$. Right: A decision tree partitions the espace des caractéristiques $\mathcal{X}$ into région de décisions. Each région de décision $\mathcal{R}_{\hat{y}} \subseteq \mathcal{X}$ corresponds to a specific leaf node in the decision tree.

autoencoder An autoencoder is an apprentissage automatique method that simultaneously learns an encoder map $h(\cdot) \in \mathcal{H}$ and a decoder map $h^*(\cdot) \in \mathcal{H}^*$. It is an instance of MRE using a perte computed from the reconstruction error $\mathbf{x} - h^*\big(h\big(\mathbf{x}\big)\big)$.

backdoor A backdoor attack refers to the intentional manipulation of the training process underlying an apprentissage automatique method. This manipulation can be implemented by perturbing the ensemble d'entraînement (données poisoning) or the optimization algorithme used by an MRE-based method. The goal of a backdoor attack is to nudge the learned hypothèse $\hat{h}$ towards specific prédictions for a certain range

of caractéristique values. This range of caractéristique values serves as a key (or trigger) to unlock a backdoor in the sense of delivering anomalous prédictions. The key $\mathbf{x}$ and the corresponding anomalous prédiction $\hat{h}(\mathbf{x})$ are only known to the attacker.

bagging Bagging (or bootstrap aggregation) is a generic technique to improve (the robustness of) a given apprentissage automatique method. The idea is to use the bootstrap to generate perturbed copies of a given jeu de données and then to learn a separate hypothèse for each copy. We then predict the étiquette of a point de données by combining or aggregating the individual prédictions of each separate hypothèse. For hypothèse maps delivering numeric étiquette values, this aggregation could be implemented by computing the average of individual prédictions.

baseline Consider some apprentissage automatique method that produces a learned hypothèse (or trained modèle) $\hat{h} \in \mathcal{H}$. We evaluate the quality of a trained modèle by computing the average perte on a ensemble de test (ou jeu de test). But how can we assess whether the resulting ensemble de test performance is sufficiently good? How can we determine if the trained modèle performs close to optimal and there is little point in investing more resources (for données collection or computation) to improve it? To this end, it is useful to have a reference (or baseline) level against which we can compare the performance of the trained modèle. Such a reference value might be obtained from human performance, e.g., the misclassification rate of dermatologists who diagnose cancer from visual inspection of skin [16]. Another source for a baseline is an

existing, but for some reason unsuitable, apprentissage automatique method. For example, the existing apprentissage automatique method might be computationally too expensive for the intended apprentissage automatique application. Nevertheless, its ensemble de test error can still serve as a baseline. Another, somewhat more principled, approach to constructing a baseline is via a probabilistic model. In many cases, given a probabilistic model $p(\mathbf{x}, y)$, we can precisely determine the minimum achievable risque among any hypotheses (not even required to belong to the espace des hypothèses $\mathcal{H}$) [17]. This minimum achievable risque (referred to as the Bayes risk) is the risque of the Bayes estimator for the étiquette y of a point de données, given its caractéristiques $\mathbf{x}$. Note that, for a given choice of fonction de perte, the Bayes estimator (if it exists) is completely determined by the loi de probabilité $p(\mathbf{x}, y)$ [17, Ch. 4]. However, computing the Bayes estimator and Bayes risk presents two main challenges:

1) The loi de probabilité $p(\mathbf{x}, y)$ is unknown and needs to be estimated.
2) Even if $p(\mathbf{x}, y)$ is known, it can be computationally too expensive to compute the Bayes risk exactly [18].

A widely used probabilistic model is the loi normale multivariée $(\mathbf{x}, y) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ for point de donnéess characterized by numeric caractéristiques and étiquettes. Here, for the squared error loss, the Bayes estimator is given by the posterior moyenne $\mu_{y|\mathbf{x}}$ of the étiquette y, given the caractéristiques $\mathbf{x}$ [17, 19]. The corresponding Bayes risk is given by the posterior variance $\sigma^2_{y|\mathbf{x}}$ (see Figure 2).

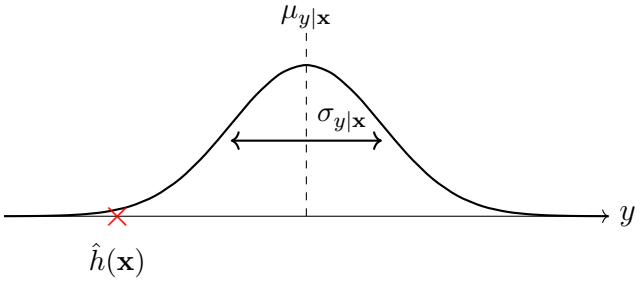


Fig. 2. If the caractéristiques and the étiquette of a point de données are drawn from a loi normale multivariée, we can achieve the minimum risque (under squared error loss) by using the Bayes estimator $\mu_{y|\mathbf{x}}$ to predict the étiquette y of a point de données with caractéristiques $\mathbf{x}$. The corresponding minimum risque is given by the posterior variance $\sigma^2_{y|\mathbf{x}}$. We can use this quantity as a baseline for the average perte of a trained modèle $\hat{h}$.

Bayes estimator Consider a probabilistic model with a joint loi de probabilité $p(\mathbf{x}, y)$ for the caractéristiques $\mathbf{x}$ and étiquette y of a point de données. For a given fonction de perte $L(\cdot, \cdot)$, we refer to a hypothèse h as a Bayes estimator if its risque $\mathbb{E}\{L((\mathbf{x}, y), h)\}$ is the minimum [17]. Note that the property of a hypothèse being a Bayes estimator depends on the underlying loi de probabilité and the choice for the fonction de perte $L(\cdot, \cdot)$.

Bayes risk Consider a probabilistic model with a joint loi de probabilité $p(\mathbf{x}, y)$ for the caractéristiques $\mathbf{x}$ and étiquette y of a point de données. The Bayes risque is the minimum possible risque that can be achieved by any hypothèse $h : \mathcal{X} \to \mathcal{Y}$. Any hypothèse that achieves the Bayes risk is referred to as a Bayes estimator [17].

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

boosting Boosting is an iterative optimization method to learn an accurate hypothèse map (or strong learner) by sequentially combining less accurate hypothèse maps (referred to as weak learners) [20, Ch. 10]. For example, weak learners are shallow arbre de décisions which are combined to obtain a deep arbre de décision. Boosting can be understood as a generalization of méthodes basées sur le gradient for MRE using parametric modèles and lisse fonction de pertes [21]. Just like descente de gradient iteratively updates paramètres du modèle to reduce the risque empirique, boosting iteratively combines (e.g., by summation) hypothèse maps to reduce the risque empirique. A widely-used instance of the generic boosting idea is referred to as gradient boosting, which

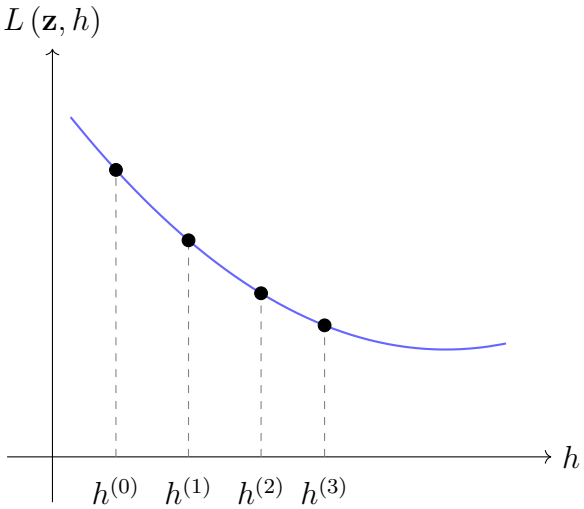


Fig. 3. Boosting methods construct a sequence of hypothèse maps $h^{(0)}, h^{(1)}, \ldots$ that are increasingly strong learners (i.e., incurring a smaller perte).

uses gradients of the fonction de perte for combining the weak learners [21].

See also: gradient step, méthodes basées sur le gradient

bootstrap For the analysis of apprentissage automatique methods, it is often useful to interpret a given set of point de donnéess $\mathcal{D} = \left\{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}\right\}$ as réalisations of i.i.d. VAs with a common loi de probabilité $p(\mathbf{z})$. In general, we do not know $p(\mathbf{z})$ exactly, but we need to estimate it. The bootstrap uses the histogram of $\mathcal{D}$ as an estimator for the underlying loi de probabilité $p(\mathbf{z})$.

borne supérieure The supremum of a set of real numbers is the smallest number that is greater than or equal to every element in the set. More

formally, a real number a is the supremum of a set $\mathcal{A} \subseteq \mathbb{R}$ if: 1) a is an upper bound of $\mathcal{A}$; and 2) no number smaller than a is an upper bound of $\mathcal{A}$. Every non-empty set of real numbers that is bounded above has a supremum, even if it does not contain its supremum as an element [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 le terme caractéristique incluent « covariable », « variable explicative », « variable indépendante », « variable d'entrée », « variable prédictive » ou « régressseur » [22], [23], [24].

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.

classification multi-classe Multi-étiquette classification problems and methods use point de donnéess that are characterized by several étiquettes. As an example, consider a point de données representing a picture with two étiquettes. One étiquette indicates the presence of a human in this picture and another étiquette indicates the presence of a car.

classifier A classifier is a hypothèse (map) $h(\mathbf{x})$ used to predict a étiquette taking values from a finite label space. We might use the function value $h(\mathbf{x})$ itself as a prédiction $\hat{y}$ for the étiquette. However, it is customary to use a map $h(\cdot)$ that delivers a numeric quantity. The prédiction is then obtained by a simple thresholding step. For example, in a binary classification problem with $\mathcal{Y} \in \{-1, 1\}$, we might use a real-valued hypothèse map $h(\mathbf{x}) \in \mathbb{R}$ as a classifier. A prédiction $\hat{y}$ can then be obtained via thresholding,

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

We can characterize a classifier by its région de décisions $\mathcal{R}_a$, for every possible étiquette value $a \in \mathcal{Y}$.

cluster A cluster is a subset of point de donnéess that are more similar to each other than to the point de donnéess outside the cluster. The quantitative measure of similarity between point de donnéess is a design choice. If point de donnéess are characterized by Euclidean vecteur de caractéristiquess $\mathbf{x} \in \mathbb{R}^d$, we can define the similarity between two point de donnéess via the Euclidean distance between their vecteur de caractéristiquess. An example of such clusters is shown in Figure 4.

clustered federated learning (CFL) CFL trains local models for the appareils in an apprentissage fédéré application by using a partitionnement de données assumption: The appareils of a réseau d'apprentissage fédéré form clusters. Two appareils in the same cluster generate jeu de données locals with similar statistical properties. CFL pools the jeu de

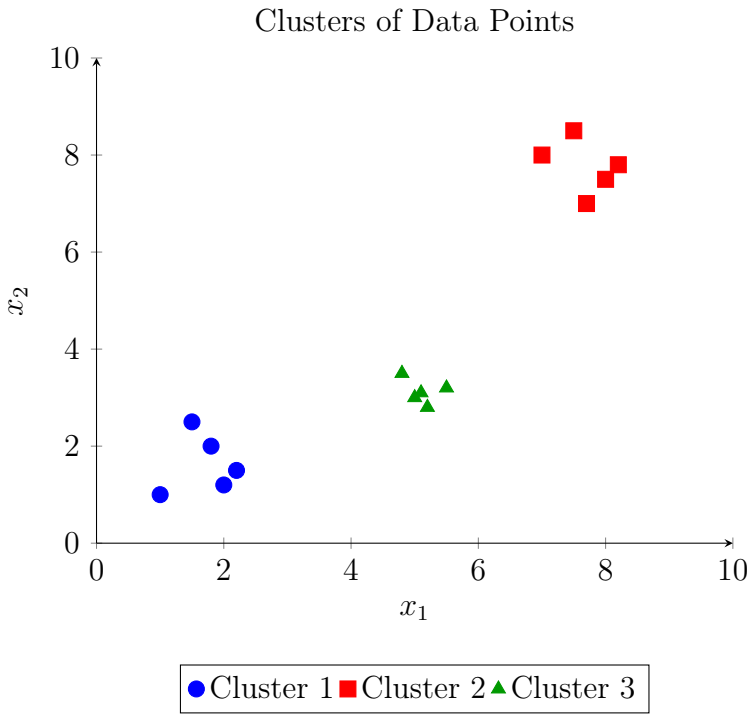


Fig. 4. Illustration of three clusters in a two-dimensional feature space. Each cluster groups data points that are more similar to each other than to those in other clusters, based on Euclidean distance.

données locals of appareils in the same cluster to obtain a ensemble d'entraînement for a cluster-specific modèle. Generalized total variation minimization (GTVMin) clusters appareils implicitly by enforcing approximate similarity of paramètres du modèle across well-connected nodes of the réseau d'apprentissage fédéré. See also: apprentissage fédéré, partitionnement de données, GTVMin.

clustering assumption The partitionnement de données assumption postulates that point de donnéess in a jeu de données form a (small) number of groups or clusters. Point de donnéess in the same cluster are more similar to each other than those outside the cluster [15]. We obtain different partitionnement de données methods by using different notions of similarity between point de donnéess.

computational aspects By computational aspects of an apprentissage automatique method, we mainly refer to the computational resources required for its implementation. For example, if an apprentissage automatique method uses iterative optimization techniques to solve MRE, then its computational aspects include: 1) how many arithmetic operations are needed to implement a single iteration (gradient step); and 2) how many iterations are needed to obtain useful paramètres du modèle. One important example of an iterative optimization technique is descente de gradient.

condition number The condition number $\kappa(\mathbf{Q}) \geq 1$ of a positive definite matrix $\mathbf{Q} \in \mathbb{R}^{d \times d}$ is the ratio α/β between the largest α and the smallest β valeur propre of $\mathbf{Q}$. The condition number is useful for the analysis of

apprentissage automatique methods. The computational complexity of méthodes basées sur le gradient for linear regression crucially depends on the condition number of the matrix $\mathbf{Q} = \mathbf{X}\mathbf{X}^T$, with the feature matrix $\mathbf{X}$ of the ensemble d'entraînement. Thus, from a computational perspective, we prefer caractéristiques of point de donnéess such that $\mathbf{Q}$ has a condition number close to 1.

confusion matrix Consider point de donnéess, which are characterized by caractéristiques $\mathbf{x}$ and étiquette y, having values from the finite label space $\mathcal{Y} = \{1, \ldots, k\}$. For a given hypothèse h, the confusion matrix is a $k \times k$ matrix with rows representing the elements of $\mathcal{Y}$. The columns of a confusion matrix correspond to the prédiction $h(\mathbf{x})$. The (c, c')-th entry of the confusion matrix is the fraction of point de donnéess with étiquette $y{=}c$ and resulting in a prédiction $h(\mathbf{x}){=}c'$.

connected graph An undirected graphe $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is connected if every non-empty subset $\mathcal{V}' \subset \mathcal{V}$ has at least one edge connecting it to $\mathcal{V} \setminus \mathcal{V}'$.

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

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

Here, $\|\mathbf{u}\|_p := \left(\sum_{j=1}^{d} |u_j|^p \right)^{1/p}$ denotes the p-norme (for $p \geq 1$). It turns out that many of the optimal vectors $\widehat{\mathbf{w}}^{(1)}, \ldots, \widehat{\mathbf{w}}^{(m)}$ coincide. A cluster then consists of those point de donnéess $r \in \{1, \ldots, m\}$ with identical $\widehat{\mathbf{w}}^{(r)}$ [25, 26].

convexe A subset $\mathcal{C} \subseteq \mathbb{R}^d$ of the Euclidean space $\mathbb{R}^d$ is referred to as convex if it contains the line segment between any two points $\mathbf{x}, \mathbf{y} \in \mathcal{C}$ in that set. A function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if its epigraph $\left\{ \left(\mathbf{w}^T, t \right)^T \in \mathbb{R}^{d+1} : t \geq f(\mathbf{w}) \right\}$ is a convex set [27]. We illustrate one example of a convex set and a convex function in Figure 5.

Fig. 5. Left: A convex set $\mathcal{C} \subseteq \mathbb{R}^d$. Right: A convex function $f : \mathbb{R}^d \to \mathbb{R}$.

Courant–Fischer–Weyl min-max characterization Consider a semi-définie positive matrix $\mathbf{Q} \in \mathbb{R}^{d \times d}$ with décomposition en éléments propres (or spectral decomposition),

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

Here, we use the ordered (in increasing fashion) valeur propres

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

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

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

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

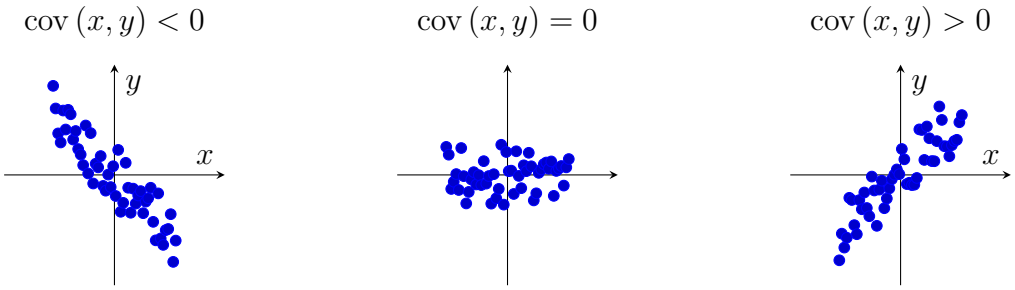


Fig. 6. Scatterplots illustrant des réalisations issues de trois probabilistic models différents pour deux VA avec des valeurs de covariance négative (gauche), nulle (centre) et positive (droite).

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 6 pour des exemples visuels.

critère d'arrêt Many apprentissage automatique methods use iterative algorithmes that construct a sequence of paramètres du modèle (such as the poids of a linear map or the poids of an RNA). These parameters (hopefully) converge to an optimal choice for the paramètres du modèle. In practice, given finite computational resources, we need to stop iterating after a finite number of repetitions. A stopping criterion is any well-defined condition required for stopping the iteration.

data augmentation Données augmentation methods add synthetic point de donnéess to an existing set of point de donnéess. These synthetic point de donnéess are obtained by perturbations (e.g., adding noise to

physical measurements) or transformations (e.g., rotations of images) of the original point de donnéess. These perturbations and transformations are such that the resulting synthetic point de donnéess should still have the same étiquette. As a case in point, a rotated cat image is still a cat image even if their vecteur de caractéristiquess (obtained by stacking pixel color intensities) are very different (see Figure 7). Données augmentation can be an efficient form of régularisation.

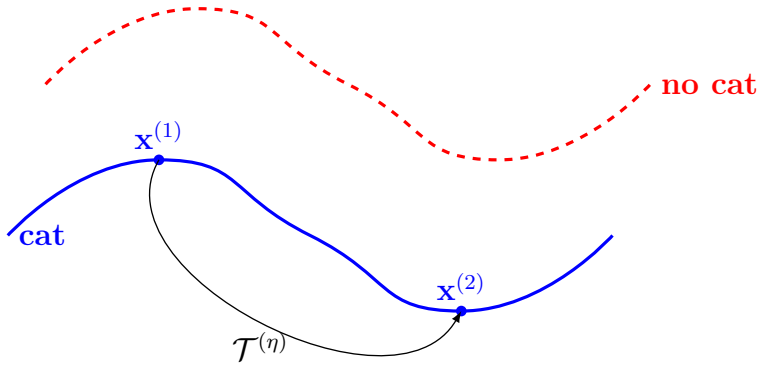


Fig. 7. Données augmentation exploits intrinsic symmetries of point de donnéess in some espace des caractéristiques $\mathcal{X}$. We can represent a symmetry by an operator $\mathcal{T}^{(\eta)} : \mathcal{X} \to \mathcal{X}$, parametrized by some number $\eta \in \mathbb{R}$. For example, $\mathcal{T}^{(\eta)}$ might represent the effect of rotating a cat image by η degrees. A point de données with vecteur de caractéristiques $\mathbf{x}^{(2)} = \mathcal{T}^{(\eta)}\left(\mathbf{x}^{(1)}\right)$ must have the same étiquette $y^{(2)} = y^{(1)}$ as a point de données with vecteur de caractéristiques $\mathbf{x}^{(1)}$.

data minimization principle European données protection regulation includes a données minimization principle. This principle requires a données controller to limit the collection of personal information to

what is directly relevant and necessary to accomplish a specified purpose. The données should be retained only for as long as necessary to fulfill that purpose [28, Article 5(1)(c)], [29].

data normalization Données normalization refers to transformations applied to the vecteur de caractéristiquess of point de donnéess to improve the apprentissage automatique method's statistical aspects or computational aspects. For example, in linear regression with méthodes basées sur le gradient using a fixed taux d'apprentissage, convergence depends on controlling the norme of vecteur de caractéristiquess in the ensemble d'entraînement. A common approach is to normalize vecteur de caractéristiquess such that their norme does not exceed one [8, Ch. 5].

data poisoning Données poisoning refers to the intentional manipulation (or fabrication) of point de donnéess to steer the training of an apprentissage automatique modèle [30, 31]. The protection against données poisoning is particularly important in distributed apprentissage automatique applications where jeu de donnéess are decentralized.

deep net A deep net is an RNA with a (relatively) large number of hidden layers. Deep learning is an umbrella term for apprentissage automatique methods that use a deep net as their modèle [32].

degree of belonging Degree of belonging is a number that indicates the extent to which a point de données belongs to a cluster [8, Ch. 8]. The degree of belonging can be interpreted as a soft cluster assignment. Soft clustering methods can encode the degree of belonging by a real number

in the interval $[0,1]$. Hard clustering is obtained as the extreme case when the degree of belonging only takes on values 0 or 1.

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

density-based spatial clustering of applications with noise (DBSCAN) DBSCAN refers to a partitionnement de données algorithme for point de donnéess that are characterized by numeric vecteur de caractéristiquess. Like k-moyennes and soft clustering via Gaussian mixture model (GMM), also DBSCAN uses the Euclidean distances between vecteur de caractéristiquess to determine the clusters. However, in contrast to k-moyennes and GMM, DBSCAN uses a different notion of similarity between point de donnéess. DBSCAN considers two point de donnéess as similar if they are connected via a sequence (path) of close-by intermediate point de donnéess. Thus, DBSCAN might consider two point de donnéess as similar (and therefore belonging to the same cluster) even if their vecteur de caractéristiquess have a large Euclidean distance.

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

gradient step

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

with a sufficiently small taille de pas $\eta > 0$. Figure 8 illustrates the effect of a single gradient descent step (2).

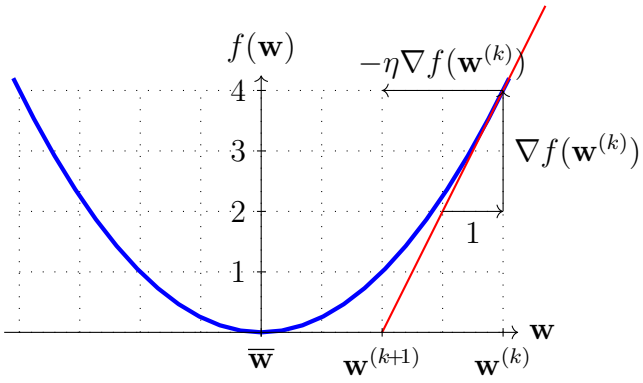


Fig. 8. A single gradient step (2) towards the minimizer $\overline{\mathbf{w}}$ of $f(\mathbf{w})$.

differential privacy (DP) Consider some apprentissage automatique method $\mathcal{A}$ that reads in a jeu de données (e.g., the ensemble d'entraînement used for MRE) and delivers some output $\mathcal{A}(\mathcal{D})$. The output could be either the learned paramètres du modèle or the prédictions for specific point de donnéess. DP is a precise measure of privacy leakage incurred by revealing the output. Roughly speaking, an apprentissage automatique method is differentially private if the loi de probabilité of the output $\mathcal{A}(\mathcal{D})$ does not change too much if the sensitive attribute of one point de données in the ensemble d'entraînement is changed. Note that DP builds on a probabilistic model for an apprentissage automatique method, i.e., we interpret its output $\mathcal{A}(\mathcal{D})$ as the réalisation of

an VA. The randomness in the output can be ensured by intentionally adding the réalisation of an auxiliary VA (noise) to the output of the apprentissage automatique method.

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 paramètres peuvent être les coefficients utilisés dans une application linéaire ou les poids et termes de biais d'un RNA.

discrepancy Consider an apprentissage fédéré application with networked data represented by an réseau d'apprentissage fédéré. apprentissage fédéré methods use a discrepancy measure to compare hypothèse maps from local models at nodes i, i' connected by an edge in the réseau d'apprentissage fédéré.

distributed algorithm A distributed algorithme is an algorithme designed for a special type of computer: a collection of interconnected computing devices (or nodes). These devices communicate and coordinate their local computations by exchanging messages over a network [33, 34]. Unlike a classical algorithme, which is implemented on a single appareil, a distributed algorithme is executed concurrently on multiple appareils with computational capabilities. Similar to a classical algorithme, a distributed algorithme can be modeled as a set of potential executions. However, each execution in the distributed setting involves both local computations and message-passing events. A generic execution might

look as follows:

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

Each appareil i starts from its own local input and performs a sequence of intermediate computations $s_k^{(i)}$ at discrete time instants $k = 1, \ldots, T_i$. These computations may depend on both: the previous local computations at the appareil and messages received from other appareils. One important application of distributed algorithmes is in apprentissage fédéré where a network of appareils collaboratively train a personal modèle for each appareil.

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 [35].

dual norm Every norme $\|\cdot\|$ defined on an Euclidean space $\mathbb{R}^d$ has an associated dual norme, denoted $\|\cdot\|_*$, defined as $\|\mathbf{y}\|_* := \sup_{\|\mathbf{x}\| \leq 1} \mathbf{y}^T \mathbf{x}$. The dual norme measures the largest possible inner product between $\mathbf{y}$ and any vector in the unit ball of the original norme. For further details, see [27, Sec. A.1.6].

décomposition en valeurs singulières The SVD for a matrix $\mathbf{A} \in \mathbb{R}^{m \times d}$ is a factorization of the form

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

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

décomposition en éléments propres The valeur propre decomposition for a square matrix $\mathbf{A} \in \mathbb{R}^{d \times d}$ is a factorization of the form

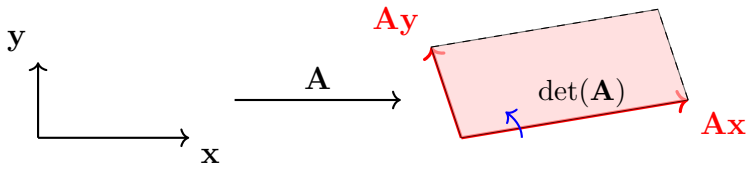
$$\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}.$$

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

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].

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], [36]. 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 $\det(\mathbf{A}) = \prod_{i=1}^{n} \lambda_i$ [37]. 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.

edge weight Each edge $\{i, i'\}$ of an réseau d'apprentissage fédéré is assigned a non-negative edge weight $A_{i,i'} \geq 0$. A zero edge weight $A_{i,i'} = 0$ indicates the absence of an edge between nodes $i, i' \in \mathcal{V}$.

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) A set of point de donnéess that have been used neither to train a modèle (e.g., via MRE) nor in a ensemble de validation to choose between different 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).

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 applica-

tion 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'Euclidean space $\mathbb{R}^d$, 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'label space. Le choix de cet espace doit tenir compte des ressources informatiques disponibles ainsi que des statistical aspects. 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 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 uncertainty dans les applications d'apprentissage automatique [6, 19, 38].

espérance Soit un vecteur de caractéristiques numérique $\mathbf{x} \in \mathbb{R}^d$ que l'on interprète comme la réalisation d'une VA suivant une loi de probabilité $p(\mathbf{x})$. On définit l'espérance de $\mathbf{x}$ comme l'intégrale $\mathbb{E}\{\mathbf{x}\} := \int \mathbf{x} p(\mathbf{x})$ [2, 6, 39]. Remarquons que l'espérance n'est définie que si cette intégrale existe, c'est-à-dire si la VA est intégrable.

estimation error Consider point de donnéess, each with vecteur de caractéristiques $\mathbf{x}$ and étiquette y. In some applications, we can model the relation between the vecteur de caractéristiques and the étiquette of a point de données as $y = \bar{h}(\mathbf{x}) + \varepsilon$. Here, we use some true underlying hypothèse $\bar{h}$ and a noise term ε which summarizes any modeling or labeling errors. The estimation error incurred by an apprentissage automatique method that learns a hypothèse $\widehat{h}$, e.g., using MRE, is defined as $\widehat{h}(\mathbf{x}) - \bar{h}(\mathbf{x})$, for some vecteur de caractéristiques. For a parametric espace des hypothèses, which consists of hypothèse maps

determined by paramètres du modèle $\mathbf{w}$, we can define the estimation error as $\Delta\mathbf{w} = \widehat{\mathbf{w}} - \overline{\mathbf{w}}$ [20, 40].

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

expectation-maximization (EM) Consider a probabilistic model $\mathbb{P}(\mathbf{z}; \mathbf{w})$ for the point de donnéess $\mathcal{D}$ generated in some apprentissage automatique application. The maximum likelihood estimator for the paramètres du modèle $\mathbf{w}$ is obtained by maximizing $\mathbb{P}(\mathcal{D}; \mathbf{w})$. However, the resulting optimization problem might be computationally challenging. Espérance-maximization approximates the maximum likelihood estimator by introducing a latent VA $\mathbf{z}$ such that maximizing $\mathbb{P}(\mathcal{D}, \mathbf{z}; \mathbf{w})$ would be easier [20, 41, 42]. Since we do not observe $\mathbf{z}$, we need to estimate it from the observed jeu de données $\mathcal{D}$ using a conditional espérance. The resulting estimate $\widehat{\mathbf{z}}$ is then used to compute a new estimate $\widehat{\mathbf{w}}$ by solving $\max_{\mathbf{w}} \mathbb{P}(\mathcal{D}, \widehat{\mathbf{z}}; \mathbf{w})$. The crux is that the conditional espérance $\widehat{\mathbf{z}}$ depends on the paramètres du modèle $\widehat{\mathbf{w}}$, which we have updated based on $\widehat{\mathbf{z}}$. Thus, we have to re-calculate $\widehat{\mathbf{z}}$, which, in turn, results in a new choice $\widehat{\mathbf{w}}$ for the paramètres du modèle. In practice, we repeat the computation of the conditional espérance (i.e., the E-step) and the update of the paramètres du modèle (i.e., the M-step) until some critère d'arrêt is met.

expert apprentissage automatique aims to learn a hypothèse h that ac-

curately predicts the étiquette of a point de données based on its caractéristiques. We measure the prédiction error using some fonction de perte. Ideally, we want to find a hypothèse that incurs minimal perte on any point de données. We can make this informal goal precise via the independent and identically distributed assumption (i.i.d. assumption) and by using the Bayes risk as the baseline for the (average) perte of a hypothèse. An alternative approach to obtaining a baseline is to use the hypothèse h' learned by an existing apprentissage automatique method. We refer to this hypothèse h' as an expert [43]. Regret minimization methods learn a hypothèse that incurs a perte comparable to the best expert [43, 44].

explainability We define the (subjective) explainability of an apprentissage automatique method as the level of simulatability [45] of the prédictions delivered by an apprentissage automatique system to a human user. Quantitative measures for the (subjective) explainability of a trained modèle can be constructed by comparing its prédictions with the prédictions provided by a user on a ensemble de test [45, 46]. Alternatively, we can use probabilistic models for données and measure the explainability of a trained apprentissage automatique modèle via the conditional (differential) entropy of its prédictions, given the user prédictions [47, 48].

explainable empirical risk minimization (EERM) Explainable MRE is an instance of SRM that adds a régularisation term to the average perte in the objective function of MRE. The régularisation term is

chosen to favor hypothèse maps that are intrinsically explainable for a specific user. This user is characterized by their prédictions provided for the point de donnéess in a ensemble d'entraînement [46].

explainable machine learning (explainable ML) Explainable apprentissage automatique methods aim at complementing each prédiction with an explanation of how the prédiction has been obtained. The construction of an explicit explanation might not be necessary if the apprentissage automatique method uses a sufficiently simple (or interpretable) modèle [49].

explanation One approach to make apprentissage automatique methods transparent is to provide an explanation along with the prédiction delivered by an apprentissage automatique method. Explanations can take on many different forms. An explanation could be some natural text or some quantitative measure for the importance of individual caractéristiques of a point de données [50]. We can also use visual forms of explanations, such as intensity plots for image classification [51].

feature learning Consider an apprentissage automatique application with point de donnéess characterized by raw caractéristiques $\mathbf{x} \in \mathcal{X}$. Caractéristique learning refers to the task of learning a map

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

that reads in raw caractéristiques $\mathbf{x} \in \mathcal{X}$ of a point de données and delivers new caractéristiques $\mathbf{x}' \in \mathcal{X}'$ from a new espace des caractéristiques $\mathcal{X}'$. Different caractéristique learning methods are obtained for different

design choices of $\mathcal{X}, \mathcal{X}'$, for a espace des hypothèses $\mathcal{H}$ of potential maps $\mathbf{\Phi}$, and for a quantitative measure of the usefulness of a specific $\mathbf{\Phi} \in \mathcal{H}$. For example, analyse en composantes principales (ACP) uses $\mathcal{X} := \mathbb{R}^d$, $\mathcal{X}' := \mathbb{R}^{d'}$ with $d' < d$, and a espace des hypothèses

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

ACP measures the usefulness of a specific map $\mathbf{\Phi}(\mathbf{x}) = \mathbf{F}\mathbf{x}$ by the minimum linear reconstruction error incurred on a jeu de données,

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

feature matrix Consider a jeu de données $\mathcal{D}$ with m point de donnéess with vecteur de caractéristiquess $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)} \in \mathbb{R}^d$. It is convenient to collect the individual vecteur de caractéristiquess into a caractéristique matrix $\mathbf{X} := \left(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}\right)^T$ of size $m \times d$.

FedAvg FedAvg refers to a family of iterative apprentissage fédéré algorithmes. It uses a server-client setting and alternatves between client-wise local models re-training, followed by the aggregation of updated paramètres du modèle at the server [52].

See also: apprentissage fédéré, algorithme.

FedGD An apprentissage fédéré distributed algorithm that can be implemented as message passing across an réseau d'apprentissage fédéré.

See also: apprentissage fédéré, algorithme, gradient step, méthodes basées sur le gradient.

FedProx FedProx refers to an iterative apprentissage fédéré algorithme that alternates between separately training local models and combining the updated local paramètres du modèle. In contrast to FedAvg, which uses stochastic gradient descent (SGD) to train local models, FedProx uses a proximal operator for the training [53].

FedRelax An apprentissage fédéré distributed algorithm.
See also: apprentissage fédéré, algorithme.

FedSGD An apprentissage fédéré distributed algorithm that can be implemented as message passing across an réseau d'apprentissage fédéré.
See also: apprentissage fédéré, algorithme, gradient step, méthodes basées sur le gradient, SGD.

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

flow-based clustering Flow-based partitionnement de données groups the nodes of an undirected graphe by applying k-moyennes partitionnement de données to node-wise vecteur de caractéristiquess. These vecteur de caractéristiquess are built from network flows between carefully selected sources and destination nodes [54].

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

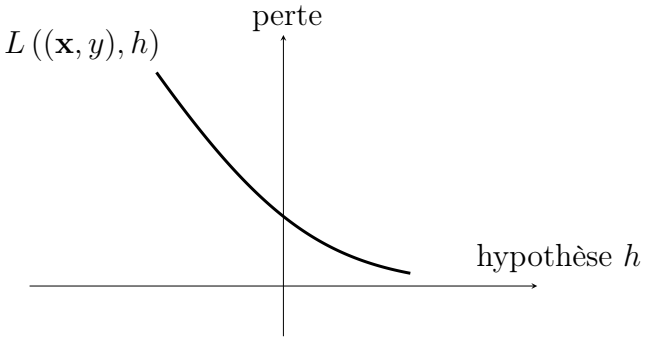


Fig. 9. Une fonction de perte $L\left((\mathbf{x}, y), h\right)$ pour un point de données fixé, de vecteur de caractéristiques $\mathbf{x}$ et d'étiquette y, et une hypothèse variable h. Les méthodes d'apprentissage automatique cherchent à trouver (ou apprendre) une hypothèse minimisant la perte.

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

Gaussian mixture model (GMM) A GMM is a particular type of probabilistic model for a numeric vector $\mathbf{x}$ (e.g., the caractéristiques of a point de données). Within a GMM, the vector $\mathbf{x}$ is drawn from a randomly selected loi normale multivariée $p^{(c)} = \mathcal{N}\left(\boldsymbol{\mu}^{(c)}, \mathbf{C}^{(c)}\right)$ with $c = I$. The index $I \in \{1, \ldots, k\}$ is an VA with probabilities $\mathbb{P}(I = c) = p_c$. Note that a GMM is parametrized by the probability p_c, the moyenne vector $\boldsymbol{\mu}^{(c)}$, and the matrice de covariance $\boldsymbol{\Sigma}^{(\mathbf{c})}$ for each $c = 1, \ldots, k$. GMMs

are widely used for partitionnement de données, density estimation, and as a generative modèle.

Gaussian process (GP) A GP is a collection of VAs $\{f(\mathbf{x})\}_{\mathbf{x}\in\mathcal{X}}$ indexed by input values $\mathbf{x}$ from some input space $\mathcal{X}$, such that for any finite subset $\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(m)} \in \mathcal{X}$, the corresponding VAs $f(\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(m)}$ have a joint multivariate Gaussian distribution:

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

For a fixed input space $\mathcal{X}$, a GP is fully specified (or parametrized) by

- a moyenne function $\mu(\mathbf{x}) = \mathbb{E}\{f(\mathbf{x})\}$,
- and a covariance function $K\big(\mathbf{x},\mathbf{x}'\big) = \mathbb{E}\{\big(f(\mathbf{x}) - \mu(\mathbf{x})\big)\big(f(\mathbf{x}') - \mu(\mathbf{x}')\big)\}$.

Example. We can interpret the temperature distribution across Finland (at a specific point in time) as the réalisation of a GP $f(\mathbf{x})$, where each input $\mathbf{x} = (\text{lat},\text{lon})$ denotes a geographic location. Temperature observations from Finnish Meteorological Institute (FMI) weather stations provide samples of $f(\mathbf{x})$ at specific locations (see Fig. 10). A GP allows to predict the temperature nearby FMI weather stations and to quantify the uncertainty of these predictions.

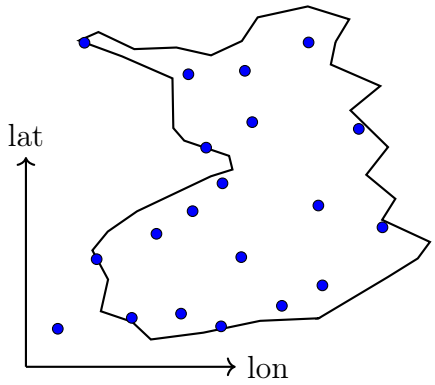


Fig. 10. We can interpret the temperature distribution over Finland as a réalisation of a GP indexed by geographic coordinates and sampled at FMI weather stations (indicated by blue dots).

generalization Many current apprentissage automatique (and intelligence artificielle (IA)) systems are based on MRE: At their core, they train a modèle (i.e., learn a hypothèse $\hat{h} \in \mathcal{H}$) by minimizing the average perte (or risque empirique) on some point de donnéess $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}$, which serve as a ensemble d'entraînement $\mathcal{D}^{(\text{train})}$. Generalization refers to an apprentissage automatique method's ability to perform well outside the ensemble d'entraînement. Any mathematical theory of generalization needs some mathematical concept for the "outside the ensemble d'entraînement." For example, statistical learning theory uses a probabilistic model such as the i.i.d. assumption for données generation: the point de donnéess in the ensemble d'entraînement are i.i.d. réalisations of some underlying loi de probabilité $p(\mathbf{z})$. A probabilistic model allows us to explore the outside of the ensemble d'entraînement by drawing

additional i.i.d. réalisations from $p(\mathbf{z})$. Moreover, using the i.i.d. assumption allows us to define the risque of a trained modèle $\hat{h} \in \mathcal{H}$ as the expected perte $\bar{L}\big(\hat{h}\big)$. What is more, we can use concentration bounds or convergence results for sequences of i.i.d. VAs to bound the deviation between the risque empirique $\widehat{L}\big(\hat{h}|\mathcal{D}^{(\text{train})}\big)$ of a trained modèle and its risque [55]. It is possible to study generalization also without using probabilistic models. For example, we could use (deterministic) perturbations of the point de donnéess in the ensemble d'entraînement to study its outside. In general, we would like the trained modèle to be robust, i.e., its prédictions should not change too much for small perturbations of a point de données. Consider a trained modèle for detecting an object in a smartphone snapshot. The detection result should not change if we mask a small number of randomly chosen pixels in the image [56].

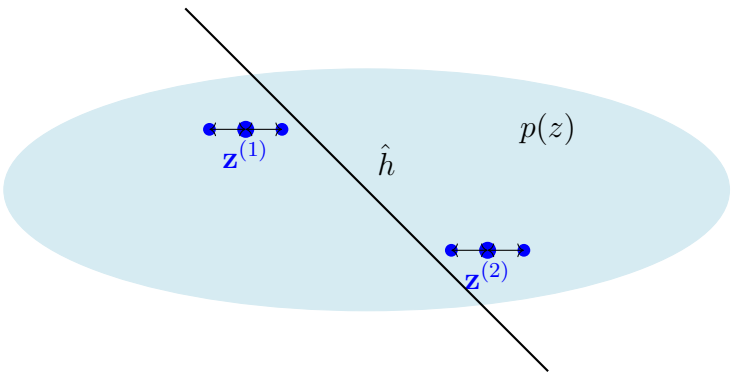


Fig. 11. Two point de donnéess $\mathbf{z}^{(1)}, \mathbf{z}^{(2)}$ that are used as a ensemble d'entraînement to learn a hypothèse $\hat{h}$ via MRE. We can evaluate $\hat{h}$ outside $\mathcal{D}^{(\text{train})}$ either by an i.i.d. assumption with some underlying loi de probabilité $p(\mathbf{z})$ or by perturbing the point de donnéess.

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

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

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

generalized total variation minimization (GTVMin) GTV minimization is an instance of regularized empirical risk minimization (RERM) using the GTV of local paramètres du modèle as a regularizer [57].

geometric median (GM) The GM of a set of input vectors $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}$ in $\mathbb{R}^d$ is a point $\mathbf{z} \in \mathbb{R}^d$ that minimizes the sum of distances to the vectors [27] such that

$$\mathbf{z} \in \operatorname*{argmin}_{\mathbf{y} \in \mathbb{R}^d} \sum_{r=1}^{m} \left\| \mathbf{y} - \mathbf{x}^{(r)} \right\|_2 . \tag{3}$$

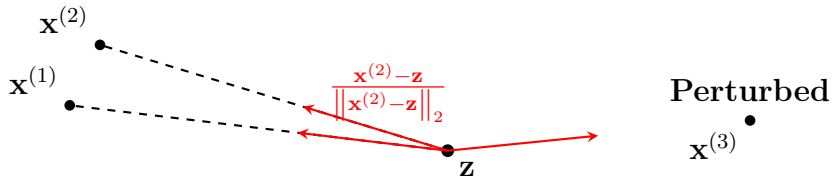


Fig. 12. Consider a solution $\mathbf{z}$ of (3) that does not coincide with any of the input vectors. The optimality condition for (3) requires that the unit vectors from $\mathbf{z}$ to the input vectors sum to zero.

Figure 12 illustrates a fundamental property of the GM: If $\mathbf{z}$ does not coincide with any of the input vectors, then the unit vectors pointing from $\mathbf{z}$ to each $\mathbf{x}^{(r)}$ must sum to zero - this is the zero-subgradient (optimality) condition of (3). It turns out that the solution to (3) cannot be arbitrarily pulled away from trustworthy input vectors as long as they are the majority [58, Th. 2.2].

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].

gradient step Given a dérivable real-valued function $f(\cdot) : \mathbb{R}^d \to \mathbb{R}$ and a vector $\mathbf{w} \in \mathbb{R}^d$, the gradient step updates $\mathbf{w}$ by adding the scaled negative gradient $\nabla f(\mathbf{w})$ to obtain the new vector (see Figure 13)

$$\widehat{\mathbf{w}} := \mathbf{w} - \eta \nabla f(\mathbf{w}). \tag{4}$$

Mathematically, the gradient step is a (typically non-linear) operator $\mathcal{T}^{(f,\eta)}$ that is parametrized by the function f and the taille de pas η.

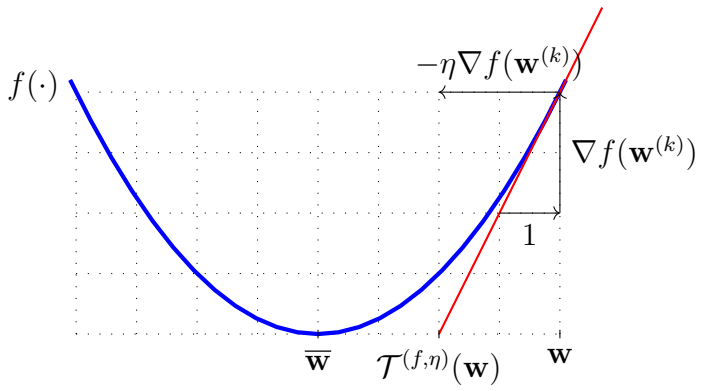


Fig. 13. The basic gradient step (4) maps a given vector $\mathbf{w}$ to the updated vector $\mathbf{w}'$. It defines an operator $\mathcal{T}^{(f,\eta)}(\cdot) : \mathbb{R}^d \to \mathbb{R}^d : \mathbf{w} \mapsto \widehat{\mathbf{w}}$.

Note that the gradient step (4) optimizes locally - in a voisinage whose size is determined by the taille de pas η - a linear approximation to the function $f(\cdot)$. A natural generalization of (4) is to locally optimize the function itself - instead of its linear approximation - such that

$$\widehat{\mathbf{w}} = \operatorname*{argmin}_{\mathbf{w}' \in \mathbb{R}^d} f(\mathbf{w}') + (1/\eta) \left\| \mathbf{w} - \mathbf{w}' \right\|_2^2 . \tag{5}$$

We intentionally use the same symbol η for the parameter in (5) as we used for the taille de pas in (4). The larger the η we choose in (5), the more progress the update will make towards reducing the function value $f(\widehat{\mathbf{w}})$. Note that, much like the gradient step (4), also the update (5) defines a (typically non-linear) operator that is parametrized by the function $f(\cdot)$ and the parameter η. For a convexe function $f(\cdot)$, this operator is known as the proximal operator of $f(\cdot)$ [59].

grand modèle de langage (GML) Large language modèles is an umbrella

term for apprentissage automatique methods that process and generate human-like text. These methods typically use deep nets with billions (or even trillions) of paramètres. A widely used choice for the network architecture is referred to as Transformers [60]. The training of large language modèles is often based on the task of predicting a few words that are intentionally removed from a large text corpus. Thus, we can construct labeled datapoints simply by selecting some words of a text as étiquettes and the remaining words as caractéristiques of point de donnéess. This construction requires very little human supervision and allows for generating sufficiently large ensemble d'entraînements for large language modèles.

graph clustering Graphe partitionnement de données aims at partitionnement de données point de donnéess that are represented as the nodes of a graphe $\mathcal{G}$. The edges of $\mathcal{G}$ represent pairwise similarities between point de donnéess. Sometimes we can quantify the extend of these similarities by an edge weight [54, 61].

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

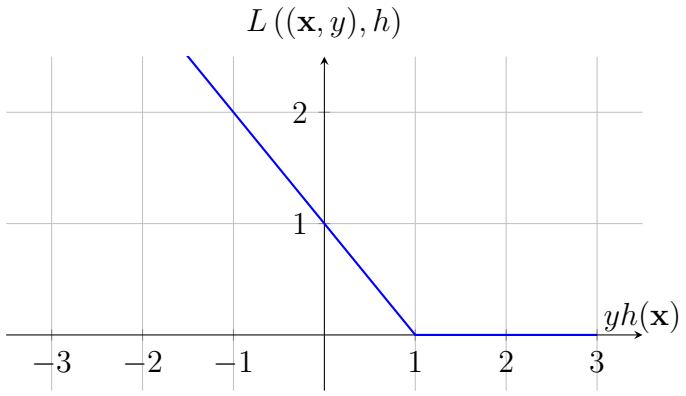
hard clustering Hard partitionnement de données refers to the task of partitioning a given set of point de donnéess into (a few) non-overlapping clusters. The most widely used hard partitionnement de données method is k-moyennes.

high-dimensional regime The high-dimensional regime of MRE is characterized by the dimension effective of the modèle being larger than the taille d'échantillon, i.e., the number of (labeled) point de donnéess in the ensemble d'entraînement. For example, linear regression methods operate in the high-dimensional regime whenever the number d of caractéristiques used to characterize point de donnéess exceeds the number of point de donnéess in the ensemble d'entraînement. Another example of apprentissage automatique methods that operate in the high-dimensional regime is large RNAs, which have far more tunable poids (and bias terms) than the total number of point de donnéess in the ensemble d'entraînement. High-dimensional statistics is a recent main thread of probabilité theory that studies the behavior of apprentissage automatique methods in the high-dimensional regime [63, 64].

Hilbert space A Hilbert space is a complete inner product space [65]. That is, it is a vector space equipped with an inner product between pairs of vectors, and it satisfies the additional requirement of completeness: every Cauchy sequence of vectors converges to a limit within the space. A canonical example of a Hilbert space is the Euclidean space $\mathbb{R}^d$, for some dimension d, consisting of vectors $\mathbf{u} = \big(u_1, \ldots, u_d\big)^T$ and the standard inner product $\mathbf{u}^T\mathbf{v}$.

hinge loss Consider a point de données characterized by a vecteur de caractéristiques $\mathbf{x} \in \mathbb{R}^d$ and a binary étiquette $y \in \{-1, 1\}$. The hinge perte incurred by a real-valued hypothèse map $h(\mathbf{x})$ is defined as

$$L\left((\mathbf{x}, y), h\right) := \max\{0, 1 - yh(\mathbf{x})\}. \tag{6}$$



A regularized variant of the hinge perte is used by the support vector machine (SVM) [66].

histogram Consider a jeu de données $\mathcal{D}$ that consists of m point de donnéess $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}$, each of them belonging to some cell $[-U, U] \times \ldots \times [-U, U] \subseteq \mathbb{R}^d$ with side length U. We partition this cell evenly into smaller elementary cells with side length Δ. The histogram of $\mathcal{D}$ assigns each elementary cell to the corresponding fraction of point de donnéess in $\mathcal{D}$ that fall into this elementary cell. A visual example of such a histogram is provided in Figure 14.

horizontal federated learning (HFL) HFL uses jeu de données locals constituted by different point de donnéess but uses the same caractéris-

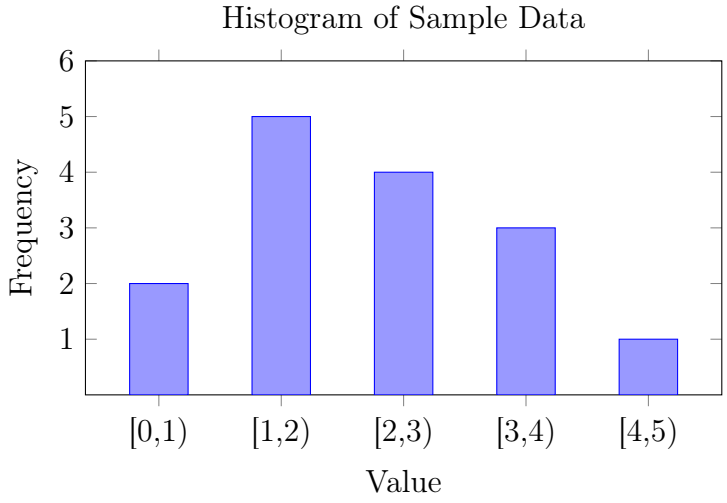


Fig. 14. A histogram representing the frequency of data points falling within discrete value ranges (bins). Each bar height shows the count of samples in the corresponding interval.

tiques to characterize them [67]. For example, weather forecasting uses a network of spatially distributed weather (observation) stations. Each weather station measures the same quantities, such as daily temperature, air pressure, and precipitation. However, different weather stations measure the characteristics or caractéristiques of different spatiotemporal regions. Each spatiotemporal region represents an individual point de données, each characterized by the same caractéristiques (e.g., daily temperature or air pressure).

See also: apprentissage fédéré, vertical federated learning (vertical FL), clustered federated learning (CFL).

Huber loss The Huber perte unifies the squared error loss and the absolute error loss.

Huber regression Huber régression refers to MRE-based methods that use the Huber loss as a measure of the prédiction error. Two important special cases of Huber régression are least absolute deviation regression and linear regression. Tuning the threshold parameter of the Huber loss allows the user to trade the robustness of the absolute error loss against the computational benefits of the lisse squared error loss.

hypothèse Une hypothèse désigne une application (ou fonction) $h : \mathcal{X} \rightarrow \mathcal{Y}$ allant de l'espace des caractéristiques $\mathcal{X}$ vers l'label space $\mathcal{Y}$. 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).

independent and identically distributed assumption (i.i.d. assumption) The i.i.d. assumption interprets point de donnéess of a jeu de données as the réalisations of i.i.d. VAs.

indépendantes et identiquement distribuées (i.i.d.) It can be useful to interpret point de donnéess $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}$ as réalisations of i.i.d. VAs with a common loi de probabilité. If these VAs are continuous-valued, their joint probability density function (pdf) is $p\big(\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}\big) = \prod_{r=1}^{m} p\big(\mathbf{z}^{(r)}\big)$, with $p(\mathbf{z})$ being the common marginal pdf of the underlying VAs.

intelligence artificielle (IA) AI refers to systems that behave rationally

in the sense of maximizing a long-term reward. The apprentissage automatique-based approach to AI is to train a modèle for predicting optimal actions. These predictions are computed from observations about the state of the environment. The choice of fonction de perte sets AI applications apart from more basic apprentissage automatique applications. AI systems rarely have access to a labeled ensemble d'entraînement that allows the average perte to be measured for any possible choice of paramètres du modèle. Instead, AI systems use observed reward signals to obtain a (point-wise) estimate for the perte incurred by the current choice of paramètres du modèle.

interpretability An apprentissage automatique method is interpretable for a specific user if they can well anticipate the prédictions delivered by the method. The notion of interpretability can be made precise using quantitative measures of the uncertainty about the prédictions [47].

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 [68], [69], [70], [71].

L'un des modèles de données les plus utilisés est le modèle relationnel, qui organise les données sous forme de tableau (ou relation) [35], [68]. 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'label space.

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 trustworthy artificial intelligence (trustworthy AI).
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 [72].

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 probabilistic models ou de l'infrastructure de communication, et sont représentées par les arêtes d'un réseau d'apprentissage fédéré.

kernel method A noyau method is an apprentissage automatique method that uses a noyau K to map the original (raw) vecteur de caractéristiques $\mathbf{x}$ of a point de données to a new (transformed) vecteur de caractéristiques $\mathbf{z} = K\big(\mathbf{x}, \cdot\big)$ [66, 73]. The motivation for transforming the vecteur de caractéristiquess is that, by using a suitable noyau, the point de donnéess have a "more pleasant" geometry in the transformed espace des caractéristiques. For example, in a binary classification problem, using transformed vecteur de caractéristiquess $\mathbf{z}$ might allow us to use modèle linéaires, even if the point de donnéess are not linearly separable in the original espace des caractéristiques (see Figure 16).

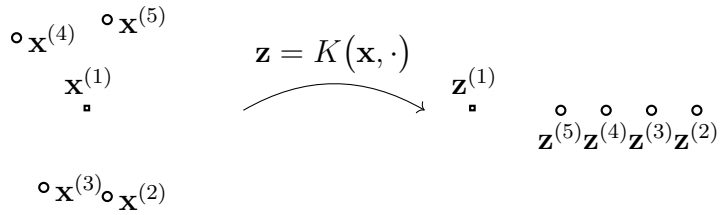


Fig. 16. Five point de donnéess characterized by vecteur de caractéristiquess $\mathbf{x}^{(r)}$ and étiquettes $y^{(r)} \in \{\circ, \square\}$, for $r = 1, \ldots, 5$. With these vecteur de caractéristiquess, there is no way to separate the two classes by a straight line (representing the frontière de décision of a linear classifier). In contrast, the transformed vecteur de caractéristiquess $\mathbf{z}^{(r)} = K\big(\mathbf{x}^{(r)}, \cdot\big)$ allow us to separate the point de donnéess using a linear classifier.

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

label space Consider an apprentissage automatique application that involves points de données characterized by caractéristiques and étiquettes. The étiquette space is constituted by all potential values that the étiquette of a point de données can take on. Régression methods, aiming at predicting numeric étiquettes, often use the étiquette space $\mathcal{Y} = \mathbb{R}$. Binary classification methods use a étiquette space that consists of two different elements, e.g., $\mathcal{Y} = \{-1, 1\}$, $\mathcal{Y} = \{0, 1\}$, or .

See also: apprentissage automatique, point de données, caractéristique, étiquette, régression, classification.

labeled datapoint A point de données whose étiquette is known or has been determined by some means which might require human labor.

law of large numbers The law of large numbers refers to the convergence of the average of an increasing (large) number of i.i.d. VAs to the moyenne of their common loi de probabilité. Different instances of the law of large numbers are obtained by using different notions of convergence [75].

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

least absolute shrinkage and selection operator (Lasso) The Lasso is an instance of SRM. It learns the poids $\mathbf{w}$ of a linear map $h(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ based on a ensemble d'entraînement. Lasso is obtained from linear regression by adding the scaled ℓ_1-norme $\alpha \left\|\mathbf{w}\right\|_1$ to the average squared error loss incurred on the ensemble d'entraînement.

linear classifier Consider point de donnéess characterized by numeric caractéristiques $\mathbf{x} \in \mathbb{R}^d$ and a étiquette $y \in \mathcal{Y}$ from some finite label space $\mathcal{Y}$. A linear classifier is characterized by having région de décisions that are separated by hyperplanes in $\mathbb{R}^d$ [8, Ch. 2].

linear regression Linear régression aims to learn a linear hypothèse map to predict a numeric étiquette based on the numeric caractéristiques of a point de données. The quality of a linear hypothèse map is measured using the average squared error loss incurred on a set of labeled

datapoints, which we refer to as the ensemble d'entraînement.

lisse (ou régulière) A real-valued function $f : \mathbb{R}^d \to \mathbb{R}$ is smooth if it is dérivable and its gradient $\nabla f(\mathbf{w})$ is continuous at all $\mathbf{w} \in \mathbb{R}^d$ [76, 77]. A smooth function f is referred to as β-smooth if the gradient $\nabla f(\mathbf{w})$ is Lipschitz continuous with Lipschitz constant β, i.e.,

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

The constant β quantifies the amount of smoothness of the function f: the smaller the β, the smoother f is. Optimization problems with a smooth objective function can be solved effectively by méthodes basées sur le gradient. Indeed, méthodes basées sur le gradient approximate the objective function locally around a current choice $\mathbf{w}$ using its gradient. This approximation works well if the gradient does not change too rapidly. We can make this informal claim precise by studying the effect of a single gradient step with taille de pas $\eta = 1/\beta$ (see Figure 17).

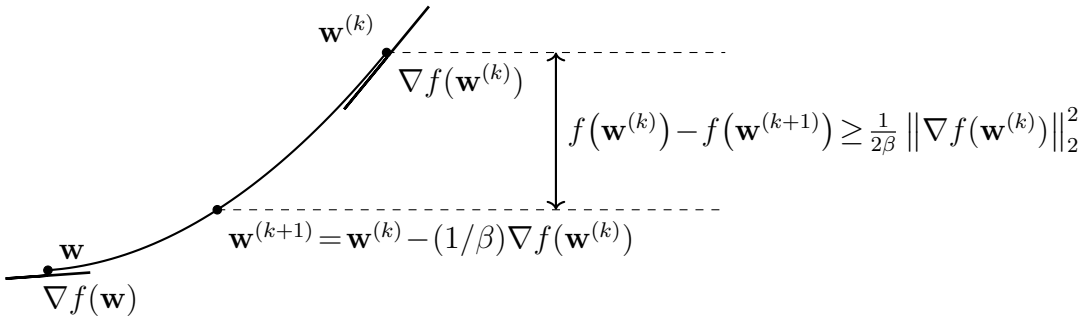


Fig. 17. Consider an objective function $f(\mathbf{w})$ that is β-smooth. Taking a gradient step, with taille de pas $\eta = 1/\beta$, decreases the objective by at least $\frac{1}{2\beta} \big\|\nabla f(\mathbf{w}^{(k)})\big\|_2^2$ [76–78]. Note that the taille de pas $\eta = 1/\beta$ becomes larger for smaller β. Thus, for smoother objective functions (i.e., those with smaller β), we can take larger steps.

Local Interpretable Model-agnostic Explanations (LIME) Consider a trained modèle (or learnt hypothèse) $\widehat{h} \in \mathcal{H}$, which maps the vecteur de caractéristiques of a point de données to the prédiction $\widehat{y} = \widehat{h}$. Local Interpretable Model-agnostic Explanations (LIME) is a technique for explaining the behaviour of $\widehat{h}$, locally around a point de données with vecteur de caractéristiques $\mathbf{x}^{(0)}$ [79]. The explanation is given in the form of a local approximation $g \in \mathcal{H}'$ of $\widehat{h}$ (see Fig.). This approximation can be obtained by an instance of MRE with carefully designed ensemble d'entraînement. In particular, the ensemble d'entraînement consists of point de donnéess with vecteur de caractéristiques $\mathbf{x}$ close to $\mathbf{x}^{(0)}$ and the (pseudo-)label $\widehat{h}(\mathbf{x})$. Note that we can use a different modèle $\mathcal{H}'$ for the approximation than the original modèle $\mathcal{H}$. For example, we can

use a arbre de décision to approximate (locally) a deep net. Another widely-used choice for $\mathcal{H}'$ is the modèle linéaire.

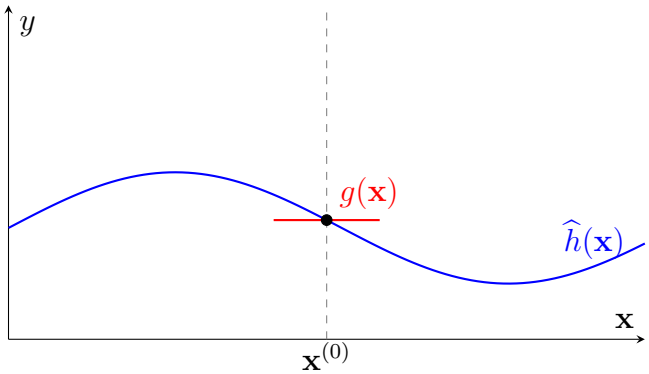


Fig. 18. To explain a trained modèle $\widehat{h} \in \mathcal{H}$, around a given vecteur de caractéristiques $\mathbf{x}^{(0)}$, we can use a local approximation $g \in \mathcal{H}'$.

local model Consider a collection of jeu de données locals that are assigned to the nodes of an réseau d'apprentissage fédéré. A local modèle $\mathcal{H}^{(i)}$ is a espace des hypothèses assigned to a node $i \in \mathcal{V}$. Different nodes might be assigned different espace des hypothèsess, i.e., in general $\mathcal{H}^{(i)} \neq \mathcal{H}^{(i')}$ for different nodes $i, i' \in \mathcal{V}$.

logistic loss Consider a point de données characterized by the caractéristiques $\mathbf{x}$ and a binary étiquette $y \in \{-1, 1\}$. We use a real-valued hypothèse h to predict the étiquette y from the caractéristiques $\mathbf{x}$. The logistic perte incurred by this prédiction is defined as

$$L\left((\mathbf{x}, y), h\right) := \log(1 + \exp(-yh(\mathbf{x}))). \tag{7}$$

Carefully note that the expression (7) for the logistic perte applies only

for the label space $\mathcal{Y} = \{-1, 1\}$ and when using the thresholding rule (1).

logistic regression Logistic régression learns a linear hypothèse map (or classifier) $h(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ to predict a binary étiquette y based on the numeric vecteur de caractéristiques $\mathbf{x}$ of a point de données. The quality of a linear hypothèse map is measured by the average logistic loss on some labeled datapoints (i.e., the ensemble d'entraînement).

loi 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 (ou distribution) 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 pdf $p(x)$ 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, 19].

loi normale multivariée La loi normale multivariée $\mathcal{N}(\mathbf{m}, \mathbf{C})$ est un probabilistic model important pour les vecteurs de caractéristiques numériques. C'est une famille de lois de probabilité pour une VA vectorielle $\mathbf{x} \in \mathbb{R}^d$ [7], [19], [80]. Chaque membre (i.e. une loi de probabilité) de cette famille est spécifié par sa moyenne $\mathbf{m}$ et sa matrice de covariance $\mathbf{C}$. Si la matrice de covariance est inversible, la loi de probabilité de $\mathbf{x}$ peut s'écrire :

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

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

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 linear regression [36], [81]. 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 pseudoinverse de Moore–Penrose, notée $\mathbf{A}^+$, fournit une généralisation unifiée de la matrice inverse [3].

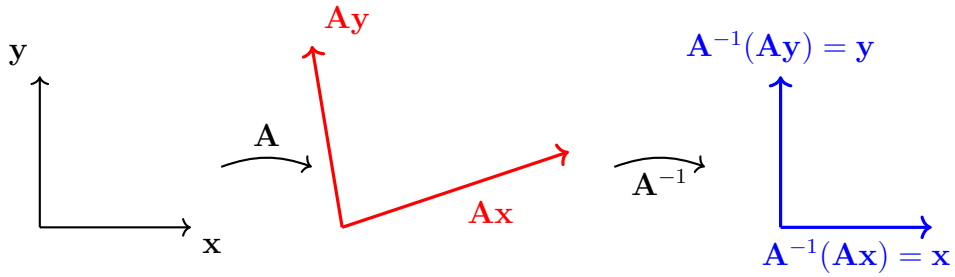


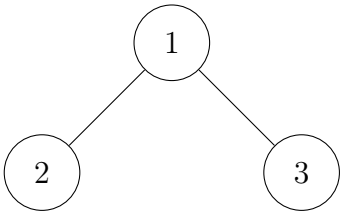
Fig. 19. Une matrice $\mathbf{A}$ représente une transformation linéaire de $\mathbb{R}^2$. La matrice inverse $\mathbf{A}^{-1}$ représente la transformation inverse.

Voir aussi : déterminant, linear regression, pseudoinverse.

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 [61, 82]. Elle est définie terme à terme par (voir Figure 20)

$$L_{i,i'}^{(\mathcal{G})} := \begin{cases} -A_{i,i'} & \text{pour } i \neq i', \{i, i'\} \in \mathcal{E}, \\ \sum_{i'' \neq i} A_{i,i''} & \text{pour } i = i', \\ 0 & \text{sinon.} \end{cases} \tag{8}$$

Ici, $A_{i,i'}$ désigne le edge weight 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. 20. 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 The maximum of a set $\mathcal{A} \subseteq \mathbb{R}$ of real numbers is the greatest element in that set, if such an element exists. A set $\mathcal{A}$ has a maximum if it is bounded above and attains its supremum (or least upper bound) [2, Sec. 1.4].

maximum likelihood Consider point de donnéess $\mathcal{D} = \{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}\}$ that are interpreted as the réalisations of i.i.d. VAs with a common loi de probabilité $\mathbb{P}(\mathbf{z}; \mathbf{w})$ which depends on the paramètres du modèle $\mathbf{w} \in \mathcal{W} \subseteq \mathbb{R}^n$. Maximum likelihood methods learn paramètres du modèle $\mathbf{w}$ by maximizing the probability (density) $\mathbb{P}(\mathcal{D}; \mathbf{w}) = \prod_{r=1}^{m} \mathbb{P}(\mathbf{z}^{(r)}; \mathbf{w})$ of the observed jeu de données. Thus, the maximum likelihood estimator is a solution to the optimization problem $\max_{\mathbf{w}\in\mathcal{W}} \mathbb{P}(\mathcal{D}; \mathbf{w})$.

mean squared estimation error (MSEE) Consider an apprentissage automatique method that learns paramètres du modèle $\widehat{\mathbf{w}}$ based on some jeu de données $\mathcal{D}$. If we interpret the point de donnéess in $\mathcal{D}$ as i.i.d. réalisations of an VA $\mathbf{z}$, we define the estimation error $\Delta\mathbf{w} := \widehat{w} - \overline{\mathbf{w}}$. Here, $\overline{\mathbf{w}}$ denotes the true paramètres du modèle of the loi de probabilité of $\mathbf{z}$. The moyenne squared estimation error is defined as the

espérance $\mathbb{E}\{\|\Delta \mathbf{w}\|^2\}$ of the squared Euclidean norme of the estimation error [17, 40].

minimisation du risque empirique (MRE) Risque empirique minimization is the optimization problem of finding a hypothèse (out of a modèle) with the minimum average perte (or risque empirique) on a given jeu de données $\mathcal{D}$ (i.e., the ensemble d'entraînement). Many apprentissage automatique methods are obtained from risque empirique via specific design choices for the jeu de données, modèle, and perte [8, Ch. 3].

minimum Given a set of real numbers, the minimum is the smallest of those numbers.

missing data Consider a jeu de données constituted by point de donnéess collected via some physical appareil. Due to imperfections and failures, some of the caractéristique or étiquette values of point de donnéess might be corrupted or simply missing. Données imputation aims at estimating these missing values [83]. We can interpret données imputation as an apprentissage automatique problem where the étiquette of a point de données is the value of the corrupted caractéristique.

model inversion TBD.

model selection In apprentissage automatique, modèle selection refers to the process of choosing between different candidate modèles. In its most basic form, modèle selection amounts to: 1) training each candidate modèle; 2) computing the erreur de validation for each trained modèle; and 3) choosing the modèle with the smallest erreur de validation [8, Ch.

6].

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], [55]. Cependant, ce terme est également utilisé dans d'autres domaines avec des significations différentes. Par exemple, un probabilistic model 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,

$$\mathcal{H}^{(d)} := \left\{ h(\mathbf{x}) = \mathbf{w}^T \mathbf{x} : \mathbf{w} \in \mathbb{R}^d \right\}. \tag{9}$$

Notons que (9) 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 computational aspects (par exemple, réduire d signifie moins de calcul), les statistical aspects (par exemple, augmenter d peut réduire l'erreur de prédiction) et l'interpretability. Un modèle linéaire utilisant peu de caractéristiques soigneusement sélectionnées a tendance à être considéré comme plus interprétable [49, 79].

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

multi-armed bandit A multi-armed bandit (MAB) problem models a repeated decision-making scenario in which, at each time step k, a learner must choose one out of several possible actions, often referred to as arms, from a finite set $\mathcal{A}$. Each arm $a \in \mathcal{A}$ yields a stochastic reward $r^{(a)}$ drawn from an unknown loi de probabilité with moyenne $\mu^{(a)}$. The learner's goal is to maximize the cumulative reward over time by strategically balancing exploration (gathering information about uncertain arms) and exploitation (selecting arms known to perform well). This balance is quantified by the notion of regret, which measures the performance gap between the learner's strategy and the optimal strategy that always selects the best arm. MAB problems form a foundational model in online learning, reinforcement learning, and sequential experimental design [84].

mutual information (MI) The MI $I(\mathbf{x}; y)$ between two VAs $\mathbf{x}$, y defined

on the same espace probabilisé is given by [74]

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

It is a measure of how well we can estimate y based solely on $\mathbf{x}$. A large value of $I(\mathbf{x}; y)$ indicates that y can be well predicted solely from $\mathbf{x}$. This prédiction could be obtained by a hypothèse learned by an MRE-based apprentissage automatique method.

méthodes basées sur le gradient Les méthodes basées sur le gradient sont des techniques itératives pour trouver le minimum (ou le maximum) d'une objective function 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 objective function. Comme leur nom l'indique, les méthodes basées sur le gradient utilisent les gradients de la objective function 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.

nearest neighbor (NN) NN methods learn a hypothèse $h : \mathcal{X} \to \mathcal{Y}$ whose function value $h(\mathbf{x})$ is solely determined by the nearest voisins within a given jeu de données. Different methods use different metrics for determining the nearest voisins. If point de donnéess are characterized by numeric vecteur de caractéristiquess, we can use their Euclidean distances as the metric.

networked data Networked données consists of jeu de données locals that are related by some notion of pairwise similarity. We can represent networked données using a graphe whose nodes carry jeu de données locals and edges encode pairwise similarities. One example of networked données arises in apprentissage fédéré applications where jeu de données locals are generated by spatially distributed appareils.

networked exponential families (nExpFam) A collection of exponential families, each of them assigned to a node of an réseau d'apprentissage fédéré. The paramètres du modèle are coupled via the network structure by requiring them to have a small GTV [85].

networked federated learning (NFL) Networked apprentissage fédéré refers to methods that learn personalized modèles in a distributed fashion. These methods learn from jeu de données locals that are related by an intrinsic network structure.

networked model A networked modèle over an réseau d'apprentissage fédéré $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ assigns a local model (i.e., a espace des hypothèses) to each node $i \in \mathcal{V}$ of the réseau d'apprentissage fédéré $\mathcal{G}$.

node degree The degree $d^{(i)}$ of a node $i \in \mathcal{V}$ in an undirected graphe is the number of its voisins, i.e., $d^{(i)} := \left|\mathcal{N}^{(i)}\right|$.

non-smooth We refer to a function as non-smooth if it is not lisse [76].

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 [37].

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 kernel methods 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 [66, 73].

objective function An objective function is a map that assigns each value of an optimization variable, such as the paramètres du modèle $\mathbf{w}$ of a hypothèse $h^{(\mathbf{w})}$, to an objective value $f(\mathbf{w})$. The objective value $f(\mathbf{w})$ could be the risque or the risque empirique of a hypothèse $h^{(\mathbf{w})}$.

online algorithm An online algorithme processes input données incrementally, receiving point de donnéess sequentially and making decisions or producing outputs (or decisions) immediately without having access to the entire input in advance [43], [44]. Unlike an offline algorithme, which has the entire input available from the start, an online algorithme must handle uncertainty about future inputs and cannot revise past decisions. Similar to an offline algorithme, we also represent an online algorithme formally as a collection of possible executions. However, the

execution sequence for an online algorithme has a distinct structure:

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

Each execution begins from an initial state (i.e., in_1) and proceeds through alternating computational steps, outputs (or decisions), and inputs. Specifically, at step k, the algorithme performs a computational step s_k, generates an output out_k, and then subsequently receives the next input (point de données) in_{k+1}. A notable example of an online algorithme in apprentissage automatique is online gradient descent (online GD), which incrementally updates paramètres du modèle as new point de donnéess arrive.

See also: online learning, online GD.

online gradient descent (online GD) Consider an apprentissage automatique method that learns paramètres du modèle $\mathbf{w}$ from some parameter space $\mathcal{W} \subseteq \mathbb{R}^d$. The learning process uses point de donnéess $\mathbf{z}^{(t)}$ that arrive at consecutive time-instants $t = 1, 2, \ldots$. Let us interpret the point de donnéess $\mathbf{z}^{(t)}$ as i.i.d. copies of an VA $\mathbf{z}$. The risque $\mathbb{E}\{L(\mathbf{z}, \mathbf{w})\}$ of a hypothèse $h^{(\mathbf{w})}$ can then (under mild conditions) be obtained as the limit $\lim_{T\to\infty}(1/T)\sum_{t=1}^{T} L\left(\mathbf{z}^{(t)}, \mathbf{w}\right)$. We might use this limit as the objective function for learning the paramètres du modèle $\mathbf{w}$. Unfortunately, this limit can only be evaluated if we wait infinitely long in order to collect all point de donnéess. Some apprentissage automatique applications require methods that learn online: as soon as a new point de données $\mathbf{z}^{(t)}$ arrives at time t, we update the current paramètres du modèle $\mathbf{w}^{(t)}$. Note that the new point de données $\mathbf{z}^{(t)}$ contributes

the component $L\left(\mathbf{z}^{(t)}, \mathbf{w}\right)$ to the risque. As its name suggests, online descente de gradient updates $\mathbf{w}^{(t)}$ via a (projected) gradient step

$$\mathbf{w}^{(t+1)} := P_{\mathcal{W}}\big(\mathbf{w}^{(t)} - \eta_t \nabla_{\mathbf{w}} L\left(\mathbf{z}^{(t)}, \mathbf{w}\right)\big). \tag{10}$$

Note that (10) is a gradient step for the current component $L\left(\mathbf{z}^{(t)}, \cdot\right)$ of the risque. The update (10) ignores all the previous components $L\left(\mathbf{z}^{(t')}, \cdot\right)$, for $t' < t$. It might therefore happen that, compared to $\mathbf{w}^{(t)}$, the updated paramètres du modèle $\mathbf{w}^{(t+1)}$ increase the retrospective average perte $\sum_{t'=1}^{t-1} L\left(\mathbf{z}^{(t')}, \cdot\right)$. However, for a suitably chosen taux d'apprentissage η_t, online descente de gradient can be shown to be optimal in practically relevant settings. By optimal, we mean that the paramètres du modèle $\mathbf{w}^{(T+1)}$ delivered by online descente de gradient after observing T point de donnéess $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(T)}$ are at least as good as those delivered by any other learning method [44, 86].

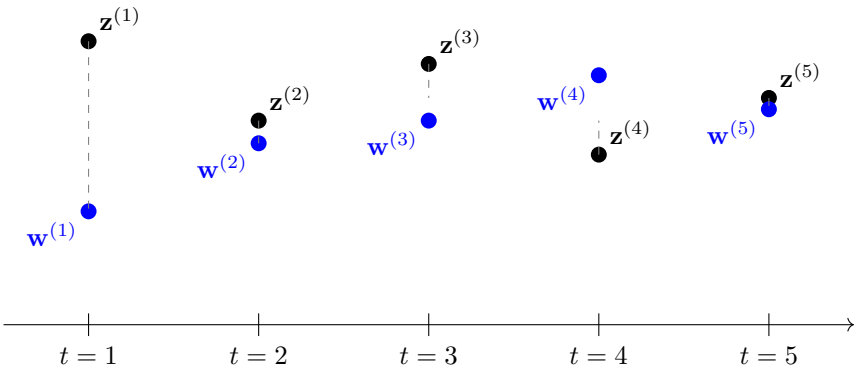


Fig. 21. An instance of online descente de gradient that updates the paramètres du modèle $\mathbf{w}^{(t)}$ using the point de données $\mathbf{z}^{(t)} = x^{(t)}$ arriving at time t. This instance uses the squared error loss $L\left(\mathbf{z}^{(t)}, w\right) = (x^{(t)} - w)^2$.

online learning Some apprentissage automatique methods are designed to process données in a sequential manner, updating their paramètres du modèle as new point de donnéess become available—one at a time. A typical example is time series data, such as daily minimum and maximum temperatures recorded by a FMI weather station. These values form a chronological sequence of observations. In online learning, the hypothèse (or its paramètres du modèle) is refined incrementally with each newly observed point de données, without revisiting past données.

See also online GD, online algorithm.

optimism in the face of uncertainty apprentissage automatique methods learn paramètres du modèle $\mathbf{w}$ according to some performance criterion $\bar{f}(\mathbf{w})$. However, they usually cannot access $\bar{f}(\mathbf{w})$ directly but rely on an estimate (or approximation) $f(\mathbf{w})$ of $\bar{f}(\mathbf{w})$. As a case in point, MRE-based methods use the average perte on a given jeu de données (i.e., the ensemble d'entraînement) as an estimate for the risque of a hypothèse. Using a probabilistic model, one can construct a confidence interval $\left[l^{(\mathbf{w})}, u^{(\mathbf{w})}\right]$ for each choice $\mathbf{w}$ for the paramètres du modèle. One simple construction is $l^{(\mathbf{w})} := f(\mathbf{w}) - \sigma/2$, $u^{(\mathbf{w})} := f(\mathbf{w}) + \sigma/2$, with σ being a measure of the (expected) deviation of $f(\mathbf{w})$ from $\bar{f}(\mathbf{w})$. We can also use other constructions for this interval as long as they ensure that $\bar{f}(\mathbf{w}) \in \left[l^{(\mathbf{w})}, u^{(\mathbf{w})}\right]$ with a sufficiently high probability. An optimist chooses the paramètres du modèle according to the most favourable - yet still plausible - value $\tilde{f}(\mathbf{w}) := l^{(\mathbf{w})}$ of the performance criterion. Two examples of apprentissage automatique methods that use such an

optimistic construction of an objective function are SRM [55, Ch. 11] and upper confidence bound (UCB) methods for sequential decision making [84, Sec. 2.2].

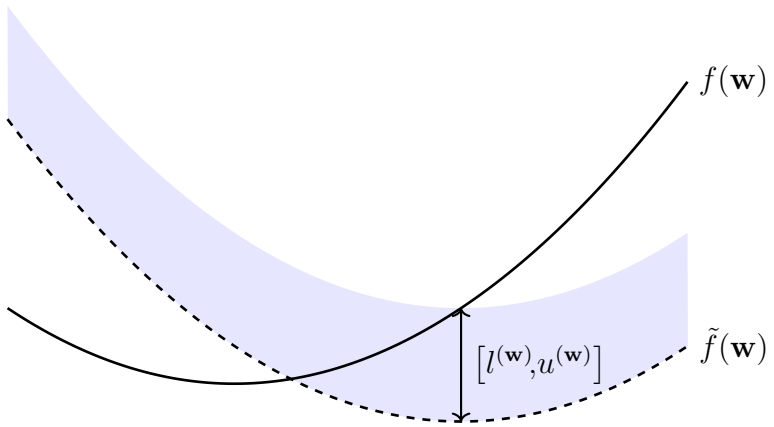


Fig. 22. apprentissage automatique methods learn paramètres du modèle $\mathbf{w}$ by using some estimate of $f(\mathbf{w})$ for the ultimate performance criterion $\bar{f}(\mathbf{w})$. Using a probabilistic model, one can use $f(\mathbf{w})$ to construct confidence intervals $\left[l^{(\mathbf{w})}, u^{(\mathbf{w})}\right]$ which contain $\bar{f}(\mathbf{w})$ with high probability. The best plausible performance measure for a specific choice $\mathbf{w}$ of paramètres du modèle is $\tilde{f}(\mathbf{w}) := l^{(\mathbf{w})}$.

outlier Many apprentissage automatique methods are motivated by the i.i.d. assumption, which interprets point de donnéess as réalisations of i.i.d. VAs with a common loi de probabilité. The i.i.d. assumption is useful for applications where the statistical properties of the données generation process are stationary (or time-invariant) [87]. However, in some applications the données consists of a majority of regular point de donnéess that conform with an i.i.d. assumption as well as a small

number of point de donnéess that have fundamentally different statistical properties compared to the regular point de donnéess. We refer to a point de données that substantially deviates from the statistical properties of most point de donnéess as an outlier. Different methods for outlier detection use different measures for this deviation. Stastistical learning theory studies fundamental limits on the ability to mitigate outliers reliably [88, 89].

parameter space The parameter space $\mathcal{W}$ of an apprentissage automatique modèle $\mathcal{H}$ is the set of all feasible choices for the paramètres du modèle (see Figure 23). Many important apprentissage automatique methods use a modèle that is parametrized by vectors of the Euclidean space $\mathbb{R}^d$. Two widely used examples of parametrized modèles are modèle linéaires and deep nets. The parameter space is then often a subset $\mathcal{W} \subseteq \mathbb{R}^d$, e.g., all vectors $\mathbf{w} \in \mathbb{R}^d$ with a norme smaller than one.

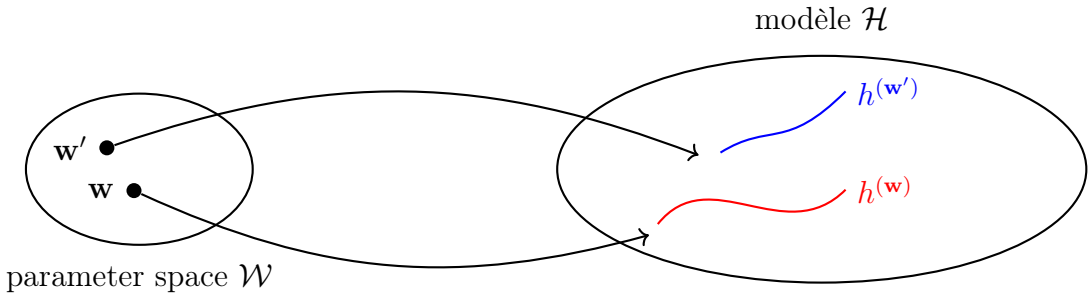


Fig. 23. The parameter space $\mathcal{W}$ of an apprentissage automatique modèle $\mathcal{H}$ consists of all feasible choices for the paramètres du modèle. Each choice $\mathbf{w}$ for the paramètres du modèle selects a hypothèse map $h^{(\mathbf{w})} \in \mathcal{H}$.

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

partitionnement de données Clustering methods decompose a given set of point de donnéess into a few subsets, which are referred to as clusters. Each cluster consists of point de donnéess that are more similar to each other than to point de donnéess outside the cluster. Different clustering methods use different measures for the similarity between point de donnéess and different forms of cluster representations. The clustering method k-moyennes uses the average caractéristique vector (cluster moyenne) of a cluster as its representative. A popular soft clustering method based on GMM represents a cluster by a loi normale multivariée.

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.

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 $\mathbf{w} = \left(w_1, \ldots, w_d\right)^T$ 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.

point de données Un point de données correspond à tout objet qui transmet de l'information [74]. Les points de données peuvent être des étudiants, des signaux radio, des arbres, des forêts, des images, des VA, des nombres réels ou des protéines. On caractérise les points de données à l'aide de deux types d'attributs. Le premier type d'attributs est appelé caractéristique. Les caractéristiques sont des attributs d'un point de données qui peuvent être mesurés ou calculés automatiquement. L'autre type d'attributs est appelé étiquette. L'étiquette d'un point de données représente un fait (ou une quantité d'intérêt) de plus haut niveau. Contrairement aux caractéristiques, déterminer l'étiquette d'un point de données nécessite généralement des experts humains (experts du

domaine). De manière générale, l'apprentissage automatique vise à prédire l'étiquette d'un point de données uniquement à partir de ses caractéristiques.

polynomial regression Polynomial régression aims at learning a polynomial hypothèse map to predict a numeric étiquette based on the numeric caractéristiques of a point de données. For point de donnéess characterized by a single numeric caractéristique, polynomial régression uses the espace des hypothèses $\mathcal{H}_d^{(\text{poly})} := \{h(x) = \sum_{j=0}^{d-1} x^j w_j\}$. The quality of a polynomial hypothèse map is measured using the average squared error loss incurred on a set of labeled datapoints (which we refer to as the ensemble d'entraînement).

predictor A predictor is a real-valued hypothèse map. Given a point de données with caractéristiques $\mathbf{x}$, the value $h(\mathbf{x}) \in \mathbb{R}$ is used as a prédiction for the true numeric étiquette $y \in \mathbb{R}$ of the point de données.

privacy funnel The privacy funnel is a method for learning privacy-friendly caractéristiques of point de donnéess [90].

privacy leakage Consider an apprentissage automatique application that processes a jeu de données $\mathcal{D}$ and delivers some output, such as the prédictions obtained for new point de donnéess. Privacy leakage arises if the output carries information about a private (or sensitive) caractéristique of a point de données (which might be a human) of $\mathcal{D}$. Based on a probabilistic model for the données generation, we can measure the privacy leakage via the MI between the output and the senstive caractéristique. Another quantitative measure of privacy leakage is DP.

The relations between different measures of privacy leakage have been studied in the literature (see [91]).

privacy protection Consider some apprentissage automatique method $\mathcal{A}$ that reads in a jeu de données $\mathcal{D}$ and delivers some output $\mathcal{A}(\mathcal{D})$. The output could be the learned paramètres du modèle $\widehat{\mathbf{w}}$ or the prédiction $\hat{h}(\mathbf{x})$ obtained for a specific point de données with caractéristiques $\mathbf{x}$. Many important apprentissage automatique applications involve point de donnéess representing humans. Each point de données is characterized by caractéristiques $\mathbf{x}$, potentially a étiquette y, and a sensitive attribute s (e.g., a recent medical diagnosis). Roughly speaking, privacy protection means that it should be impossible to infer, from the output $\mathcal{A}(\mathcal{D})$, any of the sensitive attributes of point de donnéess in $\mathcal{D}$. Mathematically, privacy protection requires non-invertibility of the map $\mathcal{A}(\mathcal{D})$. In general, just making $\mathcal{A}(\mathcal{D})$ non-invertible is typically insufficient for privacy protection. We need to make $\mathcal{A}(\mathcal{D})$ sufficiently non-invertible.

probabilistic model A probabilistic modèle interprets point de donnéess as réalisations of VAs with a joint loi de probabilité. This joint loi de probabilité typically involves paramètres which have to be manually chosen or learned via statistical inference methods such as maximum likelihood estimation [17].

probabilistic principal component analysis (PPCA) Probabilistic ACP extends basic ACP by using a probabilistic model for point de donnéess. The probabilistic model of probabilistic ACP reduces the task of dimen-

sionality reduction to an estimation problem that can be solved using EM methods.

probability density function (pdf) The probabilité density function $p(x)$ of a real-valued VA $x \in \mathbb{R}$ is a particular representation of its loi de probabilité. If the probabilité density function exists, it can be used to compute the probabilité that x takes on a value from a (measurable) set $\mathcal{B} \subseteq \mathbb{R}$ via $\mathbb{P}(x \in \mathcal{B}) = \int_{\mathcal{B}} p(x')dx'$ [7, Ch. 3]. The probabilité density function of a vector-valued VA $\mathbf{x} \in \mathbb{R}^d$ (if it exists) allows us to compute the probabilité of $\mathbf{x}$ belonging to a (measurable) region $\mathcal{R}$ via $\mathbb{P}(\mathbf{x} \in \mathcal{R}) = \int_{\mathcal{R}} p(\mathbf{x}')dx'_1 \dots dx'_d$ [7, Ch. 3].

probabilité We assign a probability value, typically chosen in the interval $[0, 1]$, to each event that might occur in a random experiment [6,7,39,92].

projected gradient descent (projected GD) Consider an MRE-based method that uses a parametrized modèle with parameter space $\mathcal{W} \subseteq \mathbb{R}^d$. Even if the objective function of MRE is lisse, we cannot use basic descente de gradient, as it does not take into account contraints on the optimization variable (i.e., the paramètres du modèle). Projected descente de gradient extends basic descente de gradient to handle constraints on the optimization variable (i.e., the paramètres du modèle). A single iteration of projected descente de gradient consists of first taking a gradient step and then projecting the result back onto the parameter space.

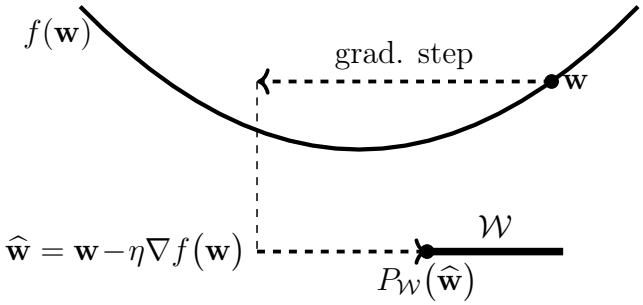


Fig. 24. Projected descente de gradient augments a basic gradient step with a projection back onto the constraint set $\mathcal{W}$.

projection Consider a subset $\mathcal{W} \subseteq \mathbb{R}^d$ of the d-dimensional Euclidean space. We define the projection $P_{\mathcal{W}}(\mathbf{w})$ of a vector $\mathbf{w} \in \mathbb{R}^d$ onto $\mathcal{W}$ as

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

In other words, $P_{\mathcal{W}}(\mathbf{w})$ is the vector in $\mathcal{W}$ which is closest to $\mathbf{w}$. The projection is only well-defined for subsets $\mathcal{W}$ for which the above minimum exists [27].

proximable A convexe function for which the proximal operator can be computed efficiently is sometimes referred to as proximable or simple [93].

proximal operator Given a convexe function $f(\mathbf{w}')$, we define its proximal operator as [59, 94]

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

As illustrated in Figure 25, evaluating the proximal operator amounts to minimizing a penalized variant of $f(\mathbf{w}')$. The penalty term is the scaled

squared Euclidean distance to a given vector $\mathbf{w}$ (which is the input to the proximal operator). The proximal operator can be interpreted as a generalization of the gradient step, which is defined for a lisse convexe function $f(\mathbf{w}')$. Indeed, taking a gradient step with taille de pas η at the current vector $\mathbf{w}$ is the same as applying the proximal operator of the function $\tilde{f}(\mathbf{w}') = \big(\nabla f(\mathbf{w})\big)^T(\mathbf{w}' - \mathbf{w})$ and using $\rho = 1/\eta$.

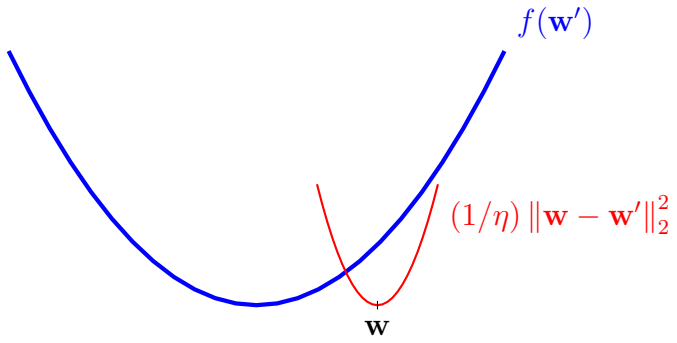


Fig. 25. A generalized gradient step updates a vector $\mathbf{w}$ by minimizing a penalized version of the function $f(\cdot)$. The penalty term is the scaled squared Euclidean distance between the optimization variable $\mathbf{w}'$ and the given vector $\mathbf{w}$.

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.

pseudoinverse The Moore–Penrose pseudoinverse $\mathbf{A}^+$ of a matrix $\mathbf{A} \in \mathbb{R}^{m \times d}$

generalizes the notion of an matrice inverse [3]. The pseudoinverse arises naturally within ridge regression when applied to a jeu de données with arbitrary étiquettes $\mathbf{y}$ and a feature matrix $\mathbf{X} = \mathbf{A}$ [20, Ch. 3]. The paramètres du modèle learned by ridge regression are given by

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

We can then define the pseudoinverse $\mathbf{A}^+ \in \mathbb{R}^{d\times m}$ via the limit [95, Ch. 3]

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

See also: matrice inverse, ridge regression, jeu de données, étiquette, feature matrix, paramètres du modèle, ridge regression.

quadratic function A function $f : \mathbb{R}^d \to \mathbb{R}$ of the form

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

with some matrix $\mathbf{Q} \in \mathbb{R}^{d\times d}$, vector $\mathbf{q} \in \mathbb{R}^d$, and scalar $a \in \mathbb{R}$.

random forest A random forest is a set of different arbre de décisions. Each of these arbre de décisions is obtained by fitting a perturbed copy of the original jeu de données.

rectified linear unit (ReLU) The ReLU is a popular choice for the fonction d'activation of a neuron within an RNA. It is defined as $\sigma(z) = \max\{0, z\}$, with z being the weighted input of the artificial neuron.

regret The regret of a hypothèse h relative to another hypothèse h', which serves as a baseline, is the difference between the perte incurred by h and the perte incurred by h' [43]. The baseline hypothèse h' is also referred to as an expert.

regularized empirical risk minimization (RERM) Basic MRE learns a hypothèse (or trains a modèle) $h \in \mathcal{H}$ based solely on the risque empirique $\widehat{L}\big(h|\mathcal{D}\big)$ incurred on a ensemble d'entraînement $\mathcal{D}$. To make MRE less prone to surapprentissage, we can implement régularisation by including a (scaled) regularizer $\mathcal{R}\big\{h\big\}$ in the learning objective. This leads to regularized empirical risk minimization (RERM),

$$\hat{h} \in \operatorname*{argmin}_{h \in \mathcal{H}} \widehat{L}\big(h|\mathcal{D}\big) + \alpha \mathcal{R}\big\{h\big\}. \tag{12}$$

The parameter $\alpha \geq 0$ controls the régularisation strength. For $\alpha = 0$, we recover standard MRE without régularisation. As α increases, the learned hypothèse is increasingly biased toward small values of $\mathcal{R}\big\{h\big\}$. The component $\alpha\mathcal{R}\big\{h\big\}$ in the objective function of (12) can be intuitively understood as a surrogate for the increased average perte that may occur when predicting étiquettes for point de donnéess outside the ensemble d'entraînement. This intuition can be made precise in various ways. For example, consider a modèle linéaire trained using squared error loss and the regularizer $\mathcal{R}\big\{h\big\} = \|\mathbf{w}\|_2^2$. In this setting, $\alpha\mathcal{R}\big\{h\big\}$ corresponds to the expected increase in perte caused by adding VA normale centrée réduites to the vecteur de caractéristiquess in the ensemble d'entraînement [8, Ch. 3]. A principled construction for the regularizer $\mathcal{R}\big\{h\big\}$ arises from approximate upper bounds on

the generalization error. The resulting RERM instance is known as SRM [96, Sec. 7.2].

regularized loss minimization (RLM) See RERM.

regularizer A regularizer assigns each hypothèse h from a espace des hypothèses $\mathcal{H}$ a quantitative measure $\mathcal{R}\big\{h\big\}$ for how much its prédiction error on a ensemble d'entraînement might differ from its prédiction errors on point de donnéess outside the ensemble d'entraînement. Ridge regression uses the regularizer $\mathcal{R}\big\{h\big\} := \|\mathbf{w}\|_2^2$ for linear hypothèse maps $h^{(\mathbf{w})}(\mathbf{x}) := \mathbf{w}^T\mathbf{x}$ [8, Ch. 3]. Lasso uses the regularizer $\mathcal{R}\big\{h\big\} := \|\mathbf{w}\|_1$ for linear hypothèse maps $h^{(\mathbf{w})}(\mathbf{x}) := \mathbf{w}^T\mathbf{x}$ [8, Ch. 3].

Rényi divergence The Rényi divergence measures the (dis)similarity between two loi de probabilités [97].

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

ridge regression Ridge régression learns the poids $\mathbf{w}$ of a linear hypothèse map $h^{(\mathbf{w})}(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$. The quality of a particular choice for the paramètres du modèle $\mathbf{w}$ is measured by the sum of two components.

The first component is the average squared error loss incurred by $h^{(\mathbf{w})}$ on a set of labeled datapoints (i.e., the ensemble d'entraînement). The second component is the scaled squared Euclidean norme $\alpha\|\mathbf{w}\|_2^2$ with a régularisation parameter $\alpha > 0$. Adding $\alpha\|\mathbf{w}\|_2^2$ to the average squared error loss is equivalent to replacing each original point de donnéess by the réalisation of (infinitely many) i.i.d. VAs centered around these point de donnéess (see régularisation).

risque Consider a hypothèse h used to predict the étiquette y of a point de données based on its caractéristiques $\mathbf{x}$. We measure the quality of a particular prédiction using a fonction de perte $L\left((\mathbf{x}, y), h\right)$. If we interpret point de donnéess as the réalisations of i.i.d. VAs, also the $L\left((\mathbf{x}, y), h\right)$ becomes the réalisation of an VA. The i.i.d. assumption allows us to define the risk of a hypothèse as the expected perte $\mathbb{E}\big\{L\left((\mathbf{x}, y), h\right)\big\}$. Note that the risk of h depends on both the specific choice for the fonction de perte and the loi de probabilité of the point de donnéess.

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 $\mathcal{D}$.

robustness TBD

Règlement général sur la protection des données (RGPD) The GDPR was enacted by the European Union (EU), effective from May 25, 2018 [28]. It safeguards the privacy and données rights of individuals in the EU. The GDPR has significant implications for how données is

collected, stored, and used in apprentissage automatique applications. Key provisions include the following:

- Data minimization principle: apprentissage automatique systems should only use the necessary amount of personal données for their purpose.
- Transparency and explainability: apprentissage automatique systems should enable their users to understand how the systems make decisions that impact the users.
- Données subject rights: Users should get an opportunity to access, rectify, and delete their personal données, as well as to object to automated decision-making and profiling.
- Accountability: Organizations must ensure robust données security and demonstrate compliance through documentation and regular audits.

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, 39]. Une réalisation de x est tout élément $a' \in \mathcal{N}$ pour lequel il existe un élément $\omega' \in \mathcal{P}$ tel que $x(\omega') = a'$.

réduction de dimension Dimensionality reduction methods map (typically many) raw caractéristiques to a (relatively small) set of new caractéristiques. These methods can be used to visualize point de donnéess by learning two caractéristiques that can be used as the coordinates of a depiction in a scatterplot.

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].

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 linear regression).

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

La figure 26 illustre ces trois approches de régularisation. Ces approches sont étroitement liées et parfois entièrement équivalentes : la data augmentation qui utilise des VA normales centrées réduites pour perturber les vecteurs de caractéristiques de l'ensemble d'entraînement dans le cas de la linear regression 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 ridge regression). 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 data augmentation que de réaliser un élagage de modèle.

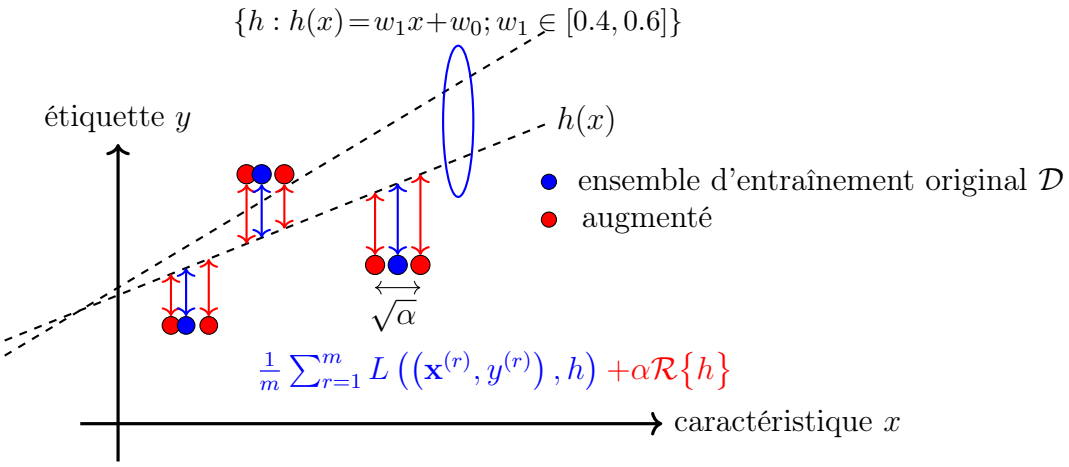


Fig. 26. Trois approches pour la régularisation: 1) data augmentation; 2) pénalisation de la perte; et 3) élagage du modèle (via des contraintes sur les paramètres du modèle).

réseau d'apprentissage fédéré Un réseau d'apprentissage fédéré est un graphe non orienté pondéré dont les nœuds représentent des générateurs de donnéess 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 local model. 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.

sample A finite sequence (or list) of point de donnéess $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}$ that is obtained or interpreted as the réalisation of m i.i.d. VAs with a common loi de probabilité $p(\mathbf{z})$. The length m of the sequence is referred to as the taille d'échantillon.

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

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

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

sample mean The sample moyenne $\mathbf{m} \in \mathbb{R}^d$ for a given jeu de données, with vecteur de caractéristiquess $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)} \in \mathbb{R}^d$, is defined as

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

scatterplot A visualization technique that depicts point de donnéess by markers in a two-dimensional plane. Fig. 27 depicts an example of a scatterplot.

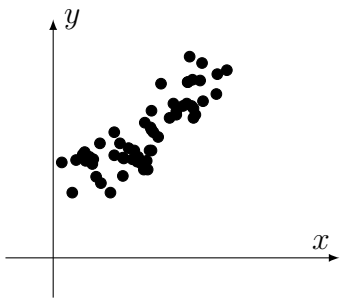


Fig. 27. A scatterplot with circle markers, where the point de donnéess represent daily weather conditions in Finland. Each point de données is characterized by its minimum daytime temperature x as the caractéristique and its maximum daytime temperature y as the étiquette. The temperatures have been measured at the FMI weather station Helsinki Kaisaniemi during 1.9.2024 - 28.10.2024.

A scatterplot can enable the visual inspection of points de données that are naturally represented by vecteurs de caractéristiques in high-dimensional spaces.

See also: réduction de dimension.

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 [73].

sensitive attribute apprentissage automatique revolves around learning a hypothèse map that allows us to predict the étiquette of a point de données from its caractéristiques. In some applications, we must ensure that the output delivered by an apprentissage automatique system does not allow us to infer sensitive attributes of a point de données. Which part of a point de données is considered a sensitive attribute is a design choice that varies across different application domains.

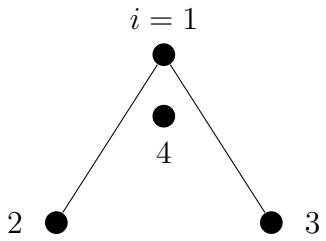
similarity graph Some apprentissage automatique applications generate point de donnéess that are related by a domain-specific notion of similarity. These similarities can be represented conveniently using a similarity graphe $\mathcal{G} = \big(\mathcal{V} := \{1, \ldots, m\}, \mathcal{E}\big)$. The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

soft clustering Soft partitionnement de données refers to the task of partitioning a given set of point de donnéess into (a few) overlapping clusters. Each point de données is assigned to several different clusters with varying degrees of belonging. Soft partitionnement de données methods determine the degree of belonging (or soft cluster assignment) for each point de données and each cluster. A principled approach to soft partitionnement de données is by interpreting point de donnéess as i.i.d. réalisations of a GMM. We then obtain a natural choice for the degree of belonging as the conditional probabilité of a point de données belonging to a specific mixture component.

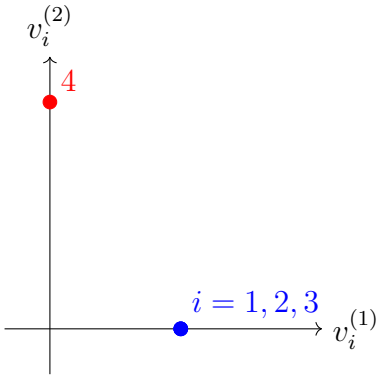
sous-apprentissage Consider an apprentissage automatique method that

uses MRE to learn a hypothèse with the minimum risque empirique on a given ensemble d'entraînement. Such a method is underfitting the ensemble d'entraînement if it is not able to learn a hypothèse with a sufficiently small risque empirique on the ensemble d'entraînement. If a method is underfitting, it will typically also not be able to learn a hypothèse with a small risque.

spectral clustering Spectral partitionnement de données is a particular instance of graph clustering, i.e., it clusters point de donnéess represented as the nodes $i = 1, \ldots, n$ of a graphe $\mathcal{G}$. Spectral partitionnement de données uses the vecteur propres of the matrice laplacienne $\mathbf{L}^{(\mathcal{G})}$ to construct vecteur de caractéristiquess $\mathbf{x}^{(i)} \in \mathbb{R}^d$ for each node (i.e., for each point de données) $i = 1, \ldots, n$. We can feed these vecteur de caractéristiquess into Euclidean space-based partitionnement de données methods, such as k-moyennes or soft clustering via GMM. Roughly speaking, the vecteur de caractéristiquess of nodes belonging to a well-connected subset (or cluster) of nodes in $\mathcal{G}$ are located nearby in the Euclidean space $\mathbb{R}^d$ (see Figure 28).



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



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

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

Fig. 28. **Top.** Left: An undirected graphe $\mathcal{G}$ with four nodes $i = 1, 2, 3, 4$, each representing a point de données. Right: The matrice laplacienne $\mathbf{L}^{(\mathcal{G})} \in \mathbb{R}^{4\times4}$ and its décomposition en éléments propres. **Bottom.** Left: A scatterplot of point de donnéess using the vecteur de caractéristiquess $\mathbf{x}^{(i)} = \left(v_i^{(1)}, v_i^{(2)}\right)^T$. Right: Two vecteur propres $\mathbf{v}^{(1)}, \mathbf{v}^{(2)} \in \mathbb{R}^d$ corresponding to the valeur propre $\lambda = 0$ of the matrice laplacienne $\mathbf{L}^{(\mathcal{G})}$.

spectrogram A spectrogram represents the time-frequency distribution of the energy of a time signal $x(t)$. Intuitively, it quantifies the amount of signal energy present within a specific time segment $[t_1, t_2] \subseteq \mathbb{R}$ and frequency interval $[f_1, f_2] \subseteq \mathbb{R}$. Formally, the spectrogram of a signal is defined as the squared magnitude of its short-time Fourier transform (STFT) [98]. Figure 29 depicts a time signal along with its spectrogram.

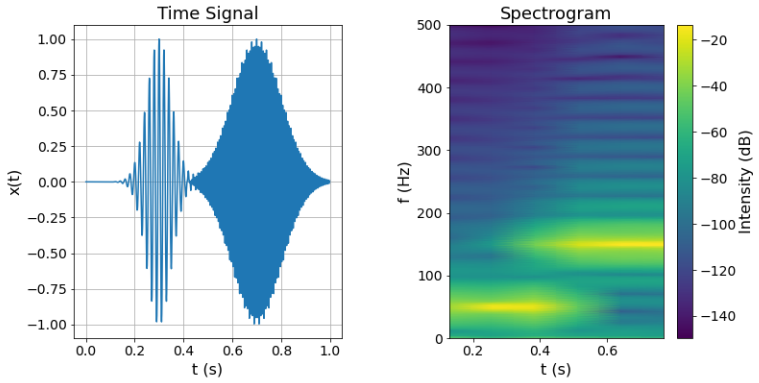


Fig. 29. Left: A time signal consisting of two modulated Gaussian pulses. Right: An intensity plot of the spectrogram.

The intensity plot of its spectrogram can serve as an image of a signal. A simple recipe for audio signal classification is to feed this signal image into deep nets originally developed for image classification and object detection [99]. It is worth noting that, beyond the spectrogram, several alternative representations exist for the time-frequency distribution of signal energy [100, 101].

squared error loss The squared error perte measures the prédiction error of a hypothèse h when predicting a numeric étiquette $y \in \mathbb{R}$ from the

caractéristiques $\mathbf{x}$ of a point de données. It is defined as

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

stability Stability is a desirable property of a apprentissage automatique method $\mathcal{A}$ that maps a jeu de données $\mathcal{D}$ (e.g., a ensemble d'entraînement) to an output $\mathcal{A}(\mathcal{D})$, such as learned paramètres du modèle or the prédiction for a specific point de données. Intuitively, $\mathcal{A}$ is stable if small changes in the input jeu de données $\mathcal{D}$ lead to small changes in the output $\mathcal{A}(\mathcal{D})$. Several formal notions of stability exist that enable bounds on the generalization error or risque of the method; see [55, Ch. 13]. To build intuition, consider the three datasets depicted in Fig. 30, each of which is equally likely under the same données-generating loi de probabilité. Since the optimal paramètres du modèle are determined by this underlying loi de probabilité, an accurate apprentissage automatique method $\mathcal{A}$ should return the same (or very similar) output $\mathcal{A}(\mathcal{D})$ for all three jeu de données. In other words, any useful $\mathcal{A}$ must be robust to variability in sample réalisations from the same loi de probabilité, i.e., it must be stable.

statistical aspects By statistical aspects of an apprentissage automatique method, we refer to (properties of) the loi de probabilité of its output under a probabilistic model for the données fed into the method.

stochastic block model (SBM) The stochastic block modèle is a probabilistic generative modèle for an undirected graphe $\mathcal{G} = \big(\mathcal{V}, \mathcal{E}\big)$ with a given set of nodes $\mathcal{V}$ [102]. In its most basic variant, the stochastic

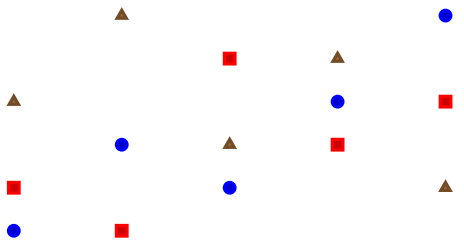


Fig. 30. Three jeu de donnéess $\mathcal{D}^{(*)}$, $\mathcal{D}^{(\square)}$, and $\mathcal{D}^{(\triangle)}$, each sampled independently from the same données-generating loi de probabilité. A stable apprentissage automatique method should return similar outputs when trained on any of these jeu de donnéess.

block modèle generates a graphe by first randomly assigning each node $i \in \mathcal{V}$ to a cluster index $c_i \in \{1, \ldots, k\}$. A pair of different nodes in the graphe is connected by an edge with probabilité $p_{i,i'}$ that depends solely on the étiquettes $c_i, c_{i'}$. The presence of edges between different pairs of nodes is statistically independent.

stochastic gradient descent (SGD) Stochastic descente de gradient is obtained from descente de gradient by replacing the gradient of the objective function with a stochastic approximation. A main application of stochastic descente de gradient is to train a parametrized modèle via MRE on a ensemble d'entraînement $\mathcal{D}$ that is either very large or not readily available (e.g., when point de donnéess are stored in a database distributed all over the planet). To evaluate the gradient of the risque

empirique (as a function of the paramètres du modèle $\mathbf{w}$), we need to compute a sum $\sum_{r=1}^{m} \nabla_{\mathbf{w}} L\left(\mathbf{z}^{(r)}, \mathbf{w}\right)$ over all point de donnéess in the ensemble d'entraînement. We obtain a stochastic approximation to the gradient by replacing the sum $\sum_{r=1}^{m} \nabla_{\mathbf{w}} L\left(\mathbf{z}^{(r)}, \mathbf{w}\right)$ with a sum $\sum_{r \in \mathcal{B}} \nabla_{\mathbf{w}} L\left(\mathbf{z}^{(r)}, \mathbf{w}\right)$ over a randomly chosen subset $\mathcal{B} \subseteq \{1, \ldots, m\}$ (see Figure 31). We often refer to these randomly chosen point de donnéess as a lot. The lot size $|\mathcal{B}|$ is an important parameter of stochastic descente de gradient. Stochastic descente de gradient with $|\mathcal{B}| > 1$ is referred to as mini-lot stochastic descente de gradient [103].

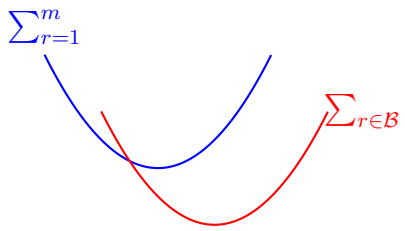


Fig. 31. Stochastic descente de gradient for MRE approximates the gradient $\sum_{r=1}^{m} \nabla_{\mathbf{w}} L\left(\mathbf{z}^{(r)}, \mathbf{w}\right)$ by replacing the sum over all point de donnéess in the ensemble d'entraînement (indexed by $r = 1, \ldots, m$) with a sum over a randomly chosen subset $\mathcal{B} \subseteq \{1, \ldots, m\}$.

strongly convex A continuously dérivable real-valued function $f(\mathbf{x})$ is strongly convexe with coefficient σ if $f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})^T(\mathbf{y}-\mathbf{x}) + (\sigma/2)\left\|\mathbf{y}-\mathbf{x}\right\|_2^2$ [76], [78, Sec. B.1.1].

structural risk minimization (SRM) Structural risk minimization (SRM) is an instance of RERM, which which the modèle $\mathcal{H}$ can be expressed as a countable union of sub-models: $\mathcal{H} = \bigcup_{n=1}^{\infty} \mathcal{H}^{(n)}$. Each sub-model

$\mathcal{H}^{(n)}$ permits the derivation of an approximate upper bound on the generalization error incurred when applying MRE to train $\mathcal{H}^{(n)}$. These individual bounds—one for each sub-model—are then combined to form a regularizer used in the RERM objective. These approximate upper bounds (one for each $\mathcal{H}^{(n)}$) are then combined to construct a regularizer for RERM [55, Sec. 7.2].

subgradient For a real-valued function $f : \mathbb{R}^d \to \mathbb{R} : \mathbf{w} \mapsto f(\mathbf{w})$, a vector $\mathbf{a}$ such that $f(\mathbf{w}) \geq f(\mathbf{w}') + \big(\mathbf{w} - \mathbf{w}'\big)^T \mathbf{a}$ is referred to as a subgradient of f at $\mathbf{w}'$ [104, 105].

subgradient descent Subgradient descent is a generalization of descente de gradient that does not require differentiability of the function to be minimized. This generalization is obtained by replacing the concept of a gradient with that of a subgradient. Similar to gradients, also subgradients allow us to construct local approximations of an objective function. The objective function might be the risque empirique $\widehat{L}\big(h^{(\mathbf{w})} \big| \mathcal{D}\big)$ viewed as a function of the paramètres du modèle $\mathbf{w}$ that select a hypothèse $h^{(\mathbf{w})} \in \mathcal{H}$.

support vector machine (SVM) The SVM is a binary classification method that learns a linear hypothèse map. Thus, like linear regression and logistic regression, it is also an instance of MRE for the modèle linéaire. However, the SVM uses a different fonction de perte from the one used in those methods. As illustrated in Figure 32, it aims to maximally separate point de donnéess from the two different classes in the espace des caractéristiques (i.e., maximum margin principle). Maximizing this

separation is equivalent to minimizing a regularized variant of the hinge loss (6) [41, 66, 106].

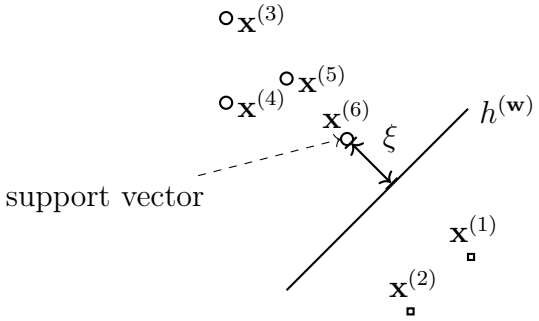


Fig. 32. The SVM learns a hypothèse (or classifier) $h^{(\mathbf{w})}$ with minimal average soft-margin hinge loss. Minimizing this perte is equivalent to maximizing the margin ξ between the frontière de décision of $h^{(\mathbf{w})}$ and each class of the ensemble d'entraînement.

The above basic variant of SVM is only useful if the point de donnéess from different categories can be (approximately) linearly separated. For an apprentissage automatique application where the categories are not derived from a noyau.

surapprentissage Consider an apprentissage automatique method that uses MRE to learn a hypothèse with the minimum risque empirique on a given ensemble d'entraînement. Such a method is overfitting the ensemble d'entraînement if it learns a hypothèse with a small risque empirique on the ensemble d'entraînement but a significantly larger perte outside the ensemble d'entraînement.

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 projected gradient descent (projected GD). 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].

total variation See GTV.

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 kernel methods [73]. Par ailleurs, les couches cachées

d'un deep net 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 interpretability en apprenant un petit nombre de caractéristiques pertinentes [79]. Le cas particulier d'une transformation de caractéristiques produisant deux caractéristiques numériques est particulièrement utile pour la visualisation des données. En effet, on peut représenter les points de données dans un scatterplot en utilisant ces deux caractéristiques comme coordonnées.

transparency Transparency is a fundamental requirement for trustworthy AI [107]. In the context of apprentissage automatique methods, transparency is often used interchangeably with explainability [47, 108]. However, in the broader scope of IA systems, transparency extends beyond explainability and includes providing information about the system's limitations, reliability, and intended use. In medical diagnosis systems, transparency requires disclosing the confidence level for the prédictions delivered by a trained modèle. In credit scoring, IA-based lending decisions should be accompanied by explanations of contributing factors, such as income level or credit history. These explanations allow humans (e.g., a loan applicant) to understand and contest automated decisions. Some apprentissage automatique methods inherently offer transparency. For example, logistic regression provides a quantitative measure of classification reliability through the value $|h(\mathbf{x})|$. Arbre de décisions are another example, as they allow human-readable decision

rules [49]. Transparency also requires a clear indication when a user is engaging with an IA system. For example, IA-powered chatbots should notify users that they are interacting with an automated system rather than a human. Furthermore, transparency encompasses comprehensive documentation detailing the purpose and design choices underlying the IA system. For instance, modèle datasheets [72] and IA system cards [109] help practitioners understand the intended use cases and limitations of an IA system [110].

trustworthy artificial intelligence (trustworthy AI) Besides the computational aspects and statistical aspects, a third main design aspect of apprentissage automatique methods is their trustworthiness [111]. The EU has put forward seven key requirements (KRs) for trustworthy IA (that typically build on apprentissage automatique methods) [112]:

1) KR1 - Human agency and oversight;
2) KR2 - Technical robustness and safety;
3) KR3 - Privacy and data governance;
4) KR4 - Transparency;
5) KR5 - Diversity, non-discrimination and fairness;
6) KR6 - Societal and environmental well-being;
7) KR7 - Accountability.

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'label space 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 objective function. 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 [113], [114], [115].

uncertainty Uncertainty refers to the degree of confidence—or lack thereof—associated with a quantity such as a model prediction, parameter estimate, or observed data point. In apprentissage automatique, uncertainty arises from various sources, including noisy data, limited training samples, or ambiguity in model assumptions. Probability theory offers a principled framework for representing and quantifying such uncertainty.

upper confidence bound (UCB) Consider a apprentissage automatique application that requires selecting, at each time step k, an action a_k from a finite set of alternatives $\mathcal{A}$. The utility of selecting action a_k is quantified by a numeric reward signal $r^{(a_k)}$. A widely used proba-

bilistic model for this type of sequential decision-making problem is the stochastic multi-armed bandit setting [84]. In this model, the reward $r^{(a)}$ is viewed as the réalisation of a VA with unknown moyenne $\mu^{(a)}$. Ideally, we would always choose the action with the largest expected reward $\mu^{(a)}$, but these means are unknown and must be estimated from observed données. Simply choosing the action with the largest estimate $\widehat{\mu}^{(a)}$ can lead to suboptimal outcomes due to estimation uncertainty. The UCB strategy addresses this by selecting actions not only based on their estimated means but also by incorporating a term that reflects the uncertainty in these estimates—favouring actions with high potential reward and high uncertainty. Theoretical guarantees for the performance of UCB strategies,including logarithmic regret bounds, are established in [84].

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

validation Consider a hypothèse $\hat{h}$ that has been learned via some apprentissage automatique method, e.g., by solving MRE on a ensemble d'entraînement $\mathcal{D}$. Validation refers to the practice of evaluating the perte incurred by the hypothèse $\hat{h}$ on a set of point de donnéess that are not contained in the ensemble d'entraînement $\mathcal{D}$.

Vapnik–Chervonenkis dimension (VC dimension) The VC dimension of an infinite espace des hypothèses is a widely-used measure for its size. We refer to the literature (see [55]) for a precise definition of

VC dimension as well as a discussion of its basic properties and use in apprentissage automatique.

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 [19], [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'Euclidean space $\mathbb{R}^d$.

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

variable aléatoire normale centrée réduite Une VA normale centrée réduite est une VA réelle x dont la pdf est donnée par [7], [19], [75]

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

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 [116].

Les VA normales sont largement utilisées comme probabilistic models 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 [38].

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 Euclidean space 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 kernel method).

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

vertical federated learning (vertical FL) Vertical apprentissage fédéré uses jeu de données locals that are constituted by the same point de donnéess but characterizing them with different caractéristiques [117]. For example, different healthcare providers might all contain information about the same population of patients. However, different healthcare providers collect different measurements (e.g., blood values, electrocardiography, lung X-ray) for the same patients.

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

voisins The neighbors of a node $i \in \mathcal{V}$ within an réseau d'apprentissage fédéré are those nodes $i' \in \mathcal{V} \setminus \{i\}$ that are connected (via an edge) to node i.

zero-gradient condition Consider the unconstrained optimization problem $\min_{\mathbf{w} \in \mathbb{R}^{d}} f(\mathbf{w})$ with a lisse and convexe objective function $f(\mathbf{w})$. A necessary and sufficient condition for a vector $\widehat{\mathbf{w}} \in \mathbb{R}^{d}$ to solve this problem is that the gradient $\nabla f\big(\widehat{\mathbf{w}}\big)$ is the zero vector,

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

0/1 loss The 0/1 perte $L^{(0/1)}\left((\mathbf{x}, y), h\right)$ measures the quality of a classifier $h(\mathbf{x})$ that delivers a prédiction $\hat{y}$ (e.g., via thresholding (1)) for the étiquette y of a point de données with caractéristiques $\mathbf{x}$. It is equal to 0 if the prédiction is correct, i.e., $L^{(0/1)}\left((\mathbf{x}, y), h\right) = 0$ when $\hat{y} = y$. It is equal to 1 if the prédiction is wrong, i.e., $L^{(0/1)}\left((\mathbf{x}, y), h\right) = 1$ when $\hat{y} \neq y$.

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 [27], [104].

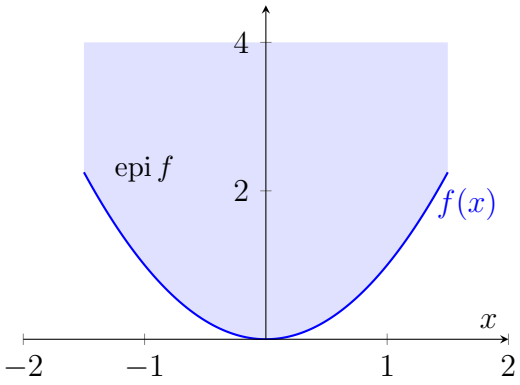


Fig. 33. Épigraphe de la fonction $f(x) = x^2$ (i.e., la zone colorée).

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 » [22], [23], [24].

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

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

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

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

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

[16] 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.

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

[18] 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.

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

[20] 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.

[21] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001, doi: 10.1214/aos/1013203451.

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

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

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

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

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

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

[28] 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

[29] 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

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

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

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

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

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

[35] 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.

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

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

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

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

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

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

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

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

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

[45] 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.*, 2022.

[46] 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.

[47] 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.

[48] 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

[49] 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.

[50] 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/

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

[52] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. 20th Int. Conf. Artif. Intell. Statist.*, 2017, A. Singh and J. Zhu, Eds. vol. 54, 2017, pp. 1273–1282. [Online]. Available: https://proceedings.mlr.press/v54/mcmahan17a.html

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

[54] Y. SarcheshmehPour, Y. Tian, L. Zhang, and A. Jung, "Flow-based

clustering and spectral clustering: A comparison," in *2021 55th Asilomar Conf. Signals, Syst., Comput.*, 2021.

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

[56] 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.

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

[58] H. P. Lopuhaä and P. J. Rousseeuw, "Breakdown points of affine equivariant estimators of multivariate location and covariance matrices," *Ann. Statist.*, vol. 19, no. 1, pp. 229–248, Mar. 1991, doi: 10.1214/aos/1176347978.

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

[60] A. Vaswani et al., "Attention is all you need," in *Adv. Neural Inf. Process. Syst.*, 2017, I. Guyon, U. von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. vol. 30, 2017, pp. 5998–6008. [Online]. Available: https://papers.nips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[61] 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.

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

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

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

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

[66] 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.

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

[68] 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/

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

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

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

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

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

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

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

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

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

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

[79] 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.

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

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

[82] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Adv. Neural Inf. Process. Syst.*, 2001. [Online]. Available: https://papers.nips.cc/paper_files/paper/2001/hash/801272ee79cfde7fa5960571fee36b9b-Abstract.html

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

[84] 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.

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

[86] A. Rakhlin, O. Shamir, and K. Sridharan, "Making gradient descent optimal for strongly convex stochastic optimization," in

Proc. 29th Int. Conf. Mach. Learn., 2012, J. Langford and J. Pineau, Eds. 2012, pp. 449–456. [Online]. Available: https://icml.cc/Conferences/2012/papers/261.pdf

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

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

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

[90] A. Makhdoumi, S. Salamatian, N. Fawaz, and M. Médard, "From the information bottleneck to the privacy funnel," in *2014 IEEE Inf. Theory Workshop*, 2014, pp. 501–505, doi: 10.1109/ITW.2014.6970882.

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

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

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

[94] 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.

[95] A. Ben-Israel and T. N. E. Greville, *Generalized Inverses: Theory and Applications*, 2nd ed., ser. CMS Books in Mathematics. New York, NY: Springer, 2003, originally published by Wiley-Interscience, 1974. [Online]. Available: https://doi.org/10.1007/b97366

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

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

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

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

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

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

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

[103] 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.

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

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

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

[107] 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

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

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

[110] 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.

[111] 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

[112] 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

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

[114] 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.

[115] 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.

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

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